

CareRoute source code

Source revision 6cacdcf (full revision is in the source ZIP Git provenance notes). Complete tracked source is also supplied in CareRoute-source.zip; dependencies and third-party UI components are credited in package.json.

.oxfmtc.json

```
1 {
2   "$schema": "./node_modules/oxfmt/configuration_schema.json",
3   "singleQuote": true,
4   "printWidth": 80,
5   "sortPackageJson": false,
6   "ignorePatterns": [
7     "package-lock.json",
8     "pnpm-lock.yaml",
9     "yarn.lock",
10    "bun.lock",
11    "bun.lockb"
12  ]
13 }
```

.oxlintrc.json

```
1 {
2   "$schema": "./node_modules/oxlint/configuration_schema.json",
3   "plugins": [
4     "eslint",
5     "typescript",
6     "unicorn",
7     "oxc",
8     "react",
9     "import",
10    "jsx-a11y",
11    "nextjs"
12  ],
13  "categories": {
14    "correctness": "error"
15  },
16  "env": {
17    "builtin": true,
18    "browser": true,
19    "node": true
20  },
21  "options": {
22    "typeAware": true,
23    "typeCheck": true
24  },
25  "ignorePatterns": [
26    ".next/**",
27    ".vinext/**",
28    "out/**",
29    "build/**",
30    "dist/**",
31    "coverage/**",
32    "next-env.d.ts"
33  ],
34  "rules": {
35    "no-array-constructor": "error",
36    "no-var": "error",
37    "prefer-const": "error",
38    "prefer-rest-params": "error",
39    "prefer-spread": "error",
40    "import/no-anonymous-default-export": "warn",
41    "react/display-name": "error",
42    "react/jsx-no-comment-textnodes": "error",
43    "react/no-unescaped-entities": "error",
44    "react/react-compiler": "error",
45    "react/require-render-return": "error",
46    "react/rules-of-hooks": "error",
47    "typescript/ban-ts-comment": "error",
48    "typescript/no-deprecated": "error",
49    "typescript/no-empty-object-type": "error",
50    "typescript/no-explicit-any": "error",
51    "typescript/no-namespace": "error",
52    "typescript/no-require-imports": "error",
53    "typescript/no-unnecessary-type-constraint": "error",
54    "typescript/no-unsafe-function-type": "error"
55  }
56 }
```

app/globals.css

```
1 @import 'tailwindcss';
2 @import 'tw-animate-css';
3 @import 'shadcn/tailwind.css';
4
5 @custom-variant dark (&:is(.dark *));
6
7 @theme inline {
8   --color-background: var(--background);
9   --color-foreground: var(--foreground);
10  --font-sans: Arial, Helvetica, sans-serif;
11  --font-mono: monospace;
12  --font-heading: var(--font-sans);
13  --color-sidebar-ring: var(--sidebar-ring);
14  --color-sidebar-border: var(--sidebar-border);
15  --color-sidebar-accent-foreground: var(--sidebar-accent-foreground);
16  --color-sidebar-accent: var(--sidebar-accent);
17  --color-sidebar-primary-foreground: var(--sidebar-primary-foreground);
18  --color-sidebar-primary: var(--sidebar-primary);
19  --color-sidebar-foreground: var(--sidebar-foreground);
20  --color-sidebar: var(--sidebar);
21  --color-chart-5: var(--chart-5);
22  --color-chart-4: var(--chart-4);
23  --color-chart-3: var(--chart-3);
24  --color-chart-2: var(--chart-2);
25  --color-chart-1: var(--chart-1);
26  --color-ring: var(--ring);
27  --color-input: var(--input);
28  --color-border: var(--border);
29  --color-destructive: var(--destructive);
30  --color-accent-foreground: var(--accent-foreground);
31  --color-accent: var(--accent);
32  --color-muted-foreground: var(--muted-foreground);
33  --color-muted: var(--muted);
34  --color-secondary-foreground: var(--secondary-foreground);
35  --color-secondary: var(--secondary);
36  --color-primary-foreground: var(--primary-foreground);
37  --color-primary: var(--primary);
38  --color-popover-foreground: var(--popover-foreground);
39  --color-popover: var(--popover);
40  --color-card-foreground: var(--card-foreground);
41  --color-card: var(--card);
42  --radius-sm: calc(var(--radius) * 0.6);
43  --radius-md: calc(var(--radius) * 0.8);
44  --radius-lg: var(--radius);
45  --radius-xl: calc(var(--radius) * 1.4);
46  --radius-2xl: calc(var(--radius) * 1.8);
47  --radius-3xl: calc(var(--radius) * 2.2);
48  --radius-4xl: calc(var(--radius) * 2.6);
49 }
50
51 body {
52   font-family: Arial, Helvetica, sans-serif;
53   background:
54     radial-gradient(
55       circle at 8% 8%,
56       oklch(0.93 0.08 173 / 0.75),
57       transparent 30rem
58     ),
59     radial-gradient(
60       circle at 92% 18%,
61       oklch(0.94 0.06 65 / 0.55),
62       transparent 25rem
63     ),
64     var(--background);
65 }
66
67 :root {
68   --background: oklch(0.975 0.012 176);
69   --foreground: oklch(0.22 0.035 207);
70   --card: oklch(1 0 0);
71   --card-foreground: oklch(0.145 0 0);
72   --popover: oklch(1 0 0);
73   --popover-foreground: oklch(0.145 0 0);
74   --primary: oklch(0.48 0.105 181);
75   --primary-foreground: oklch(0.985 0 0);
76   --secondary: oklch(0.94 0.035 176);
77   --secondary-foreground: oklch(0.205 0 0);
78   --muted: oklch(0.95 0.018 180);
79   --muted-foreground: oklch(0.48 0.035 205);
80   --accent: oklch(0.97 0 0);
81   --accent-foreground: oklch(0.205 0 0);
82   --destructive: oklch(0.577 0.245 27.325);
```

```

83 --border: oklch(0.87 0.025 184);
84 --input: oklch(0.87 0.025 184);
85 --ring: oklch(0.58 0.1 181);
86 --chart-1: oklch(0.87 0 0);
87 --chart-2: oklch(0.556 0 0);
88 --chart-3: oklch(0.439 0 0);
89 --chart-4: oklch(0.371 0 0);
90 --chart-5: oklch(0.269 0 0);
91 --radius: 0.85rem;
92 --sidebar: oklch(0.985 0 0);
93 --sidebar-foreground: oklch(0.145 0 0);
94 --sidebar-primary: oklch(0.205 0 0);
95 --sidebar-primary-foreground: oklch(0.985 0 0);
96 --sidebar-accent: oklch(0.97 0 0);
97 --sidebar-accent-foreground: oklch(0.205 0 0);
98 --sidebar-border: oklch(0.922 0 0);
99 --sidebar-ring: oklch(0.708 0 0);
100 }
101
102 .route-line {
103   position: absolute;
104   inset: -4rem -12rem auto auto;
105   width: 30rem;
106   height: 30rem;
107   border: 2px dashed oklch(0.62 0.09 181 / 0.23);
108   border-radius: 52% 48% 68% 32%;
109   transform: rotate(18deg);
110   pointer-events: none;
111 }
112
113 ::selection {
114   background: oklch(0.82 0.08 181);
115   color: oklch(0.2 0.04 207);
116 }
117
118 .dark {
119   --background: oklch(0.145 0 0);
120   --foreground: oklch(0.985 0 0);
121   --card: oklch(0.205 0 0);
122   --card-foreground: oklch(0.985 0 0);
123   --popover: oklch(0.205 0 0);
124   --popover-foreground: oklch(0.985 0 0);
125   --primary: oklch(0.922 0 0);
126   --primary-foreground: oklch(0.205 0 0);
127   --secondary: oklch(0.269 0 0);
128   --secondary-foreground: oklch(0.985 0 0);
129   --muted: oklch(0.269 0 0);
130   --muted-foreground: oklch(0.708 0 0);
131   --accent: oklch(0.269 0 0);
132   --accent-foreground: oklch(0.985 0 0);
133   --destructive: oklch(0.704 0.191 22.216);
134   --border: oklch(1 0 0 / 10%);
135   --input: oklch(1 0 0 / 15%);
136   --ring: oklch(0.556 0 0);
137   --chart-1: oklch(0.87 0 0);
138   --chart-2: oklch(0.556 0 0);
139   --chart-3: oklch(0.439 0 0);
140   --chart-4: oklch(0.371 0 0);
141   --chart-5: oklch(0.269 0 0);
142   --sidebar: oklch(0.205 0 0);
143   --sidebar-foreground: oklch(0.985 0 0);
144   --sidebar-primary: oklch(0.488 0.243 264.376);
145   --sidebar-primary-foreground: oklch(0.985 0 0);
146   --sidebar-accent: oklch(0.269 0 0);
147   --sidebar-accent-foreground: oklch(0.985 0 0);
148   --sidebar-border: oklch(1 0 0 / 10%);
149   --sidebar-ring: oklch(0.556 0 0);
150 }
151
152 @layer base {
153   * {
154     @apply border-border outline-ring/50;
155   }
156   body {
157     @apply bg-background text-foreground;
158   }
159   html {
160     @apply font-sans;
161   }
162 }
163
164 button:focus-visible, a:focus-visible { outline: 3px solid var(--primary); outline-offset: 3px; }
165 @media (prefers-reduced-motion: reduce) { *, *::before, *::after { animation: none !important;
  transition: none !important; scroll-behavior: auto !important; } }
166 @media print { header, .route-line, #care-inputs { display: none !important; } body { background: white;

```


app/page.tsx

```
1 'use client';
2
3 import { useEffect, useRef, useState } from 'react';
4 import { suggestIntent } from '@lib/intent-model.mjs';
5 import {
6   ArrowRight,
7   CheckCircle2,
8   ExternalLink,
9   HeartPulse,
10  MapPin,
11  PhoneCall,
12  ShieldCheck,
13  Sparkles,
14  Stethoscope,
15 } from 'lucide-react';
16 import { Alert, AlertDescription, AlertTitle } from '@components/ui/alert';
17 import { Badge } from '@components/ui/badge';
18 import { Button } from '@components/ui/button';
19 import { Card, CardContent } from '@components/ui/card';
20 import { Checkbox } from '@components/ui/checkbox';
21 import { Textarea } from '@components/ui/textarea';
22
23 type Mode = 'now' | 'affordable' | 'prepare';
24 const MODES = [
25   { id: 'now' as const, label: 'I need care now', icon: HeartPulse },
26   { id: 'affordable' as const, label: 'Find affordable care', icon: MapPin },
27   { id: 'prepare' as const, label: 'Prepare for a visit', icon: Stethoscope },
28 ];
29 const RED_FLAGS = [
30   'Trouble breathing, fainting, or not responding',
31   'Chest pain, stroke signs, or a severe injury',
32   'Bleeding that will not stop',
33   'Thoughts of suicide or harming someone',
34 ];
35
36 export default function Home() {
37   const [mode, setMode] = useState<Mode>('now');
38   const [concern, setConcern] = useState('');
39   const [flags, setFlags] = useState<string[]>([]);
40   const [generated, setGenerated] = useState(false);
41   const crisis = flags.includes(RED_FLAGS[3]);
42   const emergency = flags.some(flag => flag !== RED_FLAGS[3]);
43   const firstWords = concern.trim() || 'your concern';
44   const suggestion = suggestIntent(concern);
45   const resultRef = useRef<HTMLDivElement>(null);
46   useEffect(() => { if (generated) resultRef.current?.focus(); }, [generated]);
47
48   function toggleFlag(flag: string) {
49     setFlags((current) =>
50       current.includes(flag)
51         ? current.filter((item) => item !== flag)
52         : [...current, flag],
53     );
54     setGenerated(false);
55   }
56
57   return (
58     <main className="min-h-screen overflow-hidden">
59       <header className="mx-auto flex max-w-6xl items-center justify-between px-5 py-5 md:px-8">
60         <a
61           className="flex items-center gap-2 font-semibold tracking-tight"
62           href="#top"
63           aria-label="CareRoute home"
64         >
65           <span className="grid size-9 place-items-center rounded-xl bg-primary text-primary-foreground
66             shadow-sm">
67             <HeartPulse className="size-5" />
68             </span>
69           <span className="text-lg">CareRoute</span>
70         </a>
71         <Badge
72           variant="outline"
73           className="border-primary/20 bg-white/60 text-primary"
74         >
75           <ShieldCheck /> Privacy-first prototype
76         </Badge>
77       </header>
78
79       <section
80         id="top"
81         className="relative mx-auto max-w-6xl px-5 pb-16 pt-8 md:px-8 md:pt-14"
82       >
```

```

82     <div className="route-line" aria-hidden="true" />
83     <div className="relative grid items-start gap-10 lg:grid-cols-[0.78fr_1.22fr] lg:gap-14">
84         <div className="pt-3">
85             <Badge className="mb-5 bg-primary/10 text-primary hover:bg-primary/10">
86                 Free • no sign-in
87             </Badge>
88             <h1 className="max-w-xl text-balance text-5xl font-semibold leading-[0.98]
tracking-[-0.055em] md:text-6xl">
89                 Know where to go.{ ' '}
90                 <span className="text-primary">Know what to ask.</span>
91             </h1>
92             <p className="mt-6 max-w-lg text-pretty text-base leading-7 text-muted-foreground md:text-
lg">
93                 A two-minute guide to safer next steps, affordable local care, and
94                 a more prepared medical visit.
95             </p>
96             <div className="mt-8 grid max-w-md gap-3 text-sm text-muted-foreground sm:grid-cols-3">
97                 {[ 'No diagnosis', 'No data saved', 'Official resources' ].map(
98                     (item) => (
99                     <div key={item} className="flex items-center gap-2">
100                         <CheckCircle2 className="size-4 text-primary" /> {item}
101                     </div>
102                 ),
103                 )}
104             </div>
105         </div>
106
107         <Card className="relative overflow-visible rounded-[1.75rem] border-0 bg-white/90 py-0
shadow-[0_24px_80px_-28px_rgba(15,55,60,0.35)] ring-1 ring-primary/10 backdrop-blur">
108             <CardContent className="p-4 md:p-6">
109                 <div id="care-inputs">
110                     <div className="grid gap-2 rounded-2xl bg-muted p-1.5 sm:grid-cols-3" aria-label="Choose a
care tool">
111                         {MODES.map(({ id, label, icon: Icon }) => (
112                             <button
113                                 key={id}
114                                 type="button"
115                                 aria-pressed={mode === id}
116                                 onClick={() => {
117                                     setMode(id);
118                                     setGenerated(false);
119                                 }}
120                                 className={`flex min-h-11 items-center justify-center gap-2 rounded-xl px-3 text-sm
font-medium transition ${mode === id ? 'bg-white text-foreground shadow-sm' : 'text-muted-foreground
hover:text-foreground'}}`>
121                             >
122                             <Icon className="size-4" /> {label}
123                         </button>
124                     )})
125                 </div>
126                 <div className="mt-6">
127                     <p className="text-xs font-semibold uppercase tracking-[0.16em] text-primary">
Safety check
128                     </p>
129                     <h2 className="mt-2 text-2xl font-semibold tracking-tight">
Is any of this happening right now?
130                     </h2>
131                     <div className="mt-4 grid gap-3 sm:grid-cols-2">
132                         {RED_FLAGS.map((flag) => (
133                             <label
134                                 key={flag}
135                                 className="flex cursor-pointer items-start gap-3 rounded-xl border bg-
background/70 p-3 text-sm leading-5 transition hover:border-primary/30"
136                             >
137                                 <Checkbox
138                                     checked={flags.includes(flag)}
139                                     onChange={() => toggleFlag(flag)}
140                                 />
141                                 <span>{flag}</span>
142                             </label>
143                         )})
144                     </div>
145                     <label
146                         className="mt-5 block text-sm font-medium"
147                         htmlFor="concern"
148                     >
Briefly describe what you need help with{ ' '}
149                     <span className="font-normal text-muted-foreground">
(optional)
150                     </span>
151                 </label>
152                 <Textarea
153                     id="concern"
154                     value={concern}
155                     onChange={(event) => {

```

```

160         setConcern(event.target.value);
161         setGenerated(false);
162     }}
163     className="mt-2 min-h-24 resize-none bg-background"
164     placeholder="Example: I have had a sore throat for two days and do not have
insurance."
165     maxLength={1000}
166     aria-describedby="privacy-note"
167     />
168     <p id="privacy-note" className="mt-2 text-xs leading-5 text-muted-foreground">US
resources. Your text stays on this device and clears when you reload. Avoid names or identifying
details.</p>
169     {suggestion && !flags.length && suggestion !== mode && (
170         <div className="mt-3 rounded-xl border bg-background p-3 text-sm">
171             <p>Suggested tool: {suggestion === 'affordable' ? 'affordable care' : 'visit
preparation'}.</p>
172             <p className="mt-1 text-xs text-muted-foreground">A small on-device model suggests a
tool from your wording. You decide; it does not assess your health.</p>
173             <Button variant="outline" className="mt-2" onClick={() =>
{setMode(suggestion);setGenerated(false);}}>Use suggested tool</Button>
174         </div>
175     )}
176     <Button
177         size="lg"
178         className="mt-5 h-11 w-full rounded-xl"
179         onClick={() => setGenerated(true)}
180     >
181         Build my next-step plan <ArrowRight data-icon="inline-end" />
182     </Button>
183 </div>
184 </div>
185     {(generated || emergency || crisis) && (
186         <div ref={resultRef} tabIndex={-1} role="region" aria-label="Your next steps" aria-
live="polite" className="mt-6 rounded-2xl outline-none animate-in fade-in slide-in-from-bottom-2">
187             {emergency ? (
188                 <Alert variant="destructive" className="border-destructive/30 bg-destructive/5 p-4">
189                     <PhoneCall className="size-5" />
190                     <AlertTitle>Call 911 now</AlertTitle>
191                     <AlertDescription>
192                         These warning signs may be an emergency. Do not wait for this website. <a
href="tel:911" className="font-semibold underline">Call 911</a>.
193                         {crisis && <span> You also selected a crisis warning. Tell the emergency
operator; <a href="tel:988" className="font-semibold underline">988</a> offers additional US crisis
support.</span>}
194                     </AlertDescription>
195                     </Alert>
196                 ) : crisis ? (
197                     <Alert
198                         variant="destructive"
199                         className="border-destructive/30 bg-destructive/5 p-4"
200                     >
201                         <PhoneCall className="size-5" />
202                         <AlertTitle>Get immediate crisis support</AlertTitle>
203                         <AlertDescription>
204                             Call or text <a href="tel:988" className="font-semibold underline">988</a> now.
If anyone is in
205                             immediate danger, call <a href="tel:911" className="font-semibold
underline">911</a>.
206                         </AlertDescription>
207                     </Alert>
208                 ) : (
209                     <Plan mode={mode} concern={firstWords} />
210                 )}
211             {!flags.length && <Button variant="outline" className="mt-3" onClick={() =>
window.print()}>Print my plan</Button>}
212         </div>
213     )}
214     {(concern || flags.length > 0 || generated) && <Button variant="ghost" className="mt-3"
onClick={() => {setConcern('');setFlags([]);setGenerated(false);setMode('now');}}>Clear and start
over</Button>}
215 </CardContent>
216 </Card>
217 </div>
218 </section>
219
220 <section className="border-t bg-white/55">
221     <div className="mx-auto grid max-w-6xl gap-8 px-5 py-10 text-sm text-muted-foreground md:grid-
cols-[1fr_auto] md:px-8">
222         <div>
223             <p className="font-semibold text-foreground">
224                 CareRoute is an educational prototype, not medical advice.
225             </p>
226             <p className="mt-1 max-w-2xl">
227                 It does not diagnose conditions or replace a clinician.
228                 Information entered here stays in this browser session.

```

```

229         Not selecting a warning sign does not rule out an emergency. These resources are for the
United States.
230     </p>
231 </div>
232 <div className="flex flex-wrap gap-4">
233     <a
234         className="hover:text-primary"
235         href="https://medlineplus.gov/ency/article/001927.htm"
236         target="_blank"
237         rel="noreferrer"
238     >
239         Emergency guidance <ExternalLink className="inline size-3" />
240     </a>
241     <a
242         className="hover:text-primary"
243         href="https://findahealthcenter.hrsa.gov/"
244         target="_blank"
245         rel="noreferrer"
246     >
247         HRSA centers <ExternalLink className="inline size-3" />
248     </a>
249     <a
250         className="hover:text-primary"
251         href="https://211.org/"
252         target="_blank"
253         rel="noreferrer"
254     >
255         211 help <ExternalLink className="inline size-3" />
256     </a>
257 </div>
258 </div>
259 </section>
260 </main>
261 );
262 }
263
264 function Plan({ mode, concern }: { mode: Mode; concern: string }) {
265     if (mode === 'affordable')
266         return (
267             <div className="rounded-2xl border border-primary/15 bg-primary/5 p-5">
268                 <p className="flex items-center gap-2 font-semibold text-primary">
269                     <MapPin className="size-4" /> Your affordable-care route
270                 </p>
271                 <ol className="mt-4 space-y-3 text-sm leading-6">
272                     <li>
273                         <strong>1. Search a community health center.</strong> They serve
274                         people with or without insurance and use a sliding fee scale.
275                     </li>
276                     <li>
277                         <strong>2. Call 211.</strong> Ask for nearby low-cost clinics,
278                         transportation, or prescription help.
279                     </li>
280                     <li>
281                         <strong>3. Before visiting, ask:</strong> "What will this cost
282                         without insurance, and do you offer financial assistance?"
283                     </li>
284                 </ol>
285                 <div className="mt-5 flex flex-wrap gap-2">
286                     <Button
287                         render={
288                             <a
289                                 href="https://findahealthcenter.hrsa.gov/"
290                                 target="_blank"
291                                 rel="noreferrer"
292                                 aria-label="Find a community health center"
293                             />
294                         >
295                     <MapPin /> Find a health center
296                 </Button>
297                     <Button
298                         variant="outline"
299                         render={
300                             <a
301                                 href="https://211.org/"
302                                 target="_blank"
303                                 rel="noreferrer"
304                                 aria-label="Open 211 local services"
305                             />
306                         >
307                     <MapPin /> Open 211
308                 </Button>
309             </div>
310         );
311     }
312 }

```

```

313     );
314     if (mode === 'prepare')
315         return (
316             <div className="rounded-2xl border border-primary/15 bg-primary/5 p-5">
317                 <p className="flex items-center gap-2 font-semibold text-primary">
318                     <Sparkles className="size-4" /> Your visit-prep card
319                 </p>
320                 <p className="mt-3 text-sm text-muted-foreground">
321                     Bring this summary: "I'd like help with {concern}."
322                 </p>
323                 <ul className="mt-4 space-y-2 text-sm">
324                     {[
325                         'When did it start, and what makes it better or worse?',
326                         'What medications, allergies, or health conditions should I mention?',
327                         'What should I watch for, and when should I seek urgent help?',
328                         'What will the recommended care cost?',
329                     ].map((item) => (
330                         <li key={item} className="flex gap-2">
331                             <CheckCircle2 className="mt-0.5 size-4 shrink-0 text-primary" />{' '}
332                             {item}
333                         </li>
334                     ))}
335                 </ul>
336             </div>
337         );
338     return (
339         <div className="rounded-2xl border border-primary/15 bg-primary/5 p-5">
340             <p className="flex items-center gap-2 font-semibold text-primary">
341                 <ShieldCheck className="size-4" /> No emergency warning signs selected
342             </p>
343             <p className="mt-3 text-sm leading-6">
344                 For {concern}, consider calling your primary-care clinic or a nurse
345                 advice line. If you cannot get timely help, an urgent-care clinic may be
346                 appropriate. If symptoms become severe or a red flag appears, call 911.
347             </p>
348             <Button
349                 variant="outline"
350                 className="mt-4"
351                 render={
352                     <a
353                         href="https://findahealthcenter.hrsa.gov/"
354                         target="_blank"
355                         rel="noreferrer"
356                         aria-label="Find nearby care"
357                     />
358                 }
359             >
360                 <MapPin /> Find nearby care
361             </Button>
362         </div>
363     );
364 }

```

components.json

```
1 {
2   "$schema": "https://ui.shadcn.com/schema.json",
3   "style": "base-nova",
4   "rsc": true,
5   "tsx": true,
6   "tailwind": {
7     "config": "",
8     "css": "app/globals.css",
9     "baseColor": "neutral",
10    "cssVariables": true,
11    "prefix": ""
12  },
13  "iconLibrary": "lucide",
14  "rtl": false,
15  "aliases": {
16    "components": "@components",
17    "utils": "@lib/Utils",
18    "ui": "@components/ui",
19    "lib": "@lib",
20    "hooks": "@hooks"
21  },
22  "menuColor": "default",
23  "menuAccent": "subtle",
24  "registries": {}
25 }
```

components/ui/accordion.tsx

```
1 import { Accordion as AccordionPrimitive } from '@base-ui/react/accordion';
2
3 import { cn } from '@lib/utils';
4 import { ChevronDownIcon, ChevronUpIcon } from 'lucide-react';
5
6 function Accordion({ className, ...props }: AccordionPrimitive.Root.Props) {
7   return (
8     <AccordionPrimitive.Root
9       data-slot="accordion"
10      className={cn('flex w-full flex-col', className)}
11      {...props}
12    />
13  );
14 }
15
16 function AccordionItem({ className, ...props }: AccordionPrimitive.Item.Props) {
17   return (
18     <AccordionPrimitive.Item
19       data-slot="accordion-item"
20       className={cn('not-last:border-b', className)}
21       {...props}
22     />
23   );
24 }
25
26 function AccordionTrigger({
27   className,
28   children,
29   ...props
30 }: AccordionPrimitive.Trigger.Props) {
31   return (
32     <AccordionPrimitive.Header className="flex">
33       <AccordionPrimitive.Trigger
34         data-slot="accordion-trigger"
35         className={cn(
36           'focus-visible:ring-ring/50 focus-visible:border-ring focus-visible:after:border-ring
37           **:data-[slot=accordion-trigger-icon]:text-muted-foreground rounded-lg py-2.5 text-left text-sm font-
38           medium hover:underline focus-visible:ring-3 **:data-[slot=accordion-trigger-icon]:ml-auto
39           **:data-[slot=accordion-trigger-icon]:size-4 group/accordion-trigger relative flex flex-1 items-start
40           justify-between border border-transparent transition-all outline-none aria-disabled:pointer-events-none
41           aria-disabled:opacity-50',
42           className,
43         )}
44         {...props}
45       >
46         {children}
47         <ChevronDownIcon
48           data-slot="accordion-trigger-icon"
49           className="pointer-events-none shrink-0 group-aria-expanded/accordion-trigger:hidden"
50         />
51         <ChevronUpIcon
52           data-slot="accordion-trigger-icon"
53           className="pointer-events-none hidden shrink-0 group-aria-expanded/accordion-trigger:inline"
54         />
55       </AccordionPrimitive.Trigger>
56     </AccordionPrimitive.Header>
57   );
58 }
59
60 function AccordionContent({
61   className,
62   children,
63   ...props
64 }: AccordionPrimitive.Panel.Props) {
65   return (
66     <AccordionPrimitive.Panel
67       data-slot="accordion-content"
68       className="data-open:animate-accordion-down data-closed:animate-accordion-up text-sm overflow-
69       hidden"
70       {...props}
71     >
72       <div
73         className={cn(
74           'pt-0 pb-2.5 h-(--accordion-panel-height) data-ending-style:h-0 data-starting-style:h-0
75         [&a]:underline [&a]:underline-offset-3 [&a]:hover:text-foreground [&p:not(:last-child)]:mb-4',
76         className,
77       )}
78       >
79         {children}
80       </div>
81     </AccordionPrimitive.Panel>
82   );
83 }
```

```
76 }  
77  
78 export { Accordion, AccordionItem, AccordionTrigger, AccordionContent };
```

components/ui/alert-dialog.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { AlertDialog as AlertDialogPrimitive } from '@base-ui/react/alert-dialog';
5
6 import { cn } from '@lib/utlils';
7 import { Button } from '@components/ui/button';
8
9 function AlertDialog({ ...props }: AlertDialogPrimitive.Root.Props) {
10   return <AlertDialogPrimitive.Root data-slot="alert-dialog" {...props} />;
11 }
12
13 function AlertDialogTrigger({ ...props }: AlertDialogPrimitive.Trigger.Props) {
14   return (
15     <AlertDialogPrimitive.Trigger data-slot="alert-dialog-trigger" {...props} />
16   );
17 }
18
19 function AlertDialogPortal({ ...props }: AlertDialogPrimitive.Portal.Props) {
20   return (
21     <AlertDialogPrimitive.Portal data-slot="alert-dialog-portal" {...props} />
22   );
23 }
24
25 function AlertDialogOverlay({
26   className,
27   ...props
28 }: AlertDialogPrimitive.Backdrop.Props) {
29   return (
30     <AlertDialogPrimitive.Backdrop
31       data-slot="alert-dialog-overlay"
32       className={cn(
33         'data-open:animate-in data-closed:animate-out data-closed:fade-out-0 data-open:fade-in-0 bg-
34         black/10 duration-100 supports-backdrop-filter:backdrop-blur-xs fixed inset-0 isolate z-50',
35         className,
36       )}
37       {...props}
38     />
39   );
40 }
41
42 function AlertDialogContent({
43   className,
44   size = 'default',
45   ...props
46 }: AlertDialogPrimitive.Popup.Props & {
47   size?: 'default' | 'sm';
48 }) {
49   return (
50     <AlertDialogPortal>
51       <AlertDialogOverlay />
52       <AlertDialogPrimitive.Popup
53         data-slot="alert-dialog-content"
54         data-size={size}
55         className={cn(
56           'data-open:animate-in data-closed:animate-out data-closed:fade-out-0 data-open:fade-in-0 data-
57           closed:zoom-out-95 data-open:zoom-in-95 bg-popover text-popover-foreground ring-foreground/10 gap-4
58           rounded-xl p-4 ring-1 duration-100 data-[size=default]:max-w-xs data-[size=sm]:max-w-sm
59           data-[size=default]:sm:max-w-sm group/alert-dialog-content fixed top-1/2 left-1/2 z-50 grid w-full
60           -translate-x-1/2 -translate-y-1/2 outline-none',
61           className,
62         )}
63         {...props}
64       />
65     </AlertDialogPortal>
66   );
67 }
68
69 function AlertDialogHeader({
70   className,
71   ...props
72 }: React.ComponentProps<'div'>) {
73   return (
74     <div
75       data-slot="alert-dialog-header"
76       className={cn(
77         'grid grid-rows-[auto_1fr] place-items-center gap-1.5 text-center has-data-[slot=alert-dialog-
78         media]:grid-rows-[auto_auto_1fr] has-data-[slot=alert-dialog-media]:gap-x-4 sm:group-
79         data-[size=default]/alert-dialog-content:place-items-start sm:group-data-[size=default]/alert-dialog-
80         content:text-left sm:group-data-[size=default]/alert-dialog-content:has-data-[slot=alert-dialog-
81         media]:grid-rows-[auto_1fr]',
82         className,
```

```

74     })
75     {...props}
76   />
77 );
78 }
79
80 function AlertDialogFooter({
81   className,
82   ...props
83 }: React.ComponentProps<'div'>) {
84   return (
85     <div
86       data-slot="alert-dialog-footer"
87       className={cn(
88         'bg-muted/50 -mx-4 -mb-4 rounded-b-xl border-t p-4 flex flex-col-reverse gap-2 group-
data-[size=sm]/alert-dialog-content:grid group-data-[size=sm]/alert-dialog-content:grid-cols-2 sm:flex-
row sm:justify-end',
89       className,
90     )}
91     {...props}
92   />
93 );
94 }
95
96 function AlertDialogMedia({
97   className,
98   ...props
99 }: React.ComponentProps<'div'>) {
100   return (
101     <div
102       data-slot="alert-dialog-media"
103       className={cn(
104         'bg-muted mb-2 inline-flex size-10 items-center justify-center rounded-md sm:group-
data-[size=default]/alert-dialog-content:row-span-2 *:svg:not([class*="size-"]):size-6",
105       className,
106     )}
107     {...props}
108   />
109 );
110 }
111
112 function AlertDialogTitle({
113   className,
114   ...props
115 }: React.ComponentProps<typeof AlertDialogPrimitive.Title>) {
116   return (
117     <AlertDialogPrimitive.Title
118       data-slot="alert-dialog-title"
119       className={cn(
120         'text-base font-medium sm:group-data-[size=default]/alert-dialog-content:group-has-
data-[slot=alert-dialog-media]/alert-dialog-content:col-start-2 cn-font-heading',
121       className,
122     )}
123     {...props}
124   />
125 );
126 }
127
128 function AlertDialogDescription({
129   className,
130   ...props
131 }: React.ComponentProps<typeof AlertDialogPrimitive.Description>) {
132   return (
133     <AlertDialogPrimitive.Description
134       data-slot="alert-dialog-description"
135       className={cn(
136         'text-muted-foreground *:a:hover:text-foreground text-sm text-balance md:text-pretty
*:a:underline *:a:underline-offset-3',
137       className,
138     )}
139     {...props}
140   />
141 );
142 }
143
144 function AlertDialogAction({
145   className,
146   ...props
147 }: React.ComponentProps<typeof Button>) {
148   return (
149     <Button
150       data-slot="alert-dialog-action"
151       className={cn(className)}
152       {...props}
153     />

```

```
154   });
155 }
156
157 function AlertDialogCancel({
158   className,
159   variant = 'outline',
160   size = 'default',
161   ...props
162 }: AlertDialogPrimitive.Close.Props &
163   Pick<React.ComponentProps<typeof Button>, 'variant' | 'size'>) {
164   return (
165     <AlertDialogPrimitive.Close
166       data-slot="alert-dialog-cancel"
167       className={cn(className)}
168       render={<Button variant={variant} size={size} />}
169       {...props}
170     />
171   );
172 }
173
174 export {
175   AlertDialog,
176   AlertDialogAction,
177   AlertDialogCancel,
178   AlertDialogContent,
179   AlertDialogDescription,
180   AlertDialogFooter,
181   AlertDialogHeader,
182   AlertDialogMedia,
183   AlertDialogOverlay,
184   AlertDialogPortal,
185   AlertDialogTitle,
186   AlertDialogTrigger,
187 };
```

components/ui/alert.tsx

```
1 import * as React from 'react';
2 import { cva, type VariantProps } from 'class-variance-authority';
3
4 import { cn } from '@lib/utils';
5
6 const alertVariants = cva(
7   "group/alert relative grid w-full gap-0.5 rounded-lg border px-2.5 py-2 text-left text-sm has-
  data-[slot=alert-action]:relative has-data-[slot=alert-action]:pr-18 has-[>svg]:grid-cols-[auto_1fr]
  has-[>svg]:gap-x-2 *:[svg]:row-span-2 *:[svg]:translate-y-0.5 *:[svg]:text-current
  *:[svg]:not([class*='size-']):size-4",
8   {
9     variants: {
10       variant: {
11         default: 'bg-card text-card-foreground',
12         destructive:
13           'bg-card text-destructive *:data-[slot=alert-description]:text-destructive/90 *:[svg]:text-
  current',
14       },
15     },
16     defaultVariants: {
17       variant: 'default',
18     },
19   },
20 );
21
22 function Alert({
23   className,
24   variant,
25   ...props
26 }: React.ComponentProps<'div'> & VariantProps<typeof alertVariants>) {
27   return (
28     <div
29       data-slot="alert"
30       role="alert"
31       className={cn(alertVariants({ variant }), className)}
32       {...props}
33     />
34   );
35 }
36
37 function AlertTitle({ className, ...props }: React.ComponentProps<'div'>) {
38   return (
39     <div
40       data-slot="alert-title"
41       className={cn(
42         'font-medium group-has-[>svg]/alert:col-start-2 [&a]:underline [&a]:underline-offset-3
  [&a]:hover:text-foreground',
43         className,
44       )}
45       {...props}
46     />
47   );
48 }
49
50 function AlertDescription({
51   className,
52   ...props
53 }: React.ComponentProps<'div'>) {
54   return (
55     <div
56       data-slot="alert-description"
57       className={cn(
58         'text-sm text-balance text-muted-foreground md:text-pretty [&a]:underline [&a]:underline-
  offset-3 [&a]:hover:text-foreground [&p]:not(:last-child):mb-4',
59         className,
60       )}
61       {...props}
62     />
63   );
64 }
65
66 function AlertAction({ className, ...props }: React.ComponentProps<'div'>) {
67   return (
68     <div
69       data-slot="alert-action"
70       className={cn('absolute top-2 right-2', className)}
71       {...props}
72     />
73   );
74 }
75
76 export { Alert, AlertTitle, AlertDescription, AlertAction };
```

components/ui/aspect-ratio.tsx

```
1 import { cn } from '@lib/utils';
2
3 function AspectRatio({
4   ratio,
5   className,
6   ...props
7 }: React.ComponentProps<'div'> & { ratio: number }) {
8   return (
9     <div
10       data-slot="aspect-ratio"
11       style={
12         {
13           '--ratio': ratio,
14         } as React.CSSProperties
15       >
16       <div
17         className={cn('relative aspect-(--ratio)', className)}
18         {...props}
19       />
20     </div>
21   );
22 }
23
24 export { AspectRatio };
```

components/ui/attachment.tsx

```
1 import * as React from 'react';
2 import { mergeProps } from '@base-ui/react/merge-props';
3 import { useRender } from '@base-ui/react/use-render';
4 import { cva, type VariantProps } from 'class-variance-authority';
5
6 import { cn } from '@lib/utils';
7 import { Button } from '@components/ui/button';
8
9 const attachmentVariants = cva(
10   'rounded-xl w-fit focus-within:ring-1 focus-within:ring-ring/50 group/attachment relative flex max-w-full min-w-0 shrink-0 flex-wrap border bg-card text-card-foreground transition-colors has->a,>button:hover:bg-muted/50 data-[state=error]:border-destructive/30 data-[state=idle]:border-dashed',
11   {
12     variants: {
13       size: {
14         default:
15           'gap-2 has-data-[slot=attachment-content]:px-2.5 has-data-[slot=attachment-content]:py-2 has-data-[slot=attachment-media]:p-2 text-sm',
16         sm: 'gap-2.5 has-data-[slot=attachment-content]:px-2 has-data-[slot=attachment-content]:py-1.5 has-data-[slot=attachment-media]:p-1.5 text-xs',
17         xs: 'gap-1.5 has-data-[slot=attachment-content]:px-1.5 has-data-[slot=attachment-content]:py-1 has-data-[slot=attachment-media]:p-1 text-xs rounded-lg',
18       },
19       orientation: {
20         horizontal: 'min-w-40 items-center',
21         vertical: 'w-24 has-data-[slot=attachment-content]:w-30 flex-col',
22       },
23     },
24   },
25 );
26
27 function Attachment({
28   className,
29   state = 'done',
30   size = 'default',
31   orientation = 'horizontal',
32   ...props
33 }: React.ComponentProps<'div'> &
34   VariantProps<typeof attachmentVariants> & {
35     state?: 'idle' | 'uploading' | 'processing' | 'error' | 'done';
36   }) {
37   return (
38     <div
39       data-slot="attachment"
40       data-state={state}
41       data-size={size}
42       data-orientation={orientation}
43       className={cn(attachmentVariants({ size, orientation }), className)}
44       {...props}
45     />
46   );
47 }
48
49 const attachmentMediaVariants = cva(
50   'bg-muted text-foreground w-10 rounded-lg group-data-[size=sm]/attachment:w-8 group-data-[size=xs]/attachment:w-7 group-data-[size=xs]/attachment:rounded-md [&_svg:not([class*="size-"])]>size-4 group-data-[size=xs]/attachment:[&_svg:not([class*="size-"])]>size-3.5 group-data-[orientation=vertical]/attachment:w-full group-data-[orientation=vertical]/attachment:[&_svg:not([class*="size-"])]>size-6 group-data-[orientation=vertical]/attachment:*.data-[slot=spinner]:size-6! relative flex aspect-square shrink-0 items-center justify-center overflow-hidden group-data-[state=error]/attachment:bg-destructive/10 group-data-[state=error]/attachment:text-destructive [&_svg]:pointer-events-none',
51   {
52     variants: {
53       variant: {
54         icon: '',
55         image:
56           'opacity-60 group-data-[state=idle]/attachment:opacity-100 group-data-[state=done]/attachment:opacity-100 *:img:aspect-square *:img:w-full *:img:object-cover',
57       },
58     },
59     defaultVariants: {
60       variant: 'icon',
61     },
62   },
63 );
64
65 function AttachmentMedia({
66   className,
67   variant = 'icon',
```

```

68   ...props
69 }: React.ComponentProps<'div'> & VariantProps<typeof attachmentMediaVariants>) {
70   return (
71     <div
72       data-slot="attachment-media"
73       data-variant={variant}
74       className={cn(attachmentMediaVariants({ variant }), className)}
75       {...props}
76     />
77   );
78 }
79
80 function AttachmentContent({
81   className,
82   ...props
83 }: React.ComponentProps<'div'>) {
84   return (
85     <div
86       data-slot="attachment-content"
87       className={cn(
88         'leading-tight group-data-[orientation=vertical]/attachment:px-1 max-w-full min-w-0 flex-1',
89         className,
90       )}
91       {...props}
92     />
93   );
94 }
95
96 function AttachmentTitle({
97   className,
98   ...props
99 }: React.ComponentProps<'span'>) {
100   return (
101     <span
102       data-slot="attachment-title"
103       className={cn(
104         'font-medium block max-w-full min-w-0 truncate group-data-[state=processing]/attachment:shimmer
group-data-[state=uploading]/attachment:shimmer',
105         className,
106       )}
107       {...props}
108     />
109   );
110 }
111
112 function AttachmentDescription({
113   className,
114   ...props
115 }: React.ComponentProps<'span'>) {
116   return (
117     <span
118       data-slot="attachment-description"
119       className={cn(
120         'mt-0.5 text-xs block min-w-0 truncate text-muted-foreground group-
data-[state=error]/attachment:text-destructive/80',
121         'max-w-full',
122         className,
123       )}
124       {...props}
125     />
126   );
127 }
128
129 function AttachmentActions({
130   className,
131   ...props
132 }: React.ComponentProps<'div'>) {
133   return (
134     <div
135       data-slot="attachment-actions"
136       className={cn(
137         'group-data-[orientation=vertical]/attachment:absolute group-
data-[orientation=vertical]/attachment:top-3 group-data-[orientation=vertical]/attachment:right-3
relative z-20 group-data-[orientation=vertical]/attachment:gap-1 flex shrink-0 items-center',
138         className,
139       )}
140       {...props}
141     />
142   );
143 }
144
145 function AttachmentAction({
146   className,
147   variant,
148   size = 'icon-xs',

```

```

149   ...props
150 }; React.ComponentProps<typeof Button>) {
151   return (
152     <Button
153       data-slot="attachment-action"
154       variant={variant ?? 'ghost'}
155       size={size}
156       className={cn(className)}
157       {...props}
158     />
159   );
160 }
161
162 function AttachmentTrigger({
163   className,
164   render,
165   type,
166   ...props
167 }: useRender.ComponentProps<'button'>) {
168   return useRender({
169     defaultTagName: 'button',
170     props: mergeProps<'button'>({
171       {
172         type: render ? type : (type ?? 'button'),
173         className: cn('absolute inset-0 z-10 outline-none', className),
174       },
175       props,
176     },
177     render,
178     state: {
179       slot: 'attachment-trigger',
180     },
181   });
182 }
183
184 function AttachmentGroup({ className, ...props }: React.ComponentProps<'div'>) {
185   return (
186     <div
187       data-slot="attachment-group"
188       className={cn(
189         'gap-3 scroll-py-1 py-1 flex min-w-0 scroll-fade-x snap-x snap-mandatory scrollbar-none
190         overflow-x-auto overscroll-x-contain *:data-[slot=attachment]:flex-none *:data-[slot=attachment]:snap-
191         start',
192         className,
193       )}
194     />
195   );
196 }
197 export {
198   Attachment,
199   AttachmentGroup,
200   AttachmentMedia,
201   AttachmentContent,
202   AttachmentTitle,
203   AttachmentDescription,
204   AttachmentActions,
205   AttachmentAction,
206   AttachmentTrigger,
207 };

```

components/ui/avatar.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { Avatar as AvatarPrimitive } from '@base-ui/react/avatar';
5
6 import { cn } from '@lib/utils';
7
8 function Avatar({
9   className,
10   size = 'default',
11   ...props
12 }: AvatarPrimitive.Root.Props & {
13   size?: 'default' | 'sm' | 'lg';
14 }) {
15   return (
16     <AvatarPrimitive.Root
17       data-slot="avatar"
18       data-size={size}
19       className={cn(
20         'size-8 rounded-full after:rounded-full data-[size=lg]:size-10 data-[size=sm]:size-6
21         group/avatar relative flex shrink-0 select-none after:absolute after:inset-0 after:border after:border-
22         border after:mix-blend-darken dark:after:mix-blend-lighten',
23         className,
24       )}
25     >
26     {...props}
27   </>
28 );
29
30 function AvatarImage({ className, ...props }: AvatarPrimitive.Image.Props) {
31   return (
32     <AvatarPrimitive.Image
33       data-slot="avatar-image"
34       className={cn(
35         'rounded-full aspect-square size-full object-cover',
36         className,
37       )}
38     >
39     {...props}
40   </>
41 );
42
43 function AvatarFallback({
44   className,
45   ...props
46 }: AvatarPrimitive.Fallback.Props) {
47   return (
48     <AvatarPrimitive.Fallback
49       data-slot="avatar-fallback"
50       className={cn(
51         'bg-muted text-muted-foreground rounded-full flex size-full items-center justify-center text-sm
52         group-data-[size=sm]/avatar:text-xs',
53         className,
54       )}
55     >
56     {...props}
57   </>
58 );
59
60 function AvatarBadge({ className, ...props }: React.ComponentProps<'span'>) {
61   return (
62     <span
63       data-slot="avatar-badge"
64       className={cn(
65         'bg-primary text-primary-foreground ring-background absolute right-0 bottom-0 z-10 inline-flex
66         items-center justify-center rounded-full bg-blend-color ring-2 select-none',
67         'group-data-[size=sm]/avatar:size-2 group-data-[size=sm]/avatar:[&svg]:hidden',
68         'group-data-[size=default]/avatar:size-2.5 group-data-[size=default]/avatar:[&svg]:size-2',
69         'group-data-[size=lg]/avatar:size-3 group-data-[size=lg]/avatar:[&svg]:size-2',
70         className,
71       )}
72     >
73     {...props}
74   </>
75 );
76
77 function AvatarGroup({ className, ...props }: React.ComponentProps<'div'>) {
78   return (
79     <div
80       data-slot="avatar-group"
81       className={cn(
82         'group/avatar-group flex -space-x-2 *:data-[slot=avatar]:ring-2 *:data-[slot=avatar]:ring-
```

```

    background',
79     className,
80   })
81   {...props}
82   />
83 );
84 }
85
86 function AvatarGroupCount({
87   className,
88   ...props
89 }: React.ComponentProps<'div'>) {
90   return (
91     <div
92       data-slot="avatar-group-count"
93       className={cn(
94         'bg-muted text-muted-foreground size-8 rounded-full text-sm group-has-data-[size=lg]/avatar-
group:size-10 group-has-data-[size=sm]/avatar-group:size-6 [&svg]:size-4 group-has-
data-[size=lg]/avatar-group:[&svg]:size-5 group-has-data-[size=sm]/avatar-group:[&svg]:size-3 relative
flex shrink-0 items-center justify-center ring-2 ring-background',
95       className,
96     )}
97     {...props}
98   />
99 );
100 }
101
102 export {
103   Avatar,
104   AvatarImage,
105   AvatarFallback,
106   AvatarGroup,
107   AvatarGroupCount,
108   AvatarBadge,
109 };

```

components/ui/badge.tsx

```
1 import { mergeProps } from '@base-ui/react/merge-props';
2 import { useRender } from '@base-ui/react/use-render';
3 import { cva, type VariantProps } from 'class-variance-authority';
4
5 import { cn } from '@lib/utils';
6
7 const badgeVariants = cva(
8   'group/badge inline-flex h-5 w-fit shrink-0 items-center justify-center gap-1 overflow-hidden
9   rounded-4xl border border-transparent px-2 py-0.5 text-xs font-medium whitespace-nowrap transition-all
10  focus-visible:border-ring focus-visible:ring-[3px] focus-visible:ring-ring/50 has-data-[icon=inline-
11  end]:pr-1.5 has-data-[icon=inline-start]:pl-1.5 aria-invalid:border-destructive aria-invalid:ring-
12  destructive/20 dark:aria-invalid:ring-destructive/40 [&svg]:pointer-events-none [&svg]:size-3!',
13  {
14    variants: {
15      variant: {
16        default: 'bg-primary text-primary-foreground [a]:hover:bg-primary/80',
17        secondary: 'bg-secondary text-secondary-foreground [a]:hover:bg-secondary/80',
18        destructive: 'bg-destructive/10 text-destructive focus-visible:ring-destructive/20 dark:bg-destructive/20
19        dark:focus-visible:ring-destructive/40 [a]:hover:bg-destructive/20',
20        outline: 'border-border text-foreground [a]:hover:bg-muted [a]:hover:text-muted-foreground',
21        ghost: 'hover:bg-muted hover:text-muted-foreground dark:hover:bg-muted/50',
22        link: 'text-primary underline-offset-4 hover:underline',
23      },
24    },
25    defaultVariants: {
26      variant: 'default',
27    },
28  },
29 );
30
31 function Badge({
32   className,
33   variant = 'default',
34   render,
35   ...props
36 }: useRender.ComponentProps<'span'> & VariantProps<typeof badgeVariants>) {
37   return useRender({
38     defaultTagName: 'span',
39     props: mergeProps<'span'>({
40       className: cn(badgeVariants({ variant }), className),
41     }, props),
42     render,
43     state: {
44       slot: 'badge',
45       variant,
46     },
47   });
48 }
49
50 export { Badge, badgeVariants };
```

components/ui/breadcrumb.tsx

```
1 import * as React from 'react';
2 import { mergeProps } from '@base-ui/react/merge-props';
3 import { useRender } from '@base-ui/react/use-render';
4
5 import { cn } from '@lib/utils';
6 import { ChevronRightIcon, MoreHorizontalIcon } from 'lucide-react';
7
8 function Breadcrumb({ className, ...props }: React.ComponentProps<'nav'>) {
9   return (
10     <nav
11       aria-label="breadcrumb"
12       data-slot="breadcrumb"
13       className={cn(className)}
14       {...props}
15     />
16   );
17 }
18
19 function BreadcrumbList({ className, ...props }: React.ComponentProps<'ol'>) {
20   return (
21     <ol
22       data-slot="breadcrumb-list"
23       className={cn(
24         'text-muted-foreground gap-1.5 text-sm flex flex-wrap items-center wrap-break-word',
25         className,
26       )}
27       {...props}
28     />
29   );
30 }
31
32 function BreadcrumbItem({ className, ...props }: React.ComponentProps<'li'>) {
33   return (
34     <li
35       data-slot="breadcrumb-item"
36       className={cn('gap-1 inline-flex items-center', className)}
37       {...props}
38     />
39   );
40 }
41
42 function BreadcrumbLink({
43   className,
44   render,
45   ...props
46 }: useRender.ComponentProps<'a'>) {
47   return useRender({
48     defaultTagName: 'a',
49     props: mergeProps<'a'>({
50       {
51         className: cn('hover:text-foreground transition-colors', className),
52       },
53       props,
54     },
55     render,
56     state: {
57       slot: 'breadcrumb-link',
58     },
59   });
60 }
61
62 function BreadcrumbPage({ className, ...props }: React.ComponentProps<'span'>) {
63   return (
64     <span
65       data-slot="breadcrumb-page"
66       role="link"
67       aria-disabled="true"
68       aria-current="page"
69       className={cn('text-foreground font-normal', className)}
70       {...props}
71     />
72   );
73 }
74
75 function BreadcrumbSeparator({
76   children,
77   className,
78   ...props
79 }: React.ComponentProps<'li'>) {
80   return (
81     <li
82       data-slot="breadcrumb-separator"
```

```

83     role="presentation"
84     aria-hidden="true"
85     className={cn(['&svg]:size-3.5', className)}
86     {...props}
87   >
88     {children ?? <ChevronRightIcon className="cn-rtl-flip" />}
89   </li>
90 );
91 }
92
93 function BreadcrumbEllipsis({
94   className,
95   ...props
96 }: React.ComponentProps<'span'>) {
97   return (
98     <span
99       data-slot="breadcrumb-ellipsis"
100       role="presentation"
101       aria-hidden="true"
102       className={cn(
103         'size-5 [&svg]:size-4 flex items-center justify-center',
104         className,
105       )}
106       {...props}
107     >
108       <MoreHorizontalIcon />
109       <span className="sr-only">More</span>
110     </span>
111   );
112 }
113
114 export {
115   Breadcrumb,
116   BreadcrumbList,
117   BreadcrumbItem,
118   BreadcrumbLink,
119   BreadcrumbPage,
120   BreadcrumbSeparator,
121   BreadcrumbEllipsis,
122 };

```

components/ui/bubble.tsx

```
1 import * as React from 'react';
2 import { mergeProps } from '@base-ui/react/merge-props';
3 import { useRender } from '@base-ui/react/use-render';
4 import { cva, type VariantProps } from 'class-variance-authority';
5
6 import { cn } from '@lib/utils';
7
8 function BubbleGroup({ className, ...props }: React.ComponentProps<'div'>) {
9   return (
10     <div
11       data-slot="bubble-group"
12       className={cn('gap-2 flex min-w-0 flex-col', className)}
13       {...props}
14     />
15   );
16 }
17
18 const bubbleVariants = cva(
19   'gap-1 data-[align=end]:self-end max-w-[80%] data-[variant=ghost]:max-w-full group-
20     data-[align=end]/message:self-end group/bubble relative flex w-fit min-w-0 flex-col',
21   {
22     variants: {
23       variant: {
24         default:
25           '*:data-[slot=bubble-content]:bg-primary *:data-[slot=bubble-content]:text-primary-foreground
26           [&>[data-slot=bubble-content]:is(button,a):hover]:bg-primary/80',
27         secondary:
28           '*:data-[slot=bubble-content]:bg-secondary *:data-[slot=bubble-content]:text-secondary-
29           foreground [&>[data-slot=bubble-content]:is(button,a):hover]:bg-[color-
30           mix(in_oklch,var(--secondary),var(--foreground)_5%)]',
31         muted:
32           '*:data-[slot=bubble-content]:bg-muted [&>[data-slot=bubble-
33           content]:is(button,a):hover]:bg-[color-mix(in_oklch,var(--muted),var(--foreground)_5%)]',
34         tinted:
35           '*:data-[slot=bubble-content]:bg-[oklch(from_var(--primary)_0.93_calc(c*0.4)_h)]
36           dark:*:data-[slot=bubble-content]:bg-[oklch(from_var(--primary)_0.3_calc(c*0.4)_h)] *:data-[slot=bubble-
37           content]:text-foreground [&>[data-slot=bubble-
38           content]:is(button,a):hover]:bg-[oklch(from_var(--primary)_0.88_calc(c*0.5)_h)] dark:[&>[data-
39           slot=bubble-content]:is(button,a):hover]:bg-[oklch(from_var(--primary)_0.35_calc(c*0.5)_h)]',
40         outline:
41           '*:data-[slot=bubble-content]:bg-background *:data-[slot=bubble-content]:border-border
42           [&>[data-slot=bubble-content]:is(button,a):hover]:bg-muted [&>[data-slot=bubble-
43           content]:is(button,a):hover]:text-foreground dark:[&>[data-slot=bubble-content]:is(button,a):hover]:bg-
44           input/30',
45         ghost:
46           '*:data-[slot=bubble-content]:rounded-none *:data-[slot=bubble-content]:bg-transparent
47           *:data-[slot=bubble-content]:p-0 [&>[data-slot=bubble-content]:is(button,a):hover]:bg-muted [&>[data-
48           slot=bubble-content]:is(button,a):hover]:text-foreground dark:[&>[data-slot=bubble-
49           content]:is(button,a):hover]:bg-muted/50 border-none',
50         destructive:
51           '*:data-[slot=bubble-content]:bg-destructive/10 dark:*:data-[slot=bubble-content]:bg-
52           destructive/20 *:data-[slot=bubble-content]:text-destructive [&>[data-slot=bubble-
53           content]:is(button,a):hover]:bg-destructive/20 dark:[&>[data-slot=bubble-
54           content]:is(button,a):hover]:bg-destructive/30',
55       },
56     },
57     defaultVariants: {
58       variant: 'default',
59     },
60   },
61 );
62
63 function Bubble({
64   variant = 'default',
65   align = 'start',
66   className,
67   ...props
68 }: React.ComponentProps<'div'> &
69   VariantProps<typeof bubbleVariants> & {
70   align?: 'start' | 'end';
71 }) {
72   return (
73     <div
74       data-slot="bubble"
75       data-variant={variant}
76       data-align={align}
77       className={cn(bubbleVariants({ variant }), className)}
78       {...props}
79     />
80   );
81 }
82 }
```

```

65 function BubbleContent({
66   className,
67   render,
68   ...props
69 }: useRender.ComponentProps<'div'>) {
70   return useRender({
71     defaultTagName: 'div',
72     props: mergeProps<'div'>({
73       {
74         className: cn(
75           'rounded-xl border border-transparent px-3 py-2 text-sm leading-relaxed [button,a]:outline-
none [button,a]:focus-visible:border-ring [button,a]:focus-visible:ring-3 [button,a]:focus-visible:ring-
ring/50 group-data-[align=end]/bubble:self-end w-fit max-w-full min-w-0 overflow-hidden wrap-break-word
[button]:text-left [button,a]:transition-colors',
76         className,
77       ),
78     },
79     props,
80   ),
81   render,
82   state: {
83     slot: 'bubble-content',
84   },
85 });
86 }
87
88 const bubbleReactionsVariants = cva(
89   'rounded-full ring-3 ring-card bg-muted shrink-0 gap-1 px-1.5 py-0.5 has-[button]:p-0 text-sm absolute
z-10 flex w-fit items-center justify-center',
90   {
91     variants: {
92       side: {
93         top: 'top-0 -translate-y-3/4',
94         bottom: 'bottom-0 translate-y-3/4',
95       },
96       align: {
97         start: 'left-3',
98         end: 'right-3',
99       },
100     },
101     defaultVariants: {
102       side: 'bottom',
103       align: 'end',
104     },
105   },
106 );
107
108 function BubbleReactions({
109   side = 'bottom',
110   align = 'end',
111   className,
112   ...props
113 }: React.ComponentProps<'div'> & {
114   align?: 'start' | 'end';
115   side?: 'top' | 'bottom';
116 }) {
117   return (
118     <div
119       data-slot="bubble-reactions"
120       data-align={align}
121       data-side={side}
122       className={cn(bubbleReactionsVariants({ side, align }), className)}
123       {...props}
124     />
125   );
126 }
127
128 export { BubbleGroup, Bubble, BubbleContent, BubbleReactions };

```

components/ui/button-group.tsx

```
1 import { mergeProps } from '@base-ui/react/merge-props';
2 import { useRender } from '@base-ui/react/use-render';
3 import { cva, type VariantProps } from 'class-variance-authority';
4
5 import { cn } from '@lib/utils';
6 import { Separator } from '@components/ui/separator';
7
8 const buttonGroupVariants = cva(
9   "has-[>[data-slot=button-group]]:gap-2 has-[select[aria-hidden=true]:last-child]:[&[data-slot=select-trigger]:last-of-type]:rounded-r-lg flex w-fit items-stretch *:focus-visible:relative *:focus-visible:z-10 [&[data-slot=select-trigger]:not([class*='w-']]):w-fit [&input]:flex-1",
10  {
11    variants: {
12      orientation: {
13        horizontal:
14          '[&[data-slot]:not(:has(~[data-slot]))]:rounded-r-lg! *:data-slot:rounded-r-none [&[data-slot]~[data-slot]]:rounded-l-none [&[data-slot]~[data-slot]]:border-l-0',
15        vertical:
16          '[&[data-slot]:not(:has(~[data-slot]))]:rounded-b-lg! flex-col *:data-slot:rounded-b-none [&[data-slot]~[data-slot]]:rounded-t-none [&[data-slot]~[data-slot]]:border-t-0',
17      },
18    },
19    defaultVariants: {
20      orientation: 'horizontal',
21    },
22  },
23 );
24
25 function ButtonGroup({
26   className,
27   orientation,
28   ...props
29 }: React.ComponentProps<'div'> & VariantProps<typeof buttonGroupVariants>) {
30   return (
31     <div
32       role="group"
33       data-slot="button-group"
34       data-orientation={orientation}
35       className={cn(buttonGroupVariants({ orientation }), className)}
36       {...props}
37     />
38   );
39 }
40
41 function ButtonGroupText({
42   className,
43   render,
44   ...props
45 }: useRender.ComponentProps<'div'>) {
46   return useRender({
47     defaultTagName: 'div',
48     props: mergeProps<'div'>({
49       {
50         className: cn(
51           "bg-muted gap-2 rounded-lg border px-2.5 text-sm font-medium
52           [&_svg:not([class*='size-']):size-4 flex items-center [&_svg]:pointer-events-none",
53           className,
54         ),
55       },
56       props,
57     ),
58     render,
59     state: {
60       slot: 'button-group-text',
61     },
62   });
63 }
64
65 function ButtonGroupSeparator({
66   className,
67   orientation = 'vertical',
68   ...props
69 }: React.ComponentProps<typeof Separator>) {
70   return (
71     <Separator
72       data-slot="button-group-separator"
73       orientation={orientation}
74       className={cn(
75         "bg-input relative self-stretch data-horizontal:mx-px data-horizontal:w-auto data-vertical:my-px
76         data-vertical:h-auto",
77         className,
78       )}
79     />
80   );
81 }
```

```
77     {...props}
78   />
79 );
80 }
81
82 export {
83   ButtonGroup,
84   ButtonGroupSeparator,
85   ButtonGroupText,
86   buttonGroupVariants,
87 };
```

components/ui/button.tsx

```
1 import { Button as ButtonPrimitive } from '@base-ui/react/button';
2 import { cva, type VariantProps } from 'class-variance-authority';
3
4 import { cn } from '@lib/utlils';
5
6 const buttonVariants = cva(
7   "group/button inline-flex shrink-0 items-center justify-center rounded-lg border border-transparent
  bg-clip-padding text-sm font-medium whitespace-nowrap transition-all outline-none select-none focus-
  visible:border-ring focus-visible:ring-3 focus-visible:ring-ring/50 active:not-
  aria-[haspopup]:translate-y-px disabled:pointer-events-none disabled:opacity-50 aria-invalid:border-
  destructive aria-invalid:ring-3 aria-invalid:ring-destructive/20 dark:aria-invalid:border-destructive/50
  dark:aria-invalid:ring-destructive/40 [&_svg]:pointer-events-none [&_svg]:shrink-0
  [&_svg:not([class*='size-']):size-4",
8   {
9     variants: {
10       variant: {
11         default: 'bg-primary text-primary-foreground hover:bg-primary/80',
12         outline:
13           'border-border bg-background hover:bg-muted hover:text-foreground aria-expanded:bg-muted aria-
  expanded:text-foreground dark:border-input dark:bg-input/30 dark:hover:bg-input/50',
14         secondary:
15           'bg-secondary text-secondary-foreground hover:bg-[color-
  mix(in_oklch,var(--secondary),var(--foreground)_5%)] aria-expanded:bg-secondary aria-expanded:text-
  secondary-foreground',
16         ghost:
17           'hover:bg-muted hover:text-foreground aria-expanded:bg-muted aria-expanded:text-foreground
  dark:hover:bg-muted/50',
18         destructive:
19           'bg-destructive/10 text-destructive hover:bg-destructive/20 focus-visible:border-
  destructive/40 focus-visible:ring-destructive/20 dark:bg-destructive/20 dark:hover:bg-destructive/30
  dark:focus-visible:ring-destructive/40',
20         link: 'text-primary underline-offset-4 hover:underline',
21       },
22       size: {
23         default:
24           'h-8 gap-1.5 px-2.5 has-data-[icon=inline-end]:pr-2 has-data-[icon=inline-start]:pl-2',
25         xs: 'h-6 gap-1 rounded-[min(var(--radius-md),10px)] px-2 text-xs in-data-[slot=button-
  group]:rounded-lg has-data-[icon=inline-end]:pr-1.5 has-data-[icon=inline-start]:pl-1.5
  [&_svg:not([class*='size-']):size-3",
26         sm: 'h-7 gap-1 rounded-[min(var(--radius-md),12px)] px-2.5 text-[0.8rem] in-data-[slot=button-
  group]:rounded-lg has-data-[icon=inline-end]:pr-1.5 has-data-[icon=inline-start]:pl-1.5
  [&_svg:not([class*='size-']):size-3.5",
27         lg: 'h-9 gap-1.5 px-2.5 has-data-[icon=inline-end]:pr-2 has-data-[icon=inline-start]:pl-2',
28         icon: 'size-8',
29         'icon-xs':
30           "size-6 rounded-[min(var(--radius-md),10px)] in-data-[slot=button-group]:rounded-lg
  [&_svg:not([class*='size-']):size-3",
31         'icon-sm':
32           "size-7 rounded-[min(var(--radius-md),12px)] in-data-[slot=button-group]:rounded-lg",
33         'icon-lg': 'size-9',
34       },
35     },
36     defaultVariants: {
37       variant: 'default',
38       size: 'default',
39     },
40   },
41 );
42
43 function Button({
44   className,
45   variant = 'default',
46   size = 'default',
47   ...props
48 }: ButtonPrimitive.Props & VariantProps<typeof buttonVariants>) {
49   return (
50     <ButtonPrimitive
51       data-slot="button"
52       className={cn(buttonVariants({ variant, size, className }))}
53       {...props}
54     />
55   );
56 }
57
58 export { Button, buttonVariants };
```

components/ui/calendar.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import {
5   DayPicker,
6   getDefaultClassNames,
7   type DayButton,
8   type Locale,
9 } from 'react-day-picker';
10
11 import { cn } from '@lib/Utils';
12 import { Button, buttonVariants } from '@components/ui/button';
13 import {
14   ChevronLeftIcon,
15   ChevronRightIcon,
16   ChevronDownIcon,
17 } from 'lucide-react';
18
19 function Calendar({
20   className,
21   classNames,
22   showOutsideDays = true,
23   captionLayout = 'label',
24   buttonVariant = 'ghost',
25   locale,
26   formatters,
27   components,
28   ...props
29 }: React.ComponentProps<typeof DayPicker> & {
30   buttonVariant?: React.ComponentProps<typeof Button>['variant'];
31 }) {
32   const defaultClassNames = getDefaultClassNames();
33
34   return (
35     <DayPicker
36       showOutsideDays={showOutsideDays}
37       className={cn(
38         'p-2 [--cell-radius:var(--radius-md)] [--cell-size:--spacing(7)] group/calendar bg-background
in-data-[slot=card-content]:bg-transparent in-data-[slot=popover-content]:bg-transparent',
39         String.raw`rtl:*. [.rdp-button\_next>svg]:rotate-180`,
40         String.raw`rtl:*. [.rdp-button\_previous>svg]:rotate-180`,
41         className,
42       )}
43       captionLayout={captionLayout}
44       locale={locale}
45       formatters={{
46         formatMonthDropdown: (date) =>
47           date.toLocaleString(locale?.code, { month: 'short' }),
48         ...formatters,
49       }}
50       classNames={{
51         root: cn('w-fit', defaultClassNames.root),
52         months: cn(
53           'relative flex flex-col gap-4 md:flex-row',
54           defaultClassNames.months,
55         ),
56         month: cn('flex w-full flex-col gap-4', defaultClassNames.month),
57         nav: cn(
58           'absolute inset-x-0 top-0 flex w-full items-center justify-between gap-1',
59           defaultClassNames.nav,
60         ),
61         button_previous: cn(
62           buttonVariants({ variant: buttonVariant }),
63           'size-(--cell-size) p-0 select-none aria-disabled:opacity-50',
64           defaultClassNames.button_previous,
65         ),
66         button_next: cn(
67           buttonVariants({ variant: buttonVariant }),
68           'size-(--cell-size) p-0 select-none aria-disabled:opacity-50',
69           defaultClassNames.button_next,
70         ),
71         month_caption: cn(
72           'flex h-(--cell-size) w-full items-center justify-center px-(--cell-size)',
73           defaultClassNames.month_caption,
74         ),
75         dropdowns: cn(
76           'flex h-(--cell-size) w-full items-center justify-center gap-1.5 text-sm font-medium',
77           defaultClassNames.dropdowns,
78         ),
79         dropdown_root: cn(
80           'cn-calendar-dropdown-root relative rounded-(--cell-radius)',
81           defaultClassNames.dropdown_root,
```

```

82     ),
83     dropdown: cn(
84       'absolute inset-0 bg-popover opacity-0',
85       defaultClassNames.dropdown,
86     ),
87     caption_label: cn(
88       'font-medium select-none',
89       captionLayout === 'label'
90       ? 'cn-calendar-caption text-sm'
91       : 'cn-calendar-caption-label flex items-center gap-1 rounded-(--cell-radius) text-sm
[&>svg]:size-3.5 [&>svg]:text-muted-foreground',
92       defaultClassNames.caption_label,
93     ),
94     month_grid: cn('w-full border-collapse', defaultClassNames.month_grid),
95     weekdays: cn('flex', defaultClassNames.weekdays),
96     weekday: cn(
97       'flex-1 rounded-(--cell-radius) text-[0.8rem] font-normal text-muted-foreground select-none',
98       defaultClassNames.weekday,
99     ),
100    week: cn('mt-2 flex w-full', defaultClassNames.week),
101    week_number_header: cn(
102      'w-(--cell-size) select-none',
103      defaultClassNames.week_number_header,
104    ),
105    week_number: cn(
106      'text-[0.8rem] text-muted-foreground select-none',
107      defaultClassNames.week_number,
108    ),
109    day: cn(
110      'group/day relative aspect-square h-full w-full rounded-(--cell-radius) p-0 text-center
select-none [&:last-child[data-selected=true]_button]:rounded-r-(--cell-radius)',
111      props.showWeekNumber
112      ? ' [&:nth-child(2)[data-selected=true]_button]:rounded-l-(--cell-radius)'
113      : ' [&:first-child[data-selected=true]_button]:rounded-l-(--cell-radius)',
114      defaultClassNames.day,
115    ),
116    range_start: cn(
117      'relative isolate z-0 rounded-l-(--cell-radius) bg-muted after:absolute after:inset-y-0
after:right-0 after:w-4 after:bg-muted',
118      defaultClassNames.range_start,
119    ),
120    range_middle: cn('rounded-none', defaultClassNames.range_middle),
121    range_end: cn(
122      'relative isolate z-0 rounded-r-(--cell-radius) bg-muted after:absolute after:inset-y-0
after:left-0 after:w-4 after:bg-muted',
123      defaultClassNames.range_end,
124    ),
125    today: cn(
126      'rounded-(--cell-radius) bg-muted text-foreground data-[selected=true]:rounded-none',
127      defaultClassNames.today,
128    ),
129    outside: cn(
130      'text-muted-foreground aria-selected:text-muted-foreground',
131      defaultClassNames.outside,
132    ),
133    disabled: cn(
134      'text-muted-foreground opacity-50',
135      defaultClassNames.disabled,
136    ),
137    hidden: cn('invisible', defaultClassNames.hidden),
138    ...classNames,
139  })
140  components={{
141    Root: ({ className, rootRef, ...props }) => {
142      return (
143        <div
144          data-slot="calendar"
145          ref={rootRef}
146          className={cn(className)}
147          {...props}
148        />
149      );
150    },
151    Chevron: ({ className, orientation, ...props }) => {
152      if (orientation === 'left') {
153        return (
154          <ChevronLeftIcon
155            className={cn('cn-rtl-flip size-4', className)}
156            {...props}
157          />
158        );
159      }
160    },
161    if (orientation === 'right') {
162      return (

```

```

163         <ChevronRightIcon
164             className={cn('cn-rtl-flip size-4', className)}
165             {...props}
166         />
167     );
168 }
169
170     return (
171         <ChevronDownIcon className={cn('size-4', className)} {...props} />
172     );
173 },
174 DayButton: ({ ...props }) => (
175     <CalendarDayButton locale={locale} {...props} />
176 ),
177 WeekNumber: ({ children, ...props }) => {
178     return (
179         <td {...props}>
180             <div className="flex size-(--cell-size) items-center justify-center text-center">
181                 {children}
182             </div>
183         </td>
184     );
185 },
186 ...components,
187 }}
188 {...props}
189 />
190 );
191 }
192
193 function CalendarDayButton({
194     className,
195     day,
196     modifiers,
197     locale,
198     ...props
199 }: React.ComponentProps<typeof DayButton> & { locale?: Partial<Locale> }) {
200     const defaultClassNames = getDefaultClassNames();
201
202     const ref = React.useRef<HTMLButtonElement>(null);
203     React.useEffect(() => {
204         if (modifiers.focused) ref.current?.focus();
205     }, [modifiers.focused]);
206
207     return (
208         <Button
209             variant="ghost"
210             size="icon"
211             data-day={day.date.toLocaleDateString(locale?.code)}
212             data-selected-single={
213                 modifiers.selected &&
214                 !modifiers.range_start &&
215                 !modifiers.range_end &&
216                 !modifiers.range_middle
217             }
218             data-range-start={modifiers.range_start}
219             data-range-end={modifiers.range_end}
220             data-range-middle={modifiers.range_middle}
221             className={cn(
222                 'relative isolate z-10 flex aspect-square size-auto w-full min-w(--cell-size) flex-col gap-1
border-0 leading-none font-normal group-data-[focused=true]/day:relative group-
data-[focused=true]/day:z-10 group-data-[focused=true]/day:border-ring group-
data-[focused=true]/day:ring-[3px] group-data-[focused=true]/day:ring-ring/50 data-[range-
end=true]:rounded-(--cell-radius) data-[range-end=true]:rounded-r-(--cell-radius) data-[range-
end=true]:bg-primary data-[range-end=true]:text-primary-foreground data-[range-middle=true]:rounded-none
data-[range-middle=true]:bg-muted data-[range-middle=true]:text-foreground data-[range-
start=true]:rounded-(--cell-radius) data-[range-start=true]:rounded-l-(--cell-radius) data-[range-
start=true]:bg-primary data-[range-start=true]:text-primary-foreground data-[selected-single=true]:bg-
primary data-[selected-single=true]:text-primary-foreground dark:hover:text-foreground [&>span]:text-xs
[&>span]:opacity-70',
223                 defaultClassNames.day,
224                 className,
225             )}
226             {...props}
227         />
228     );
229 }
230
231 export { Calendar, CalendarDayButton };

```

components/ui/card.tsx

```
1 import * as React from 'react';
2
3 import { cn } from '@lib/utils';
4
5 function Card({
6   className,
7   size = 'default',
8   ...props
9 }: React.ComponentProps<'div'> & { size?: 'default' | 'sm' }) {
10   return (
11     <div
12       data-slot="card"
13       data-size={size}
14       className={cn(
15         'group/card flex flex-col gap-(--card-spacing) overflow-hidden rounded-xl bg-card py-(--card-
16         spacing) text-sm text-card-foreground ring-1 ring-foreground/10 [--card-spacing:--spacing(4)] has-
17         data-[slot=card-footer]:pb-0 has-[>img:first-child]:pt-0 data-[size=sm]:[--card-spacing:--spacing(3)]
18         data-[size=sm]:has-data-[slot=card-footer]:pb-0 *: [img:first-child]:rounded-t-xl *: [img:last-
19         child]:rounded-b-xl',
20       className,
21     )}
22     {...props}
23   );
24 }
25
26 function CardHeader({ className, ...props }: React.ComponentProps<'div'>) {
27   return (
28     <div
29       data-slot="card-header"
30       className={cn(
31         'group/card-header @container/card-header grid auto-rows-min items-start gap-1 rounded-t-xl
32         px-(--card-spacing) has-data-[slot=card-action]:grid-cols-[1fr_auto] has-data-[slot=card-
33         description]:grid-rows-[auto_auto] [.border-b]:pb-(--card-spacing)',
34       className,
35     )}
36     {...props}
37   );
38 }
39
40 function CardTitle({ className, ...props }: React.ComponentProps<'div'>) {
41   return (
42     <div
43       data-slot="card-title"
44       className={cn(
45         'font-heading text-base leading-snug font-medium group-data-[size=sm]/card:text-sm',
46         className,
47       )}
48     {...props}
49   );
50 }
51
52 function CardDescription({ className, ...props }: React.ComponentProps<'div'>) {
53   return (
54     <div
55       data-slot="card-description"
56       className={cn('text-sm text-muted-foreground', className)}
57     {...props}
58   );
59 }
60
61 function CardAction({ className, ...props }: React.ComponentProps<'div'>) {
62   return (
63     <div
64       data-slot="card-action"
65       className={cn(
66         'col-start-2 row-span-2 row-start-1 self-start justify-self-end',
67         className,
68       )}
69     {...props}
70   );
71 }
72
73 function CardContent({ className, ...props }: React.ComponentProps<'div'>) {
74   return (
75     <div
76       data-slot="card-content"
77       className={cn('px-(--card-spacing)', className)}
78     {...props}
79   );
80 }
```

```
77     {...props}
78   />
79   );
80 }
81
82 function CardFooter({ className, ...props }: React.ComponentProps<'div'>) {
83   return (
84     <div
85       data-slot="card-footer"
86       className={cn(
87         'flex items-center rounded-b-xl border-t bg-muted/50 p(--card-spacing)',
88         className,
89       )}
90       {...props}
91     />
92   );
93 }
94
95 export {
96   Card,
97   CardHeader,
98   CardFooter,
99   CardTitle,
100  CardAction,
101  CardDescription,
102  CardContent,
103 };
```

components/ui/carousel.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import useEmblaCarousel, {
5   type UseEmblaCarouselType,
6 } from 'embla-carousel-react';
7
8 import { cn } from '@lib/utils';
9 import { Button } from '@components/ui/button';
10 import { ChevronLeftIcon, ChevronRightIcon } from 'lucide-react';
11
12 type CarouselApi = UseEmblaCarouselType[1];
13 type UseCarouselParameters = Parameters<typeof useEmblaCarousel>;
14 type CarouselOptions = UseCarouselParameters[0];
15 type CarouselPlugin = UseCarouselParameters[1];
16
17 type CarouselProps = {
18   opts?: CarouselOptions;
19   plugins?: CarouselPlugin;
20   orientation?: 'horizontal' | 'vertical';
21   setApi?: (api: CarouselApi) => void;
22 };
23
24 type CarouselContextProps = {
25   carouselRef: ReturnType<typeof useEmblaCarousel>[0];
26   api: ReturnType<typeof useEmblaCarousel>[1];
27   scrollPrev: () => void;
28   scrollNext: () => void;
29   canScrollPrev: boolean;
30   canScrollNext: boolean;
31 } & CarouselProps;
32
33 const CarouselContext = React.createContext<CarouselContextProps | null>(null);
34
35 function useCarousel() {
36   const context = React.useContext(CarouselContext);
37
38   if (!context) {
39     throw new Error('useCarousel must be used within a <Carousel />');
40   }
41
42   return context;
43 }
44
45 function Carousel({
46   orientation = 'horizontal',
47   opts,
48   setApi,
49   plugins,
50   className,
51   children,
52   ...props
53 }: React.ComponentProps<'div'> & CarouselProps) {
54   const [carouselRef, api] = useEmblaCarousel(
55     {
56       ...opts,
57       axis: orientation === 'horizontal' ? 'x' : 'y',
58     },
59     plugins,
60   );
61   const [canScrollPrev, setCanScrollPrev] = React.useState(false);
62   const [canScrollNext, setCanScrollNext] = React.useState(false);
63
64   const onSelect = React.useCallback((api: CarouselApi) => {
65     if (!api) return;
66     setCanScrollPrev(api.canScrollPrev());
67     setCanScrollNext(api.canScrollNext());
68   }, []);
69
70   const scrollPrev = React.useCallback(() => {
71     api?.scrollPrev();
72   }, [api]);
73
74   const scrollNext = React.useCallback(() => {
75     api?.scrollNext();
76   }, [api]);
77
78   const handleKeyDown = React.useCallback(
79     (event: React.KeyboardEvent<HTMLDivElement>) => {
80       if (event.key === 'ArrowLeft') {
81         event.preventDefault();
82         scrollPrev();
83       }
84     },
85     [scrollPrev],
86   );
```

```

83     } else if (event.key === 'ArrowRight') {
84         event.preventDefault();
85         scrollNext();
86     }
87 },
88 [scrollPrev, scrollNext],
89 );
90
91 React.useEffect(() => {
92     if (!api || !setApi) return;
93     setApi(api);
94 }, [api, setApi]);
95
96 React.useEffect(() => {
97     if (!api) return;
98     onSelect(api);
99     api.on('reInit', onSelect);
100    api.on('select', onSelect);
101
102    return () => {
103        api?.off('select', onSelect);
104    };
105 }, [api, onSelect]);
106
107 return (
108     <CarouselContext.Provider
109         value={{
110             carouselRef,
111             api: api,
112             opts,
113             orientation:
114                 orientation || (opts?.axis === 'y' ? 'vertical' : 'horizontal'),
115             scrollPrev,
116             scrollNext,
117             canScrollPrev,
118             canScrollNext,
119         }}
120     >
121         <div
122             onKeyDownCapture={handleKeyDown}
123             className={cn('relative', className)}
124             role="region"
125             aria-roledescription="carousel"
126             data-slot="carousel"
127             {...props}
128         >
129             {children}
130         </div>
131     </CarouselContext.Provider>
132 );
133 }
134
135 function CarouselContent({ className, ...props }: React.ComponentProps<'div'>) {
136     const { carouselRef, orientation } = useCarousel();
137
138     return (
139         <div
140             ref={carouselRef}
141             className="overflow-hidden"
142             data-slot="carousel-content"
143         >
144             <div
145                 className={cn(
146                     'flex',
147                     orientation === 'horizontal' ? '-ml-4' : '-mt-4 flex-col',
148                     className,
149                 )}
150                 {...props}
151             />
152         </div>
153     );
154 }
155
156 function CarouselItem({ className, ...props }: React.ComponentProps<'div'>) {
157     const { orientation } = useCarousel();
158
159     return (
160         <div
161             role="group"
162             aria-roledescription="slide"
163             data-slot="carousel-item"
164             className={cn(
165                 'min-w-0 shrink-0 grow-0 basis-full',
166                 orientation === 'horizontal' ? 'pl-4' : 'pt-4',
167                 className,

```

```

168     })
169     {...props}
170   />
171 );
172 }
173
174 function CarouselPrevious({
175   className,
176   variant = 'outline',
177   size = 'icon-sm',
178   ...props
179 }: React.ComponentProps<typeof Button>) {
180   const { orientation, scrollPrev, canScrollPrev } = useCarousel();
181
182   return (
183     <Button
184       data-slot="carousel-previous"
185       variant={variant}
186       size={size}
187       className={cn(
188         'rounded-full absolute touch-manipulation',
189         orientation === 'horizontal'
190           ? 'inset-y-0 -left-12 my-auto'
191           : '-top-12 left-1/2 -translate-x-1/2 rotate-90',
192         className,
193       )}
194       disabled={!canScrollPrev}
195       onClick={scrollPrev}
196       {...props}
197     >
198       <ChevronLeftIcon className="cn-rtl-flip" />
199       <span className="sr-only">Previous slide</span>
200     </Button>
201   );
202 }
203
204 function CarouselNext({
205   className,
206   variant = 'outline',
207   size = 'icon-sm',
208   ...props
209 }: React.ComponentProps<typeof Button>) {
210   const { orientation, scrollNext, canScrollNext } = useCarousel();
211
212   return (
213     <Button
214       data-slot="carousel-next"
215       variant={variant}
216       size={size}
217       className={cn(
218         'rounded-full absolute touch-manipulation',
219         orientation === 'horizontal'
220           ? 'inset-y-0 -right-12 my-auto'
221           : '-bottom-12 left-1/2 -translate-x-1/2 rotate-90',
222         className,
223       )}
224       disabled={!canScrollNext}
225       onClick={scrollNext}
226       {...props}
227     >
228       <ChevronRightIcon className="cn-rtl-flip" />
229       <span className="sr-only">Next slide</span>
230     </Button>
231   );
232 }
233
234 export {
235   type CarouselApi,
236   Carousel,
237   CarouselContent,
238   CarouselItem,
239   CarouselPrevious,
240   CarouselNext,
241   useCarousel,
242 };

```

components/ui/chart.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import * as RechartsPrimitive from 'recharts';
5 import type { TooltipValueType } from 'recharts';
6
7 import { cn } from '@lib/utils';
8
9 // Format: { THEME_NAME: CSS_SELECTOR }
10 const THEMES = { light: '', dark: '.dark' } as const;
11
12 const INITIAL_DIMENSION = { width: 320, height: 200 } as const;
13 type TooltipNameType = number | string;
14
15 export type ChartConfig = Record<
16   string,
17   {
18     label?: React.ReactNode;
19     icon?: React.ComponentType;
20   } & (
21     | { color?: string; theme?: never }
22     | { color?: never; theme: Record<keyof typeof THEMES, string> }
23   )
24 >;
25
26 type ChartContextProps = {
27   config: ChartConfig;
28 };
29
30 const ChartContext = React.createContext<ChartContextProps | null>(null);
31
32 function useChart() {
33   const context = React.useContext(ChartContext);
34
35   if (!context) {
36     throw new Error('useChart must be used within a <ChartContainer />');
37   }
38
39   return context;
40 }
41
42 function ChartContainer({
43   id,
44   className,
45   children,
46   config,
47   initialDimension = INITIAL_DIMENSION,
48   ...props
49 }: React.ComponentProps<'div'> & {
50   config: ChartConfig;
51   children: React.ComponentProps<
52     typeof RechartsPrimitive.ResponsiveContainer
53   >['children'];
54   initialDimension?: {
55     width: number;
56     height: number;
57   };
58 }) {
59   const uniqueId = React.useId();
60   const chartId = `chart-${id ?? uniqueId.replace(/:/g, '')}`;
61
62   return (
63     <ChartContext.Provider value={{ config }}>
64       <div
65         data-slot="chart"
66         data-chart={chartId}
67         className={cn(
68           "flex aspect-video justify-center text-xs [&_.recharts-cartesian-axis-tick_text]:fill-muted-foreground [&_.recharts-cartesian-grid_line[stroke='#ccc']]:stroke-border/50 [&_.recharts-curve.recharts-tooltip-cursor]:stroke-border [&_.recharts-dot[stroke='#fff']]:stroke-transparent [&_.recharts-layer]:outline-hidden [&_.recharts-polar-grid_[stroke='#ccc']]:stroke-border [&_.recharts-radial-bar-background-sector]:fill-muted [&_.recharts-rectangle.recharts-tooltip-cursor]:fill-muted [&_.recharts-reference_line_[stroke='#ccc']]:stroke-border [&_.recharts-sector]:outline-hidden [&_.recharts-sector[stroke='#fff']]:stroke-transparent [&_.recharts-surface]:outline-hidden",
69           className,
70         )}
71         {...props}
72       >
73         <ChartStyle id={chartId} config={config} />
74         <RechartsPrimitive.ResponsiveContainer
75           initialDimension={initialDimension}
76         />
```

```

77         {children}
78     </RechartsPrimitive.ResponsiveContainer>
79 </div>
80 </ChartContext.Provider>
81 );
82 }
83
84 const ChartStyle = ({ id, config }: { id: string; config: ChartConfig }) => {
85     const colorConfig = Object.entries(config).filter(
86         ([, config]) => config.theme ?? config.color,
87     );
88
89     if (!colorConfig.length) {
90         return null;
91     }
92
93     return (
94         <style
95             dangerouslySetInnerHTML={{
96                 __html: Object.entries(THEMES)
97                     .map(
98                         ([theme, prefix]) => `
99 ${prefix} [data-chart=${id}] {
100 ${colorConfig
101     .map(([key, itemConfig]) => {
102         const color =
103             itemConfig.theme?.[theme as keyof typeof itemConfig.theme] ??
104             itemConfig.color;
105         return color ? `--color-${key}: ${color};` : null;
106     })
107     .join('\n')}
108 }
109 `,
110         )
111         .join('\n'),
112     )}
113     />
114 );
115 };
116
117 const ChartTooltip = RechartsPrimitive.Tooltip;
118
119 function ChartTooltipContent({
120     active,
121     payload,
122     className,
123     indicator = 'dot',
124     hideLabel = false,
125     hideIndicator = false,
126     label,
127     labelFormatter,
128     labelClassName,
129     formatter,
130     color,
131     nameKey,
132     labelKey,
133 }: React.ComponentProps<typeof RechartsPrimitive.Tooltip> &
134 React.ComponentProps<'div'> & {
135     hideLabel?: boolean;
136     hideIndicator?: boolean;
137     indicator?: 'line' | 'dot' | 'dashed';
138     nameKey?: string;
139     labelKey?: string;
140 } & Omit<
141     RechartsPrimitive.DefaultTooltipContentProps<
142         TooltipValueType,
143         TooltipNameType
144     >,
145     'accessibilityLayer'
146 >) {
147     const { config } = useChart();
148
149     const tooltipLabel = React.useMemo(() => {
150         if (hideLabel || !payload?.length) {
151             return null;
152         }
153
154         const [item] = payload;
155         const key = `${labelKey ?? item?.dataKey ?? item?.name ?? 'value'}`;
156         const itemConfig = getPayloadConfigFromPayload(config, item, key);
157         const value =
158             !labelKey && typeof label === 'string'
159                 ? (config[label]?.label ?? label)
160                 : itemConfig?.label;
161

```

```

162     if (labelFormatter) {
163         return (
164             <div className={cn('font-medium', labelClassName)}>
165                 {labelFormatter(value, payload)}
166             </div>
167         );
168     }
169
170     if (!value) {
171         return null;
172     }
173
174     return <div className={cn('font-medium', labelClassName)}>{value}</div>;
175 }, [
176     label,
177     labelFormatter,
178     payload,
179     hideLabel,
180     labelClassName,
181     config,
182     labelKey,
183 ]);
184
185 if (!active || !payload?.length) {
186     return null;
187 }
188
189 const nestLabel = payload.length === 1 && indicator !== 'dot';
190
191 return (
192     <div
193         className={cn(
194             'border-border/50 bg-background gap-1.5 rounded-lg border px-2.5 py-1.5 text-xs shadow-xl grid
min-w-32 items-start',
195             className,
196         )}
197     >
198         {!nestLabel ? tooltipLabel : null}
199         <div className="grid gap-1.5">
200             {payload
201                 .filter((item) => item.type !== 'none')
202                 .map((item, index) => {
203                     const key = `${nameKey} ?? item.name ?? item.dataKey ?? 'value'`;
204                     const itemConfig = getPayloadConfigFromPayload(config, item, key);
205                     const indicatorColor = color ?? item.payload?.fill ?? item.color;
206
207                     return (
208                         <div
209                             key={index}
210                             className={cn(
211                                 'flex w-full flex-wrap items-stretch gap-2 [>svg]:h-2.5 [>svg]:w-2.5 [>svg]:text-
muted-foreground',
212                                 indicator === 'dot' && 'items-center',
213                             )}
214                         >
215                             {formatter && item?.value !== undefined && item.name ? (
216                                 formatter(item.value, item.name, item, index, item.payload)
217                             ) : (
218                                 <>
219                                     {itemConfig?.icon ? (
220                                         <itemConfig.icon />
221                                     ) : (
222                                         !hideIndicator && (
223                                             <div
224                                                 className={cn(
225                                                     'shrink-0 rounded-[2px] border-(--color-border) bg-(--color-bg)',
226                                                     {
227                                                         'h-2.5 w-2.5': indicator === 'dot',
228                                                         'w-1': indicator === 'line',
229                                                         'w-0 border-[1.5px] border-dashed bg-transparent':
230                                                             indicator === 'dashed',
231                                                         'my-0.5': nestLabel && indicator === 'dashed',
232                                                     },
233                                                 )}
234                                                 style={
235                                                     {
236                                                         '--color-bg': indicatorColor,
237                                                         '--color-border': indicatorColor,
238                                                     } as React.CSSProperties
239                                                 }
240                                             />
241                                         )
242                                     )}
243                                 <div
244                                     className={cn(

```

```

245         'flex flex-1 justify-between leading-none',
246         nestLabel ? 'items-end' : 'items-center',
247     )}
248 >
249     <div className="grid gap-1.5">
250         {nestLabel ? tooltipLabel : null}
251         <span className="text-muted-foreground">
252             {itemConfig?.label ?? item.name}
253         </span>
254     </div>
255     {item.value !== null && (
256         <span className="font-mono font-medium text-foreground tabular-nums">
257             {typeof item.value === 'number'
258               ? item.value.toLocaleString()
259               : String(item.value)}
260         </span>
261     )}
262 </div>
263 </>
264 )}
265 </div>
266 );
267 }}}}
268 </div>
269 </div>
270 );
271 }
272
273 const ChartLegend = RechartsPrimitive.Legend;
274
275 function ChartLegendContent({
276     className,
277     hideIcon = false,
278     payload,
279     verticalAlign = 'bottom',
280     nameKey,
281 }: React.ComponentProps<'div'> & {
282     hideIcon?: boolean;
283     nameKey?: string;
284 } & RechartsPrimitive.DefaultLegendContentProps) {
285     const { config } = useChart();
286
287     if (!payload?.length) {
288         return null;
289     }
290
291     return (
292         <div
293             className={cn(
294                 'flex items-center justify-center gap-4',
295                 verticalAlign === 'top' ? 'pb-3' : 'pt-3',
296                 className,
297             )}
298         >
299             {payload
300                 .filter((item) => item.type !== 'none')
301                 .map((item, index) => {
302                     const key = `${nameKey ?? item.dataKey ?? 'value'}`;
303                     const itemConfig = getPayloadConfigFromPayload(config, item, key);
304
305                     return (
306                         <div
307                             key={index}
308                             className={cn(
309                                 'flex items-center gap-1.5 [&svg]:h-3 [&svg]:w-3 [&svg]:text-muted-foreground',
310                             )}
311                         >
312                             {itemConfig?.icon && !hideIcon ? (
313                                 <itemConfig.icon />
314                             ) : (
315                                 <div
316                                     className="h-2 w-2 shrink-0 rounded-[2px]"
317                                     style={{
318                                         backgroundColor: item.color,
319                                     }}
320                                 />
321                             )}
322                             {itemConfig?.label}
323                         </div>
324                     );
325                 })}
326         </div>
327     );
328 }
329

```

```

330 function getPayloadConfigFromPayload(
331   config: ChartConfig,
332   payload: unknown,
333   key: string,
334 ) {
335   if (typeof payload !== 'object' || payload === null) {
336     return undefined;
337   }
338
339   const payloadPayload =
340     'payload' in payload &&
341     typeof payload.payload === 'object' &&
342     payload.payload !== null
343     ? payload.payload
344     : undefined;
345
346   let configLabelKey: string = key;
347
348   if (
349     key in payload &&
350     typeof payload[key as keyof typeof payload] === 'string'
351   ) {
352     configLabelKey = payload[key as keyof typeof payload] as string;
353   } else if (
354     payloadPayload &&
355     key in payloadPayload &&
356     typeof payloadPayload[key as keyof typeof payloadPayload] === 'string'
357   ) {
358     configLabelKey = payloadPayload[
359       key as keyof typeof payloadPayload
360     ] as string;
361   }
362
363   return configLabelKey in config ? config[configLabelKey] : config[key];
364 }
365
366 export {
367   ChartContainer,
368   ChartTooltip,
369   ChartTooltipContent,
370   ChartLegend,
371   ChartLegendContent,
372   ChartStyle,
373 };

```

components/ui/checkbox.tsx

```
1 'use client';
2
3 import { Checkbox as CheckboxPrimitive } from '@base-ui/react/checkbox';
4
5 import { cn } from '@lib/utils';
6 import { CheckIcon } from 'lucide-react';
7
8 function Checkbox({ className, ...props }: CheckboxPrimitive.Root.Props) {
9   return (
10     <CheckboxPrimitive.Root
11       data-slot="checkbox"
12       className={cn(
13         'peer relative flex size-4 shrink-0 items-center justify-center rounded-[4px] border border-
14 input transition-colors outline-none group-has-disabled/field:opacity-50 group-has-[:focus-
15 visible]/field-label:ring-0 group-has-[:focus-visible]/field-label:not-data-checked:border-input
16 after:absolute after:-inset-x-3 after:-inset-y-2 focus-visible:border-ring focus-visible:ring-3 focus-
17 visible:ring-ring/50 disabled:cursor-not-allowed disabled:opacity-50 aria-invalid:border-destructive
18 aria-invalid:ring-3 aria-invalid:ring-destructive/20 aria-invalid:aria-checked:border-primary dark:bg-
19 input/30 dark:aria-invalid:border-destructive/50 dark:aria-invalid:ring-destructive/40 data-
20 checked:border-primary data-checked:bg-primary data-checked:text-primary-foreground group-has-[:focus-
21 visible]/field-label:data-checked:border-primary dark:data-checked:bg-primary',
22       className,
23     )}
24     {...props}
25   >
26     <CheckboxPrimitive.Indicator
27       data-slot="checkbox-indicator"
28       className="grid place-content-center text-current transition-none [>svg]:size-3.5"
29     >
30       <CheckIcon />
31     </CheckboxPrimitive.Indicator>
32   </CheckboxPrimitive.Root>
33 );
34 }
35
36 export { Checkbox };
```

components/ui/collapsible.tsx

```
1 'use client';
2
3 import { Collapsible as CollapsiblePrimitive } from '@base-ui/react/collapsible';
4
5 function Collapsible({ ...props }: CollapsiblePrimitive.Root.Props) {
6   return <CollapsiblePrimitive.Root data-slot="collapsible" {...props} />;
7 }
8
9 function CollapsibleTrigger({ ...props }: CollapsiblePrimitive.Trigger.Props) {
10   return (
11     <CollapsiblePrimitive.Trigger data-slot="collapsible-trigger" {...props} />
12   );
13 }
14
15 function CollapsibleContent({ ...props }: CollapsiblePrimitive.Panel.Props) {
16   return (
17     <CollapsiblePrimitive.Panel data-slot="collapsible-content" {...props} />
18   );
19 }
20
21 export { Collapsible, CollapsibleTrigger, CollapsibleContent };
```

components/ui/combobox.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { Combobox as ComboboxPrimitive } from '@base-ui/react';
5
6 import { cn } from '@lib/utils';
7 import { Button } from '@components/ui/button';
8 import {
9   InputGroup,
10  InputGroupAddon,
11  InputGroupButton,
12  InputGroupInput,
13 } from '@components/ui/input-group';
14 import { ChevronDownIcon, XIcon, CheckIcon } from 'lucide-react';
15
16 const Combobox = ComboboxPrimitive.Root;
17
18 function ComboboxValue({ ...props }: ComboboxPrimitive.Value.Props) {
19   return <ComboboxPrimitive.Value data-slot="combobox-value" {...props} />;
20 }
21
22 function ComboboxTrigger({
23   className,
24   children,
25   ...props
26 }: ComboboxPrimitive.Trigger.Props) {
27   return (
28     <ComboboxPrimitive.Trigger
29       data-slot="combobox-trigger"
30       className={cn("[&_svg:not([class*='size-'])]:size-4", className)}
31       {...props}
32     >
33       {children}
34       <ChevronDownIcon className="text-muted-foreground size-4 pointer-events-none" />
35     </ComboboxPrimitive.Trigger>
36   );
37 }
38
39 function ComboboxClear({ className, ...props }: ComboboxPrimitive.Clear.Props) {
40   return (
41     <ComboboxPrimitive.Clear
42       data-slot="combobox-clear"
43       render={<InputGroupButton variant="ghost" size="icon-xs" />}
44       className={cn(className)}
45       {...props}
46     >
47       <XIcon className="pointer-events-none" />
48     </ComboboxPrimitive.Clear>
49   );
50 }
51
52 function ComboboxInput({
53   className,
54   children,
55   disabled = false,
56   showTrigger = true,
57   showClear = false,
58   ...props
59 }: ComboboxPrimitive.Input.Props & {
60   showTrigger?: boolean;
61   showClear?: boolean;
62 }) {
63   return (
64     <InputGroup className={cn('w-auto', className)}>
65       <ComboboxPrimitive.Input
66         render={<InputGroupInput disabled={disabled} />}
67         {...props}
68       />
69       <InputGroupAddon align="inline-end">
70         {showTrigger && (
71           <InputGroupButton
72             size="icon-xs"
73             variant="ghost"
74             render={<ComboboxTrigger />}
75             data-slot="input-group-button"
76             className="group-has-data-[slot=combobox-clear]/input-group:hidden data-pressed:bg-transparent"
77             disabled={disabled}
78           />
79         )}
80         {showClear && <ComboboxClear disabled={disabled} />}
81       </InputGroupAddon>
82     </>
83   );
84 }
```

```

82     {children}
83   </InputGroup>
84   });
85 }
86
87 function ComboboxContent({
88   className,
89   side = 'bottom',
90   sideOffset = 6,
91   align = 'start',
92   alignOffset = 0,
93   anchor,
94   ...props
95 }: ComboboxPrimitive.Popup.Props &
96   Pick<
97     ComboboxPrimitive.Positioner.Props,
98     'side' | 'align' | 'sideOffset' | 'alignOffset' | 'anchor'
99   >) {
100   return (
101     <ComboboxPrimitive.Portal>
102       <ComboboxPrimitive.Positioner
103         side={side}
104         sideOffset={sideOffset}
105         align={align}
106         alignOffset={alignOffset}
107         anchor={anchor}
108         className="isolate z-50"
109       >
110         <ComboboxPrimitive.Popup
111           data-slot="combobox-content"
112           data-chips={!anchor}
113           className={cn(
114             'bg-popover text-popover-foreground data-open:animate-in data-closed:animate-out data-
closed:fade-out-0 data-open:fade-in-0 data-closed:zoom-out-95 data-open:zoom-in-95
data-[side=bottom]:slide-in-from-top-2 data-[side=left]:slide-in-from-right-2 data-[side=right]:slide-
in-from-left-2 data-[side=top]:slide-in-from-bottom-2 ring-foreground/10 *:data-[slot=input-group]:bg-
input/30 *:data-[slot=input-group]:border-input/30 overflow-hidden rounded-lg shadow-md ring-1
duration-100 *:data-[slot=input-group]:m-1 *:data-[slot=input-group]:mb-0 *:data-[slot=input-group]:h-8
*:data-[slot=input-group]:shadow-none data-[side=inline-start]:slide-in-from-right-2 data-[side=inline-
end]:slide-in-from-left-2 cn-menu-target cn-menu-translucent group/combobox-content relative
max-h-(--available-height) w-(--anchor-width) max-w-(--available-width) min-w-[calc(var(--anchor-
width)+--spacing(7))] origin-(--transform-origin) data-[chips=true]:min-w-(--anchor-width)',
115             className,
116           )}
117           {...props}
118         />
119       </ComboboxPrimitive.Positioner>
120     </ComboboxPrimitive.Portal>
121   );
122 }
123
124 function ComboboxList({ className, ...props }: ComboboxPrimitive.List.Props) {
125   return (
126     <ComboboxPrimitive.List
127       data-slot="combobox-list"
128       className={cn(
129         'no-scrollbar max-h-[min(calc(--spacing(72)---spacing(9)),calc(var(--available-
height)---spacing(9)))] scroll-py-1 p-1 data-empty:p-0 overflow-y-auto overscroll-contain',
130         className,
131       )}
132       {...props}
133     />
134   );
135 }
136
137 function ComboboxItem({
138   className,
139   children,
140   ...props
141 }: ComboboxPrimitive.Item.Props) {
142   return (
143     <ComboboxPrimitive.Item
144       data-slot="combobox-item"
145       className={cn(
146         "data-highlighted:bg-accent data-highlighted:text-accent-foreground not-
data-[variant=destructive]:data-highlighted:**:text-accent-foreground gap-2 rounded-md py-1 pr-8 pl-1.5
text-sm [&_svg:not([class*='size-']):size-4 relative flex w-full cursor-default items-center outline-
hidden select-none data-disabled:pointer-events-none data-disabled:opacity-50 [&_svg]:pointer-events-
none [&_svg]:shrink-0",
147         className,
148       )}
149       {...props}
150     >
151       {children}
152     <ComboboxPrimitive.ItemIndicator

```

```

153         render={
154             <span className="pointer-events-none absolute right-2 flex size-4 items-center justify-center"
155             />
156         }
157         <CheckIcon className="pointer-events-none" />
158     </ComboboxPrimitive.ItemIndicator>
159 </ComboboxPrimitive.Item>
160 );
161 }
162
163 function ComboboxGroup({ className, ...props }: ComboboxPrimitive.Group.Props) {
164     return (
165         <ComboboxPrimitive.Group
166             data-slot="combobox-group"
167             className={cn(className)}
168             {...props}
169         />
170     );
171 }
172
173 function ComboboxLabel({
174     className,
175     ...props
176 }: ComboboxPrimitive.GroupLabel.Props) {
177     return (
178         <ComboboxPrimitive.GroupLabel
179             data-slot="combobox-label"
180             className={cn('text-muted-foreground px-2 py-1.5 text-xs', className)}
181             {...props}
182         />
183     );
184 }
185
186 function ComboboxCollection({ ...props }: ComboboxPrimitive.Collection.Props) {
187     return (
188         <ComboboxPrimitive.Collection data-slot="combobox-collection" {...props} />
189     );
190 }
191
192 function ComboboxEmpty({ className, ...props }: ComboboxPrimitive.Empty.Props) {
193     return (
194         <ComboboxPrimitive.Empty
195             data-slot="combobox-empty"
196             className={cn(
197                 'text-muted-foreground hidden w-full justify-center py-2 text-center text-sm group-data-
198                 empty/combobox-content:flex',
199                 className,
200             )}
201             {...props}
202         />
203     );
204 }
205
206 function ComboboxSeparator({
207     className,
208     ...props
209 }: ComboboxPrimitive.Separator.Props) {
210     return (
211         <ComboboxPrimitive.Separator
212             data-slot="combobox-separator"
213             className={cn('bg-border -mx-1 my-1 h-px', className)}
214             {...props}
215         />
216     );
217 }
218
219 function ComboboxChips({
220     className,
221     ...props
222 }: React.ComponentPropsWithRef<typeof ComboboxPrimitive.Chips> &
223     ComboboxPrimitive.Chips.Props) {
224     return (
225         <ComboboxPrimitive.Chips
226             data-slot="combobox-chips"
227             className={cn(
228                 'dark:bg-input/30 border-input focus-within:border-ring focus-within:ring-ring/50 has-aria-
229                 invalid:ring-destructive/20 dark:has-aria-invalid:ring-destructive/40 has-aria-invalid:border-
230                 destructive dark:has-aria-invalid:border-destructive/50 flex min-h-8 flex-wrap items-center gap-1
231                 rounded-lg border bg-transparent bg-clip-padding px-2.5 py-1 text-sm transition-colors focus-

```

```

232 });
233 }
234
235 function ComboboxChip({
236   className,
237   children,
238   showRemove = true,
239   ...props
240 }: ComboboxPrimitive.Chip.Props & {
241   showRemove?: boolean;
242 }) {
243   return (
244     <ComboboxPrimitive.Chip
245       data-slot="combobox-chip"
246       className={cn(
247         'bg-muted text-foreground flex h-[calc(--spacing(5.25))] w-fit items-center justify-center gap-1
248         rounded-sm px-1.5 text-xs font-medium whitespace-nowrap has-data-[slot=combobox-chip-remove]:pr-0 has-
249         disabled:pointer-events-none has-disabled:cursor-not-allowed has-disabled:opacity-50',
250         className,
251       )}
252       {...props}
253     >
254       {children}
255       {showRemove && (
256         <ComboboxPrimitive.ChipRemove
257           render={<Button variant="ghost" size="icon-xs" />}
258           className="-ml-1 opacity-50 hover:opacity-100"
259           data-slot="combobox-chip-remove"
260         >
261           <XIcon className="pointer-events-none" />
262         </ComboboxPrimitive.ChipRemove>
263       )}
264     </ComboboxPrimitive.Chip>
265   );
266 }
267
268 function ComboboxChipsInput({
269   className,
270   ...props
271 }: ComboboxPrimitive.Input.Props) {
272   return (
273     <ComboboxPrimitive.Input
274       data-slot="combobox-chip-input"
275       className={cn('min-w-16 flex-1 outline-none', className)}
276       {...props}
277     />
278   );
279 }
280
281 function useComboboxAnchor() {
282   return React.useRef<HTMLDivElement | null>(null);
283 }
284
285 export {
286   Combobox,
287   ComboboxInput,
288   ComboboxContent,
289   ComboboxList,
290   ComboboxItem,
291   ComboboxGroup,
292   ComboboxLabel,
293   ComboboxCollection,
294   ComboboxEmpty,
295   ComboboxSeparator,
296   ComboboxChips,
297   ComboboxChip,
298   ComboboxChipsInput,
299   ComboboxTrigger,
300   ComboboxValue,
301   useComboboxAnchor,
302 };

```

components/ui/command.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { Command as CommandPrimitive } from 'cmdk';
5
6 import { cn } from '@/lib/utils';
7 import {
8   Dialog,
9   DialogContent,
10  DialogDescription,
11  DialogHeader,
12  DialogTitle,
13 } from '@/components/ui/dialog';
14 import { InputGroup, InputGroupAddon } from '@/components/ui/input-group';
15 import { SearchIcon, CheckIcon } from 'lucide-react';
16
17 function Command({
18   className,
19   ...props
20 }: React.ComponentProps<typeof CommandPrimitive>) {
21   return (
22     <CommandPrimitive
23       data-slot="command"
24       className={cn(
25         'bg-popover text-popover-foreground rounded-xl! p-1 flex size-full flex-col overflow-hidden',
26         className,
27       )}
28       {...props}
29     />
30   );
31 }
32
33 function CommandDialog({
34   title = 'Command Palette',
35   description = 'Search for a command to run...',
36   children,
37   className,
38   showCloseButton = false,
39   ...props
40 }: Omit<React.ComponentProps<typeof Dialog>, 'children'> & {
41   title?: string;
42   description?: string;
43   className?: string;
44   showCloseButton?: boolean;
45   children: React.ReactNode;
46 }) {
47   return (
48     <Dialog {...props}>
49       <DialogHeader className="sr-only">
50         <DialogTitle>{title}</DialogTitle>
51         <DialogDescription>{description}</DialogDescription>
52       </DialogHeader>
53       <DialogContent
54         className={cn(
55           'rounded-xl! top-1/3 translate-y-0 overflow-hidden p-0',
56           className,
57         )}
58         showCloseButton={showCloseButton}
59       >
60         {children}
61       </DialogContent>
62     </Dialog>
63   );
64 }
65
66 function CommandInput({
67   className,
68   ...props
69 }: React.ComponentProps<typeof CommandPrimitive.Input>) {
70   return (
71     <div data-slot="command-input-wrapper" className="p-1 pb-0">
72       <InputGroup className="bg-input/30 border-input/30 h-8! rounded-lg! shadow-none!
73       *:data-[slot=input-group-addon]:pl-2!">
74         <CommandPrimitive.Input
75           data-slot="command-input"
76           className={cn(
77             'w-full text-sm outline-hidden disabled:cursor-not-allowed disabled:opacity-50',
78             className,
79           )}
80           {...props}
81         />
82         <InputGroupAddon>
```

```

82         <SearchIcon className="size-4 shrink-0 opacity-50" />
83     </InputGroupAddon>
84 </InputGroup>
85 </div>
86 );
87 }
88
89 function CommandList({
90     className,
91     ...props
92 }): React.ComponentProps<typeof CommandPrimitive.List> {
93     return (
94         <CommandPrimitive.List
95             data-slot="command-list"
96             className={cn(
97                 'no-scrollbar max-h-72 scroll-py-1 outline-none overflow-x-hidden overflow-y-auto',
98                 className,
99             )}
100             {...props}
101         />
102     );
103 }
104
105 function CommandEmpty({
106     className,
107     ...props
108 }): React.ComponentProps<typeof CommandPrimitive.Empty> {
109     return (
110         <CommandPrimitive.Empty
111             data-slot="command-empty"
112             className={cn('py-6 text-center text-sm', className)}
113             {...props}
114         />
115     );
116 }
117
118 function CommandGroup({
119     className,
120     ...props
121 }): React.ComponentProps<typeof CommandPrimitive.Group> {
122     return (
123         <CommandPrimitive.Group
124             data-slot="command-group"
125             className={cn(
126                 'text-foreground **:[[cmdk-group-heading]]:text-muted-foreground overflow-hidden p-1 **:[[cmdk-
group-heading]]:px-2 **:[[cmdk-group-heading]]:py-1.5 **:[[cmdk-group-heading]]:text-xs **:[[cmdk-group-
heading]]:font-medium',
127                 className,
128             )}
129             {...props}
130         />
131     );
132 }
133
134 function CommandSeparator({
135     className,
136     ...props
137 }): React.ComponentProps<typeof CommandPrimitive.Separator> {
138     return (
139         <CommandPrimitive.Separator
140             data-slot="command-separator"
141             className={cn('bg-border -mx-1 h-px', className)}
142             {...props}
143         />
144     );
145 }
146
147 function CommandItem({
148     className,
149     children,
150     ...props
151 }): React.ComponentProps<typeof CommandPrimitive.Item> {
152     return (
153         <CommandPrimitive.Item
154             data-slot="command-item"
155             className={cn(
156                 "data-selected:bg-muted data-selected:text-foreground data-selected:*:[svg]:text-foreground
relative flex cursor-default items-center gap-2 rounded-sm px-2 py-1.5 text-sm outline-hidden select-
none in-data-[slot=dialog-content]:rounded-lg! [&_svg:not([class='size-'])]:size-4 group/command-item
data-[disabled=true]:pointer-events-none data-[disabled=true]:opacity-50 [&_svg]:pointer-events-none
[&_svg]:shrink-0",
157                 className,
158             )}
159             {...props}
160         >

```

```

161     {children}
162     <CheckIcon className="ml-auto opacity-0 group-has-data-[slot=command-shortcut]/command-item:hidden
group-data-[checked=true]/command-item:opacity-100" />
163   </CommandPrimitive.Item>
164 );
165 }
166
167 function CommandShortcut({
168   className,
169   ...props
170 }: React.ComponentProps<'span'>) {
171   return (
172     <span
173       data-slot="command-shortcut"
174       className={cn(
175         'text-muted-foreground group-data-selected/command-item:text-foreground ml-auto text-xs
tracking-widest',
176         className,
177       )}
178       {...props}
179     />
180   );
181 }
182
183 export {
184   Command,
185   CommandDialog,
186   CommandInput,
187   CommandList,
188   CommandEmpty,
189   CommandGroup,
190   CommandItem,
191   CommandShortcut,
192   CommandSeparator,
193 };

```

components/ui/context-menu.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { ContextMenu as ContextMenuPrimitive } from '@base-ui/react/context-menu';
5
6 import { cn } from '@lib/utls';
7 import { ChevronRightIcon, CheckIcon } from 'lucide-react';
8
9 function ContextMenu({ ...props }: ContextMenuPrimitive.Root.Props) {
10   return <ContextMenuPrimitive.Root data-slot="context-menu" {...props} />;
11 }
12
13 function ContextMenuPortal({ ...props }: ContextMenuPrimitive.Portal.Props) {
14   return (
15     <ContextMenuPrimitive.Portal data-slot="context-menu-portal" {...props} />
16   );
17 }
18
19 function ContextMenuTrigger({
20   className,
21   ...props
22 }: ContextMenuPrimitive.Trigger.Props) {
23   return (
24     <ContextMenuPrimitive.Trigger
25       data-slot="context-menu-trigger"
26       className={cn('select-none', className)}
27       {...props}
28     />
29   );
30 }
31
32 function ContextMenuContent({
33   className,
34   align = 'start',
35   alignOffset = 4,
36   side = 'right',
37   sideOffset = 0,
38   ...props
39 }: ContextMenuPrimitive.Popup.Props &
40   Pick<
41     ContextMenuPrimitive.Positioner.Props,
42     'align' | 'alignOffset' | 'side' | 'sideOffset'
43   >) {
44   return (
45     <ContextMenuPrimitive.Portal>
46       <ContextMenuPrimitive.Positioner
47         className="isolate z-50 outline-none"
48         align={align}
49         alignOffset={alignOffset}
50         side={side}
51         sideOffset={sideOffset}
52       >
53         <ContextMenuPrimitive.Popup
54           data-slot="context-menu-content"
55           className={cn(
56             'data-open:animate-in data-closed:animate-out data-closed:fade-out-0 data-open:fade-in-0
57             data-closed:zoom-out-95 data-open:zoom-in-95 data-[side=bottom]:slide-in-from-top-2
58             data-[side=left]:slide-in-from-right-2 data-[side=right]:slide-in-from-left-2 data-[side=top]:slide-in-
59             from-bottom-2 ring-foreground/10 bg-popover text-popover-foreground min-w-36 rounded-lg p-1 shadow-md
60             ring-1 duration-100 data-[side=inline-start]:slide-in-from-right-2 data-[side=inline-end]:slide-in-from-
61             left-2 cn-menu-target cn-menu-translucent z-50 max-h-(--available-height) origin-(--transform-origin)
62             overflow-x-hidden overflow-y-auto outline-none',
63             className,
64           )}
65         >
66           <ContextMenuPrimitive.Group data-slot="context-menu-group" {...props} />
67         </ContextMenuPrimitive.Popup>
68       </ContextMenuPrimitive.Positioner>
69     </ContextMenuPrimitive.Portal>
70   );
71 }
72
73 function ContextMenuLabel({
74   className,
75   inset,
76   ...props
77 }: ContextMenuPrimitive.GroupLabel.Props & {
```

```

77   inset?: boolean;
78 }) {
79   return (
80     <ContextMenuPrimitive.GroupLabel
81       data-slot="context-menu-label"
82       data-inset={inset}
83       className={cn(
84         'text-muted-foreground px-1.5 py-1 text-xs font-medium data-inset:pl-7',
85         className,
86       )}
87       {...props}
88     />
89   );
90 }
91
92 function ContextMenuItem({
93   className,
94   inset,
95   variant = 'default',
96   ...props
97 }: ContextMenuPrimitive.Item.Props & {
98   inset?: boolean;
99   variant?: 'default' | 'destructive';
100 }) {
101   return (
102     <ContextMenuPrimitive.Item
103       data-slot="context-menu-item"
104       data-inset={inset}
105       data-variant={variant}
106       className={cn(
107         "focus:bg-accent focus:text-accent-foreground data-[variant=destructive]:text-destructive
108         data-[variant=destructive]:focus:bg-destructive/10 dark:data-[variant=destructive]:focus:bg-
109         destructive/20 data-[variant=destructive]:focus:text-destructive
110         data-[variant=destructive]:*[svg]:text-destructive focus:*[svg]:text-accent-foreground gap-1.5
111         rounded-md px-1.5 py-1 text-sm data-inset:pl-7 [&_svg:not([class*='size-'])]:size-4 group/context-menu-
112         item relative flex cursor-default items-center outline-hidden select-none data-disabled:pointer-events-
113         none data-disabled:opacity-50 [&_svg]:pointer-events-none [&_svg]:shrink-0",
114         className,
115       )}
116       {...props}
117     />
118   );
119 }
120
121 function ContextMenuSub({ ...props }: ContextMenuPrimitive.SubmenuRoot.Props) {
122   return (
123     <ContextMenuPrimitive.SubmenuRoot data-slot="context-menu-sub" {...props} />
124   );
125 }
126
127 function ContextMenuSubTrigger({
128   className,
129   inset,
130   children,
131   ...props
132 }: ContextMenuPrimitive.SubmenuTrigger.Props & {
133   inset?: boolean;
134 }) {
135   return (
136     <ContextMenuPrimitive.SubmenuTrigger
137       data-slot="context-menu-sub-trigger"
138       data-inset={inset}
139       className={cn(
140         "focus:bg-accent focus:text-accent-foreground data-open:bg-accent data-open:text-accent-
141         foreground gap-1.5 rounded-md px-1.5 py-1 text-sm data-inset:pl-7 [&_svg:not([class*='size-'])]:size-4
142         flex cursor-default items-center outline-hidden select-none [&_svg]:pointer-events-none
143         [&_svg]:shrink-0",
144         className,
145       )}
146       {...props}
147       >
148         {children}
149         <ChevronRightIcon className="cn-rtl-flip ml-auto" />
150       </ContextMenuPrimitive.SubmenuTrigger>
151     );
152 }
153
154 function ContextMenuSubContent({
155   ...props
156 }: React.ComponentProps<typeof ContextMenuContent>) {
157   return (
158     <ContextMenuContent
159       data-slot="context-menu-sub-content"
160       className="shadow-lg cn-menu-target cn-menu-translucent"
161       side="right"
162     />
163   );
164 }

```

```

153     {...props}
154   />
155 );
156 }
157
158 function ContextMenuCheckboxItem({
159   className,
160   children,
161   checked,
162   inset,
163   ...props
164 }: ContextMenuPrimitive.CheckboxItem.Props & {
165   inset?: boolean;
166 }) {
167   return (
168     <ContextMenuPrimitive.CheckboxItem
169       data-slot="context-menu-checkbox-item"
170       data-inset={inset}
171       className={cn(
172         "focus:bg-accent focus:text-accent-foreground gap-1.5 rounded-md py-1 pr-8 pl-1.5 text-sm data-
inset:pl-7 [&_svg:not([class*='size-']):size-4 relative flex cursor-default items-center outline-hidden
select-none data-disabled:pointer-events-none data-disabled:opacity-50 [&_svg]:pointer-events-none
[&_svg]:shrink-0",
173         className,
174       )}
175       checked={checked}
176       {...props}
177     >
178       <span className="absolute right-2 pointer-events-none">
179         <ContextMenuPrimitive.CheckboxItemIndicator>
180           <CheckIcon />
181         </ContextMenuPrimitive.CheckboxItemIndicator>
182       </span>
183       {children}
184     </ContextMenuPrimitive.CheckboxItem>
185   );
186 }
187
188 function ContextMenuRadioGroup({
189   ...props
190 }: ContextMenuPrimitive.RadioGroup.Props) {
191   return (
192     <ContextMenuPrimitive.RadioGroup
193       data-slot="context-menu-radio-group"
194       {...props}
195     />
196   );
197 }
198
199 function ContextMenuRadioItem({
200   className,
201   children,
202   inset,
203   ...props
204 }: ContextMenuPrimitive.RadioItem.Props & {
205   inset?: boolean;
206 }) {
207   return (
208     <ContextMenuPrimitive.RadioItem
209       data-slot="context-menu-radio-item"
210       data-inset={inset}
211       className={cn(
212         "focus:bg-accent focus:text-accent-foreground gap-1.5 rounded-md py-1 pr-8 pl-1.5 text-sm data-
inset:pl-7 [&_svg:not([class*='size-']):size-4 relative flex cursor-default items-center outline-hidden
select-none data-disabled:pointer-events-none data-disabled:opacity-50 [&_svg]:pointer-events-none
[&_svg]:shrink-0",
213         className,
214       )}
215       {...props}
216     >
217       <span className="absolute right-2 pointer-events-none">
218         <ContextMenuPrimitive.RadioItemIndicator>
219           <CheckIcon />
220         </ContextMenuPrimitive.RadioItemIndicator>
221       </span>
222       {children}
223     </ContextMenuPrimitive.RadioItem>
224   );
225 }
226
227 function ContextMenuSeparator({
228   className,
229   ...props
230 }: ContextMenuPrimitive.Separator.Props) {
231   return (

```

```

232     <ContextMenuPrimitive.Separator
233       data-slot="context-menu-separator"
234       className={cn('bg-border -mx-1 my-1 h-px', className)}
235       {...props}
236     />
237   );
238 }
239
240 function ContextMenuShortcut({
241   className,
242   ...props
243 }: React.ComponentProps<'span'>) {
244   return (
245     <span
246       data-slot="context-menu-shortcut"
247       className={cn(
248         'text-muted-foreground group-focus/context-menu-item:text-accent-foreground ml-auto text-xs
tracking-widest',
249         className,
250       )}
251       {...props}
252     />
253   );
254 }
255
256 export {
257   ContextMenu,
258   ContextMenuTrigger,
259   ContextMenuContent,
260   ContextMenuItem,
261   ContextMenuCheckboxItem,
262   ContextMenuRadioItem,
263   ContextMenuLabel,
264   ContextMenuSeparator,
265   ContextMenuShortcut,
266   ContextMenuGroup,
267   ContextMenuPortal,
268   ContextMenuSub,
269   ContextMenuSubContent,
270   ContextMenuSubTrigger,
271   ContextMenuRadioGroup,
272 };

```

components/ui/dialog.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { Dialog as DialogPrimitive } from '@base-ui/react/dialog';
5
6 import { cn } from '@lib/utls';
7 import { Button } from '@components/ui/button';
8 import { XIcon } from 'lucide-react';
9
10 function Dialog({ ...props }: DialogPrimitive.Root.Props) {
11   return <DialogPrimitive.Root data-slot="dialog" {...props} />;
12 }
13
14 function DialogTrigger({ ...props }: DialogPrimitive.Trigger.Props) {
15   return <DialogPrimitive.Trigger data-slot="dialog-trigger" {...props} />;
16 }
17
18 function DialogPortal({ ...props }: DialogPrimitive.Portal.Props) {
19   return <DialogPrimitive.Portal data-slot="dialog-portal" {...props} />;
20 }
21
22 function DialogClose({ ...props }: DialogPrimitive.Close.Props) {
23   return <DialogPrimitive.Close data-slot="dialog-close" {...props} />;
24 }
25
26 function DialogOverlay({
27   className,
28   ...props
29 }: DialogPrimitive.Backdrop.Props) {
30   return (
31     <DialogPrimitive.Backdrop
32       data-slot="dialog-overlay"
33       className={cn(
34         'fixed inset-0 isolate z-50 bg-black/10 duration-100 supports-backdrop-filter:backdrop-blur-xs
35 data-open:animate-in data-open:fade-in-0 data-closed:animate-out data-closed:fade-out-0',
36         className,
37       )}
38       {...props}
39     />
40   );
41 }
42
43 function DialogContent({
44   className,
45   children,
46   showCloseButton = true,
47   ...props
48 }: DialogPrimitive.Popup.Props & {
49   showCloseButton?: boolean;
50 }) {
51   return (
52     <DialogPortal>
53       <DialogOverlay />
54       <DialogPrimitive.Popup
55         data-slot="dialog-content"
56         className={cn(
57           'fixed top-1/2 left-1/2 z-50 grid w-full max-w-[calc(100%-2rem)] -translate-x-1/2
58 -translate-y-1/2 gap-4 rounded-xl bg-popover p-4 text-sm text-popover-foreground ring-1 ring-
59 foreground/10 duration-100 outline-none sm:max-w-sm data-open:animate-in data-open:fade-in-0 data-
60 open:zoom-in-95 data-closed:animate-out data-closed:fade-out-0 data-closed:zoom-out-95',
61           className,
62         )}
63       >
64         {children}
65         {showCloseButton && (
66           <DialogPrimitive.Close
67             data-slot="dialog-close"
68             render={
69               <Button
70                 variant="ghost"
71                 className="absolute top-2 right-2"
72                 size="icon-sm"
73               />
74             }
75           >
76             <XIcon />
77             <span className="sr-only">Close</span>
78           </DialogPrimitive.Close>
79         )}
80       </DialogPrimitive.Popup>
81     </DialogPortal>
82   );
83 }
```

```

79   });
80 }
81
82 function DialogHeader({ className, ...props }: React.ComponentProps<'div'>) {
83   return (
84     <div
85       data-slot="dialog-header"
86       className={cn('flex flex-col gap-2', className)}
87       {...props}
88     />
89   );
90 }
91
92 function DialogFooter({
93   className,
94   showCloseButton = false,
95   children,
96   ...props
97 }: React.ComponentProps<'div'> & {
98   showCloseButton?: boolean;
99 }) {
100   return (
101     <div
102       data-slot="dialog-footer"
103       className={cn(
104         '-mx-4 -mb-4 flex flex-col-reverse gap-2 rounded-b-xl border-t bg-muted/50 p-4 sm:flex-row
sm:justify-end',
105         className,
106       )}
107       {...props}
108     >
109       {children}
110       {showCloseButton && (
111         <DialogPrimitive.Close render={<Button variant="outline" />}>
112           Close
113         </DialogPrimitive.Close>
114       )}
115     </div>
116   );
117 }
118
119 function DialogTitle({ className, ...props }: DialogPrimitive.Title.Props) {
120   return (
121     <DialogPrimitive.Title
122       data-slot="dialog-title"
123       className={cn(
124         'font-heading text-base leading-none font-medium',
125         className,
126       )}
127       {...props}
128     />
129   );
130 }
131
132 function DialogDescription({
133   className,
134   ...props
135 }: DialogPrimitive.Description.Props) {
136   return (
137     <DialogPrimitive.Description
138       data-slot="dialog-description"
139       className={cn(
140         'text-sm text-muted-foreground *:[a]:underline *:[a]:underline-offset-3 *:[a]:hover:text-
foreground',
141         className,
142       )}
143       {...props}
144     />
145   );
146 }
147
148 export {
149   Dialog,
150   DialogClose,
151   DialogContent,
152   DialogDescription,
153   DialogFooter,
154   DialogHeader,
155   DialogOverlay,
156   DialogPortal,
157   DialogTitle,
158   DialogTrigger,
159 };

```

components/ui/direction.tsx

```
1 'use client';
2
3 export {
4   DirectionProvider,
5   useDirection,
6 } from '@base-ui/react/direction-provider';
```

components/ui/drawer.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { Drawer as DrawerPrimitive } from '@base-ui/react/drawer';
5
6 import { cn } from '@lib/utils';
7
8 type DrawerContextProps = {
9   hasSnapPoints: boolean;
10   modal: DrawerPrimitive.Root.Props['modal'];
11   showSwipeHandle: boolean;
12   swipeDirection: NonNullable<DrawerPrimitive.Root.Props['swipeDirection']>;
13 };
14
15 const DrawerContext = React.createContext<DrawerContextProps | null>(null);
16
17 function useDrawer() {
18   const context = React.useContext(DrawerContext);
19
20   if (!context) {
21     throw new Error('useDrawer must be used within a Drawer.');
```

```

78     {...props}
79   />
80   );
81 }
82
83 function DrawerSwipeHandle({
84   className,
85   ...props
86 }: React.ComponentProps<'div'>) {
87   return (
88     <div
89       data-slot="drawer-swipe-handle"
90       aria-hidden="true"
91       className={cn(
92         'after:block after:shrink-0 after:rounded-full after:bg-muted group-data-[swipe-axis=y]/drawer-
popup:h-3 group-data-[swipe-axis=y]/drawer-popup:w-full group-data-[swipe-axis=y]/drawer-popup:justify-
center group-data-[swipe-axis=y]/drawer-popup:after:h-1 group-data-[swipe-axis=y]/drawer-
popup:after:w-24 group-data-[swipe-direction=down]/drawer-popup:items-end group-data-[swipe-
direction=up]/drawer-popup:items-start group-data-[swipe-axis=x]/drawer-popup:h-full group-data-[swipe-
axis=x]/drawer-popup:w-3 group-data-[swipe-axis=x]/drawer-popup:items-center group-data-[swipe-
axis=x]/drawer-popup:after:h-24 group-data-[swipe-axis=x]/drawer-popup:after:w-1 group-data-[swipe-
direction=left]/drawer-popup:justify-start group-data-[swipe-direction=right]/drawer-popup:justify-end
relative z-10 flex shrink-0 cursor-grab transition-opacity duration-200 group-data-nested-drawer-
open/drawer-popup:opacity-0 group-data-nested-drawer-swiping/drawer-popup:opacity-100 group-data-[swipe-
direction=left]/drawer-popup:order-last group-data-[swipe-direction=up]/drawer-popup:order-last
active:cursor-grabbing',
93         className,
94       )}
95     {...props}
96   />
97   );
98 }
99
100 function DrawerContent({
101   className,
102   children,
103   ...props
104 }: DrawerPrimitive.Popup.Props) {
105   const { hasSnapPoints, modal, showSwipeHandle, swipeDirection } = useDrawer();
106   const swipeAxis =
107     swipeDirection === 'down' || swipeDirection === 'up' ? 'y' : 'x';
108
109   return (
110     <DrawerPortal data-slot="drawer-portal">
111       {modal === true && (
112         <DrawerOverlay data-snap-points={hasSnapPoints ? '' : undefined} />
113       )}
114       <DrawerPrimitive.Viewport
115         data-slot="drawer-viewport"
116         data-modal={modal}
117         className="pointer-events-none fixed inset-0 z-50 select-none data-[modal=true]:pointer-events-
auto"
118       >
119         <DrawerPrimitive.Popup
120           data-slot="drawer-popup"
121           data-swipe-axis={swipeAxis}
122           data-snap-points={hasSnapPoints ? '' : undefined}
123           className={cn(
124             // Base.
125             'bg-popover text-popover-foreground text-sm data-[swipe-direction=down]:rounded-t-xl
data-[swipe-direction=down]:border-t data-[swipe-direction=left]:rounded-r-xl data-[swipe-
direction=left]:border-r data-[swipe-direction=right]:rounded-l-xl data-[swipe-direction=right]:border-l
data-[swipe-direction=up]:rounded-b-xl data-[swipe-direction=up]:border-b group/drawer-popup pointer-
events-auto fixed z-50 m-(-drawer-inset,0px) flex h-(-drawer-content-height) max-h-(-drawer-content-
max-height,none) min-h-0 w-(-drawer-content-width,auto)
transform-[translate3d(var(--translate-x,0px),var(--translate-y,0px),0)_scale(var(--stack-scale))] flex-
col transition-[transform,height,opacity,filter] duration-450 ease-[cubic-bezier(0.22,1,0.36,1)] will-
change-transform outline-none select-none [interpolate-size:allow-keywords]',
126             // Nested.
127             'data-nested-drawer-open:overflow-hidden data-nested-drawer-open:brightness-95',
128             // Bleed.
129             'after:pointer-events-none after:absolute after:bg-(-drawer-bleed-background,var(--color-
popover)) data-[swipe-axis=x]:after:inset-y-0 data-[swipe-axis=x]:after:w-(-bleed) data-[swipe-
axis=y]:after:inset-x-0 data-[swipe-axis=y]:after:h-(-bleed) data-[swipe-direction=down]:after:top-full
data-[swipe-direction=left]:after:right-full data-[swipe-direction=right]:after:left-full data-[swipe-
direction=up]:after:bottom-full',
130             // Sizing.
131             '[--drawer-content-height:var(--drawer-height,auto)] data-[swipe-axis=x]:[--drawer-content-
width:75%] data-[swipe-axis=y]:[--drawer-content-max-height:calc(100dvh-6rem)] data-[swipe-axis=y]:data-
snap-points:[--drawer-content-height:100dvh] data-[swipe-axis=x]:sm:[--drawer-content-width:24rem]',
132             // Stack.
133             '[--bleed:3rem] [--peek:1rem] [--stack-height:var(--drawer-frontmost-height,var(--drawer-
height,0px))] [--stack-peek-offset:max(0px,calc((var(--nested-drawers)-var(--stack-
progress))*var(--peek)))] [--stack-progress:clamp(0,var(--drawer-swipe-progress),1)] [--stack-scale-
base:max(0,calc(1-(var(--nested-drawers)*var(--stack-step)))] [--stack-scale:clamp(0,calc(var(--stack-

```

```

scale-base)+(var(--stack-step)*var(--stack-progress))),1)] [--stack-shrink:calc(1-var(--stack-scale))]]
[--stack-step:0.05]',
134     // Transitions.
135     'data-ending-style:transform-(--closed-transform) data-ending-style:opacity-[0.9999] data-
ending-style:duration-[calc(var(--drawer-swipe-strength)*400ms)] data-nested-drawer-swiping:duration-0
data-ending-style:data-nested-drawer-swiping:duration-[calc(var(--drawer-swipe-strength)*400ms)] data-
starting-style:transform-(--closed-transform) data-swiping:duration-0 data-ending-style:data-
swiping:duration-[calc(var(--drawer-swipe-strength)*400ms)]',
136     // Axis: y.
137     'data-[swipe-axis=y]:inset-x-0 data-[swipe-axis=y]:data-nested-drawer-open:h-(--stack-
height)',
138     // Axis: x.
139     'data-[swipe-axis=x]:inset-y-0 data-[swipe-axis=x]:flex-row',
140     // Direction: down.
141     'data-[swipe-direction=down]:bottom-0 data-[swipe-direction=down]:origin-bottom data-[swipe-
direction=down]:[--closed-transform:translate3d(0,calc(100%+var(--drawer-inset,0px)+2px),0)] data-[swipe-
direction=down]:[--translate-y:calc(var(--drawer-snap-point-offset,0px)+var(--drawer-swipe-
movement-y)-var(--stack-peek-offset)-(var(--stack-shrink)*var(--stack-height)))]',
142     // Direction: up.
143     'data-[swipe-direction=up]:top-0 data-[swipe-direction=up]:origin-top data-[swipe-
direction=up]:[--closed-transform:translate3d(0,calc(-100%-var(--drawer-inset,0px)-2px),0)] data-[swipe-
direction=up]:[--translate-y:calc(var(--drawer-snap-point-offset,0px)+var(--drawer-swipe-
movement-y)+var(--stack-peek-offset)+(var(--stack-shrink)*var(--stack-height)))]',
144     // Direction: left.
145     'data-[swipe-direction=left]:left-0 data-[swipe-direction=left]:origin-left data-[swipe-
direction=left]:[--closed-transform:translate3d(calc(-100%-var(--drawer-inset,0px)-2px),0,0)]
data-[swipe-direction=left]:[--translate-x:calc(var(--drawer-swipe-movement-x)+var(--stack-peek-
offset)+(var(--stack-shrink)*100%))]',
146     // Direction: right.
147     'data-[swipe-direction=right]:right-0 data-[swipe-direction=right]:origin-right data-[swipe-
direction=right]:[--closed-transform:translate3d(calc(100%+var(--drawer-inset,0px)+2px),0,0)]
data-[swipe-direction=right]:[--translate-x:calc(var(--drawer-swipe-movement-x)-var(--stack-peek-
offset)-(var(--stack-shrink)*100%))]',
148     className,
149   })
150   {...props}
151   >
152   {showSwipeHandle && <DrawerSwipeHandle />}
153   <DrawerPrimitive.Content
154     data-slot="drawer-content"
155     className={cn(
156       'flex min-h-0 flex-1 flex-col overflow-hidden overscroll-contain rounded-[inherit]
transition-opacity duration-300 ease-[cubic-bezier(0.45,1.005,0,1.005)] select-text group-data-nested-
drawer-open/drawer-popup:opacity-0 group-data-nested-drawer-swiping/drawer-popup:opacity-100 group-data-
swiping/drawer-popup:select-none',
157     )}
158   >
159     {children}
160   </DrawerPrimitive.Content>
161   </DrawerPrimitive.Popup>
162   </DrawerPrimitive.Viewport>
163   </DrawerPortal>
164 );
165 }
166
167 function DrawerHeader({ className, ...props }: React.ComponentProps<'div'>) {
168   return (
169     <div
170       data-slot="drawer-header"
171       className={cn(
172         'gap-0.5 p-4 pb-0 md:gap-0.5 md:text-left flex shrink-0 flex-col group-data-[swipe-
axis=y]/drawer-popup:text-center',
173       className,
174     )}
175     {...props}
176   />
177 );
178 }
179
180 function DrawerFooter({ className, ...props }: React.ComponentProps<'div'>) {
181   return (
182     <div
183       data-slot="drawer-footer"
184       className={cn('gap-2 p-4 pt-0 mt-auto flex shrink-0 flex-col', className)}
185       {...props}
186     />
187 );
188 }
189
190 function DrawerTitle({ className, ...props }: DrawerPrimitive.Title.Props) {
191   return (
192     <DrawerPrimitive.Title
193       data-slot="drawer-title"
194       className={cn(
195         'text-foreground text-base font-medium cn-font-heading',

```

```
196         className,
197     )}
198     {...props}
199   />
200 );
201 }
202
203 function DrawerDescription({
204   className,
205   ...props
206 }: DrawerPrimitive.Description.Props) {
207   return (
208     <DrawerPrimitive.Description
209       data-slot="drawer-description"
210       className={cn('text-muted-foreground text-sm text-balance', className)}
211       {...props}
212     />
213   );
214 }
215
216 export {
217   Drawer,
218   DrawerPortal,
219   DrawerOverlay,
220   DrawerSwipeHandle,
221   DrawerTrigger,
222   DrawerClose,
223   DrawerContent,
224   DrawerHeader,
225   DrawerFooter,
226   DrawerTitle,
227   DrawerDescription,
228 };
```

components/ui/dropdown-menu.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { Menu as MenuPrimitive } from '@base-ui/react/menu';
5
6 import { cn } from '@lib/utils';
7 import { ChevronRightIcon, CheckIcon } from 'lucide-react';
8
9 function DropdownMenu({ ...props }: MenuPrimitive.Root.Props) {
10   return <MenuPrimitive.Root data-slot="dropdown-menu" {...props} />;
11 }
12
13 function DropdownMenuPortal({ ...props }: MenuPrimitive.Portal.Props) {
14   return <MenuPrimitive.Portal data-slot="dropdown-menu-portal" {...props} />;
15 }
16
17 function DropdownMenuTrigger({ ...props }: MenuPrimitive.Trigger.Props) {
18   return <MenuPrimitive.Trigger data-slot="dropdown-menu-trigger" {...props} />;
19 }
20
21 function DropdownMenuContent({
22   align = 'start',
23   alignOffset = 0,
24   side = 'bottom',
25   sideOffset = 4,
26   className,
27   ...props
28 }: MenuPrimitive.Popup.Props &
29   Pick<
30     MenuPrimitive.Positioner.Props,
31     'align' | 'alignOffset' | 'side' | 'sideOffset'
32   >() {
33   return (
34     <MenuPrimitive.Portal>
35       <MenuPrimitive.Positioner
36         className="isolate z-50 outline-none"
37         align={align}
38         alignOffset={alignOffset}
39         side={side}
40         sideOffset={sideOffset}
41       >
42         <MenuPrimitive.Popup
43           data-slot="dropdown-menu-content"
44           className={cn(
45             'z-50 max-h-(--available-height) w-(--anchor-width) min-w-32 origin-(--transform-origin)
46             overflow-x-hidden overflow-y-auto rounded-lg bg-popover p-1 text-popover-foreground shadow-md ring-1
47             ring-foreground/10 duration-100 outline-none data-[side=bottom]:slide-in-from-top-2 data-[side=inline-
48             end]:slide-in-from-left-2 data-[side=inline-start]:slide-in-from-right-2 data-[side=left]:slide-in-from-
49             right-2 data-[side=right]:slide-in-from-left-2 data-[side=top]:slide-in-from-bottom-2 data-open:animate-
50             in data-open:fade-in-0 data-open:zoom-in-95 data-closed:animate-out data-closed:overflow-hidden data-
51             closed:fade-out-0 data-closed:zoom-out-95',
52             className,
53           )}
54           {...props}
55         />
56       </MenuPrimitive.Positioner>
57     </MenuPrimitive.Portal>
58   );
59 }
60
61 function DropdownMenuGroup({ ...props }: MenuPrimitive.Group.Props) {
62   return <MenuPrimitive.Group data-slot="dropdown-menu-group" {...props} />;
63 }
64
65 function DropdownMenuLabel({
66   className,
67   inset,
68   ...props
69 }: MenuPrimitive.GroupLabel.Props & {
70   inset?: boolean;
71 }) {
72   return (
73     <MenuPrimitive.GroupLabel
74       data-slot="dropdown-menu-label"
75       data-inset={inset}
76       className={cn(
77         'px-1.5 py-1 text-xs font-medium text-muted-foreground data-inset:pl-7',
78         className,
79       )}
80       {...props}
81     />
82   );
83 }
```

```

77 }
78
79 function DropdownMenuItem({
80   className,
81   inset,
82   variant = 'default',
83   ...props
84 }: MenuPrimitive.Item.Props & {
85   inset?: boolean;
86   variant?: 'default' | 'destructive';
87 }) {
88   return (
89     <MenuPrimitive.Item
90       data-slot="dropdown-menu-item"
91       data-inset={inset}
92       data-variant={variant}
93       className={cn(
94         "group/dropdown-menu-item relative flex cursor-default items-center gap-1.5 rounded-md px-1.5
py-1 text-sm outline-hidden select-none focus:bg-accent focus:text-accent-foreground not-
data-[variant=destructive]:focus:**:text-accent-foreground data-inset:pl-7
data-[variant=destructive]:text-destructive data-[variant=destructive]:focus:bg-destructive/10
data-[variant=destructive]:focus:text-destructive dark:data-[variant=destructive]:focus:bg-
destructive/20 data-disabled:pointer-events-none data-disabled:opacity-50 [&_svg]:pointer-events-none
[&_svg]:shrink-0 [&_svg]:not([class*=size-']):size-4 data-[variant=destructive]:*:[svg]:text-
destructive",
95         className,
96       )}
97       {...props}
98     />
99   );
100 }
101
102 function DropdownMenuSub({ ...props }: MenuPrimitive.SubmenuRoot.Props) {
103   return <MenuPrimitive.SubmenuRoot data-slot="dropdown-menu-sub" {...props} />;
104 }
105
106 function DropdownMenuSubTrigger({
107   className,
108   inset,
109   children,
110   ...props
111 }: MenuPrimitive.SubmenuTrigger.Props & {
112   inset?: boolean;
113 }) {
114   return (
115     <MenuPrimitive.SubmenuTrigger
116       data-slot="dropdown-menu-sub-trigger"
117       data-inset={inset}
118       className={cn(
119         "flex cursor-default items-center gap-1.5 rounded-md px-1.5 py-1 text-sm outline-hidden select-
none focus:bg-accent focus:text-accent-foreground not-data-[variant=destructive]:focus:**:text-accent-
foreground data-inset:pl-7 data-popup-open:bg-accent data-popup-open:text-accent-foreground data-
open:bg-accent data-open:text-accent-foreground [&_svg]:pointer-events-none [&_svg]:shrink-0
[&_svg]:not([class*=size-']):size-4",
120         className,
121       )}
122       {...props}
123     >
124       {children}
125       <ChevronRightIcon className="ml-auto" />
126     </MenuPrimitive.SubmenuTrigger>
127   );
128 }
129
130 function DropdownMenuSubContent({
131   align = 'start',
132   alignOffset = -3,
133   side = 'right',
134   sideOffset = 0,
135   className,
136   ...props
137 }: React.ComponentProps<typeof DropdownMenuContent>) {
138   return (
139     <DropdownMenuContent
140       data-slot="dropdown-menu-sub-content"
141       className={cn(
142         'w-auto min-w-[96px] rounded-lg bg-popover p-1 text-popover-foreground shadow-lg ring-1 ring-
foreground/10 duration-100 data-[side=bottom]:slide-in-from-top-2 data-[side=left]:slide-in-from-right-2
data-[side=right]:slide-in-from-left-2 data-[side=top]:slide-in-from-bottom-2 data-open:animate-in
data-open:fade-in-0 data-open:zoom-in-95 data-closed:animate-out data-closed:fade-out-0 data-
closed:zoom-out-95',
143         className,
144       )}
145       align={align}
146       alignOffset={alignOffset}

```

```

147     side={side}
148     sideOffset={sideOffset}
149     {...props}
150   />
151 );
152 }
153
154 function DropdownMenuCheckboxItem({
155   className,
156   children,
157   checked,
158   inset,
159   ...props
160 }: MenuPrimitive.CheckboxItem.Props & {
161   inset?: boolean;
162 }) {
163   return (
164     <MenuPrimitive.CheckboxItem
165       data-slot="dropdown-menu-checkbox-item"
166       data-inset={inset}
167       className={cn(
168         "relative flex cursor-default items-center gap-1.5 rounded-md py-1 pr-8 pl-1.5 text-sm outline-
hidden select-none focus:bg-accent focus:text-accent-foreground focus:**:text-accent-foreground data-
inset:pl-7 data-disabled:pointer-events-none data-disabled:opacity-50 [&_svg]:pointer-events-none
[&_svg]:shrink-0 [&_svg:not([class*='size-'])]:size-4",
169         className,
170       )}
171       checked={checked}
172       {...props}
173     >
174       <span
175         className="pointer-events-none absolute right-2 flex items-center justify-center"
176         data-slot="dropdown-menu-checkbox-item-indicator"
177       >
178         <MenuPrimitive.CheckboxItemIndicator>
179           <CheckIcon />
180         </MenuPrimitive.CheckboxItemIndicator>
181       </span>
182       {children}
183     </MenuPrimitive.CheckboxItem>
184   );
185 }
186
187 function DropdownMenuRadioGroup({ ...props }: MenuPrimitive.RadioGroup.Props) {
188   return (
189     <MenuPrimitive.RadioGroup
190       data-slot="dropdown-menu-radio-group"
191       {...props}
192     />
193   );
194 }
195
196 function DropdownMenuRadioItem({
197   className,
198   children,
199   inset,
200   ...props
201 }: MenuPrimitive.RadioItem.Props & {
202   inset?: boolean;
203 }) {
204   return (
205     <MenuPrimitive.RadioItem
206       data-slot="dropdown-menu-radio-item"
207       data-inset={inset}
208       className={cn(
209         "relative flex cursor-default items-center gap-1.5 rounded-md py-1 pr-8 pl-1.5 text-sm outline-
hidden select-none focus:bg-accent focus:text-accent-foreground focus:**:text-accent-foreground data-
inset:pl-7 data-disabled:pointer-events-none data-disabled:opacity-50 [&_svg]:pointer-events-none
[&_svg]:shrink-0 [&_svg:not([class*='size-'])]:size-4",
210         className,
211       )}
212       {...props}
213     >
214       <span
215         className="pointer-events-none absolute right-2 flex items-center justify-center"
216         data-slot="dropdown-menu-radio-item-indicator"
217       >
218         <MenuPrimitive.RadioItemIndicator>
219           <CheckIcon />
220         </MenuPrimitive.RadioItemIndicator>
221       </span>
222       {children}
223     </MenuPrimitive.RadioItem>
224   );
225 }

```

```

226
227 function DropdownMenuSeparator({
228   className,
229   ...props
230 }: MenuPrimitive.Separator.Props) {
231   return (
232     <MenuPrimitive.Separator
233       data-slot="dropdown-menu-separator"
234       className={cn('-mx-1 my-1 h-px bg-border', className)}
235       {...props}
236     />
237   );
238 }
239
240 function DropdownMenuShortcut({
241   className,
242   ...props
243 }: React.ComponentProps<'span'>) {
244   return (
245     <span
246       data-slot="dropdown-menu-shortcut"
247       className={cn(
248         'ml-auto text-xs tracking-widest text-muted-foreground group-focus/dropdown-menu-item:text-
249         accent-foreground',
250         className,
251       )}
252       {...props}
253     />
254   );
255 }
256 export {
257   DropdownMenu,
258   DropdownMenuPortal,
259   DropdownMenuTrigger,
260   DropdownMenuContent,
261   DropdownMenuGroup,
262   DropdownMenuLabel,
263   DropdownMenuItem,
264   DropdownMenuCheckboxItem,
265   DropdownMenuRadioGroup,
266   DropdownMenuRadioItem,
267   DropdownMenuSeparator,
268   DropdownMenuShortcut,
269   DropdownMenuSub,
270   DropdownMenuSubTrigger,
271   DropdownMenuSubContent,
272 };

```

components/ui/empty.tsx

```
1 import { cva, type VariantProps } from 'class-variance-authority';
2
3 import { cn } from '@lib/utils';
4
5 function Empty({ className, ...props }: React.ComponentProps<'div'>) {
6   return (
7     <div
8       data-slot="empty"
9       className={cn(
10        'gap-4 rounded-xl border-dashed p-6 flex w-full min-w-0 flex-1 flex-col items-center justify-
11        center text-center text-balance',
12        className,
13      )}
14      {...props}
15    />
16  );
17 }
18
19 function EmptyHeader({ className, ...props }: React.ComponentProps<'div'>) {
20   return (
21     <div
22       data-slot="empty-header"
23       className={cn('gap-2 flex max-w-sm flex-col items-center', className)}
24       {...props}
25     />
26   );
27 }
28
29 const emptyMediaVariants = cva(
30   'mb-2 flex shrink-0 items-center justify-center [&_svg]:pointer-events-none [&_svg]:shrink-0',
31   {
32     variants: {
33       variant: {
34         default: 'bg-transparent',
35         icon: "bg-muted text-foreground flex size-8 shrink-0 items-center justify-center rounded-lg
36         [&_svg:not([class*='size-'])]:size-4",
37       },
38     },
39     defaultVariants: {
40       variant: 'default',
41     },
42   },
43 );
44
45 function EmptyMedia({
46   className,
47   variant = 'default',
48   ...props
49 }: React.ComponentProps<'div'> & VariantProps<typeof emptyMediaVariants>) {
50   return (
51     <div
52       data-slot="empty-icon"
53       data-variant={variant}
54       className={cn(emptyMediaVariants({ variant, className }))}
55       {...props}
56     />
57   );
58 }
59
60 function EmptyTitle({ className, ...props }: React.ComponentProps<'div'>) {
61   return (
62     <div
63       data-slot="empty-title"
64       className={cn(
65        'text-sm font-medium tracking-tight cn-font-heading',
66        className,
67      )}
68       {...props}
69     />
70   );
71 }
72
73 function EmptyDescription({ className, ...props }: React.ComponentProps<'p'>) {
74   return (
75     <div
76       data-slot="empty-description"
77       className={cn(
78        'text-sm/relaxed text-muted-foreground [&a]:underline [&a]:underline-offset-4
79        [&a:hover]:text-primary',
80        className,
81      )}
82       {...props}
83     />
84   );
85 }
```

```
80     />
81   );
82 }
83
84 function EmptyContent({ className, ...props }: React.ComponentProps<'div'>) {
85   return (
86     <div
87       data-slot="empty-content"
88       className={cn(
89         'gap-2.5 text-sm flex w-full max-w-sm min-w-0 flex-col items-center text-balance',
90         className,
91       )}
92       {...props}
93     />
94   );
95 }
96
97 export {
98   Empty,
99   EmptyHeader,
100  EmptyTitle,
101  EmptyDescription,
102  EmptyContent,
103  EmptyMedia,
104 };
```

components/ui/field.tsx

```
1 'use client';
2
3 import { useMemo } from 'react';
4 import { cva, type VariantProps } from 'class-variance-authority';
5
6 import { cn } from '@lib/utils';
7 import { Label } from '@components/ui/label';
8 import { Separator } from '@components/ui/separator';
9
10 function FieldSet({ className, ...props }: React.ComponentProps<'fieldset'>) {
11   return (
12     <fieldset
13       data-slot="field-set"
14       className={cn(
15         'gap-4 has- [>[data-slot=checkbox-group]]:gap-3 has- [>[data-slot=radio-group]]:gap-3 flex flex-
16         col',
17         className,
18       )}
19     />
20   );
21 }
22
23 function FieldLegend({
24   className,
25   variant = 'legend',
26   ...props
27 }: React.ComponentProps<'legend'> & { variant?: 'legend' | 'label' }) {
28   return (
29     <legend
30       data-slot="field-legend"
31       data-variant={variant}
32       className={cn(
33         'mb-1.5 font-medium data-[variant=label]:text-sm data-[variant=legend]:text-base',
34         className,
35       )}
36     />
37   );
38 }
39
40
41 function FieldGroup({ className, ...props }: React.ComponentProps<'div'>) {
42   return (
43     <div
44       data-slot="field-group"
45       className={cn(
46         'gap-5 data-[slot=checkbox-group]:gap-3 *:data-[slot=field-group]:gap-4 group/field-group
47         @container/field-group flex w-full flex-col',
48         className,
49       )}
50     />
51   );
52 }
53
54 const fieldVariants = cva(
55   'data-[invalid=true]:text-destructive gap-2 group/field flex w-full',
56   {
57     variants: {
58       orientation: {
59         vertical: 'flex-col *:w-full [>.sr-only]:w-auto',
60         horizontal:
61           'flex-row items-center has- [>[data-slot=field-content]]:items-start *:data-[slot=field-
62           label]:flex-auto has- [>[data-slot=field-content]]:[>[role=checkbox],[role=radio]]:mt-px',
63         responsive:
64           'flex-col *:w-full @md/field-group:flex-row @md/field-group:items-center @md/field-
65           group: *:w-auto @md/field-group:has- [>[data-slot=field-content]]:items-start @md/field-
66           group: *:data-[slot=field-label]:flex-auto [>.sr-only]:w-auto @md/field-group:has- [>[data-slot=field-
67           content]]:[>[role=checkbox],[role=radio]]:mt-px',
68       },
69     },
70     defaultVariants: {
71       orientation: 'vertical',
72     },
73   },
74 );
75
76 function Field({
77   className,
78   orientation = 'vertical',
79   ...props
80 }: React.ComponentProps<'div'> & VariantProps<typeof fieldVariants>) {
```

```

77   return (
78     <div
79       role="group"
80       data-slot="field"
81       data-orientation={orientation}
82       className={cn(fieldVariants({ orientation }), className)}
83       {...props}
84     />
85   );
86 }
87
88 function FieldContent({ className, ...props }: React.ComponentProps<'div'>) {
89   return (
90     <div
91       data-slot="field-content"
92       className={cn(
93         'gap-0.5 group/field-content flex flex-1 flex-col leading-snug',
94         className,
95       )}
96       {...props}
97     />
98   );
99 }
100
101 function FieldLabel({
102   className,
103   ...props
104 }: React.ComponentProps<typeof Label>) {
105   return (
106     <Label
107       data-slot="field-label"
108       className={cn(
109         'has-data-checked:bg-primary/5 has-data-checked:border-primary/30 dark:has-data-checked:border-
primary/20 dark:has-data-checked:bg-primary/10 gap-2 leading-snug group-
data-[disabled=true]/field:opacity-50 has-[>[data-slot=field]]:rounded-lg has-[>[data-
slot=field]]:border has-[>[data-slot=field]]:not-has-[:disabled,[data-disabled]]:hover:bg-muted/50
has-[>[data-slot=field]]:has-[:focus-visible]:border-ring has-[>[data-slot=field]]:has-[:focus-
visible]:ring-ring/50 has-[>[data-slot=field]]:has-[:focus-visible]:ring-3 *:data-[slot=field]:p-2.5
group/field-label peer/field-label flex w-fit',
110         'has-[>[data-slot=field]]:w-full has-[>[data-slot=field]]:flex-col',
111         className,
112       )}
113       {...props}
114     />
115   );
116 }
117
118 function FieldTitle({ className, ...props }: React.ComponentProps<'div'>) {
119   return (
120     <div
121       data-slot="field-label"
122       className={cn(
123         'gap-2 text-sm font-medium group-data-[disabled=true]/field:opacity-50 flex w-fit items-center',
124         className,
125       )}
126       {...props}
127     />
128   );
129 }
130
131 function FieldDescription({ className, ...props }: React.ComponentProps<'p'>) {
132   return (
133     <p
134       data-slot="field-description"
135       className={cn(
136         'text-muted-foreground text-left text-sm [[data-variant=legend]&]:-mt-1.5 leading-normal font-
normal group-has-data-horizontal/field:text-balance',
137         'last:mt-0 nth-last-2:-mt-1',
138         '[&a]:underline [&a]:underline-offset-4 [&a:hover]:text-primary',
139         className,
140       )}
141       {...props}
142     />
143   );
144 }
145
146 function FieldSeparator({
147   children,
148   className,
149   ...props
150 }: React.ComponentProps<'div'> & {
151   children?: React.ReactNode;
152 }) {
153   return (
154     <div

```

```

155     data-slot="field-separator"
156     data-content={!!children}
157     className={cn(
158       '-my-2 h-5 text-sm group-data-[variant=outline]/field-group:-mb-2 relative',
159       className,
160     )}
161     {...props}
162   >
163     <Separator className="absolute inset-0 top-1/2" />
164     {children && (
165       <span
166         className="text-muted-foreground px-2 relative mx-auto block w-fit bg-background"
167         data-slot="field-separator-content"
168       >
169         {children}
170       </span>
171     )}
172   </div>
173 );
174 }
175
176 function FieldError({
177   className,
178   children,
179   errors,
180   ...props
181 }): React.ComponentProps<'div'> & {
182   errors?: Array<{ message?: string } | undefined>;
183 } {
184   const content = useMemo(() => {
185     if (children) {
186       return children;
187     }
188
189     if (!errors?.length) {
190       return null;
191     }
192
193     const uniqueErrors = [
194       ...new Map(errors.map((error) => [error?.message, error])).values(),
195     ];
196
197     if (uniqueErrors?.length == 1) {
198       return uniqueErrors[0]?.message;
199     }
200
201     return (
202       <ul className="ml-4 flex list-disc flex-col gap-1">
203         {uniqueErrors.map(
204           (error, index) =>
205             error?.message && <li key={index}>{error.message}</li>,
206         )}
207       </ul>
208     );
209   }, [children, errors]);
210
211   if (!content) {
212     return null;
213   }
214
215   return (
216     <div
217       role="alert"
218       data-slot="field-error"
219       className={cn('text-destructive text-sm font-normal', className)}
220       {...props}
221     >
222       {content}
223     </div>
224   );
225 }
226
227 export {
228   Field,
229   FieldLabel,
230   FieldDescription,
231   FieldError,
232   FieldGroup,
233   FieldLegend,
234   FieldSeparator,
235   FieldSet,
236   FieldContent,
237   FieldTitle,
238 };

```

components/ui/hover-card.tsx

```
1 'use client';
2
3 import { PreviewCard as PreviewCardPrimitive } from '@base-ui/react/preview-card';
4
5 import { cn } from '@lib/utils';
6
7 function HoverCard({ ...props }: PreviewCardPrimitive.Root.Props) {
8   return <PreviewCardPrimitive.Root data-slot="hover-card" {...props} />;
9 }
10
11 function HoverCardTrigger({ ...props }: PreviewCardPrimitive.Trigger.Props) {
12   return (
13     <PreviewCardPrimitive.Trigger data-slot="hover-card-trigger" {...props} />
14   );
15 }
16
17 function HoverCardContent({
18   className,
19   side = 'bottom',
20   sideOffset = 4,
21   align = 'center',
22   alignOffset = 4,
23   ...props
24 }: PreviewCardPrimitive.Popup.Props &
25   Pick<
26     PreviewCardPrimitive.Positioner.Props,
27     'align' | 'alignOffset' | 'side' | 'sideOffset'
28   > ) {
29   return (
30     <PreviewCardPrimitive.Portal data-slot="hover-card-portal">
31       <PreviewCardPrimitive.Positioner
32         align={align}
33         alignOffset={alignOffset}
34         side={side}
35         sideOffset={sideOffset}
36         className="isolate z-50"
37       >
38         <PreviewCardPrimitive.Popup
39           data-slot="hover-card-content"
40           className={cn(
41             'data-open:animate-in data-closed:animate-out data-closed:fade-out-0 data-open:fade-in-0
42             data-closed:zoom-out-95 data-open:zoom-in-95 data-[side=bottom]:slide-in-from-top-2
43             data-[side=left]:slide-in-from-right-2 data-[side=right]:slide-in-from-left-2 data-[side=top]:slide-in-
44             from-bottom-2 ring-foreground/10 bg-popover text-popover-foreground w-64 rounded-lg p-2.5 text-sm
45             shadow-md ring-1 duration-100 data-[side=inline-start]:slide-in-from-right-2 data-[side=inline-
46             end]:slide-in-from-left-2 z-50 origin(--transform-origin) outline-hidden',
47             className,
48           )}
49           {...props}
50         />
51       </PreviewCardPrimitive.Positioner>
52     </PreviewCardPrimitive.Portal>
53   );
54 }
55
56 export { HoverCard, HoverCardTrigger, HoverCardContent };
```

components/ui/input-group.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { cva, type VariantProps } from 'class-variance-authority';
5
6 import { cn } from '@lib/utls';
7 import { Button } from '@components/ui/button';
8 import { Input } from '@components/ui/input';
9 import { Textarea } from '@components/ui/textarea';
10
11 function InputGroup({ className, ...props }: React.ComponentProps<'div'>) {
12   return (
13     <div
14       data-slot="input-group"
15       role="group"
16       className={cn(
17         'border-input dark:bg-input/30 has-[[data-slot=input-group-control]:focus-visible]:border-ring
has-[[data-slot=input-group-control]:focus-visible]:ring-ring/50 has-[[data-slot][aria-
invalid=true]]:ring-destructive/20 has-[[data-slot][aria-invalid=true]]:border-destructive
dark:has-[[data-slot][aria-invalid=true]]:ring-destructive/40 has-disabled:bg-input/50 dark:has-
disabled:bg-input/80 h-8 rounded-lg border transition-colors in-data-[slot=combobox-content]:focus-
within:border-inherit in-data-[slot=combobox-content]:focus-within:ring-0 has-disabled:opacity-50
has-[[data-slot=input-group-control]:focus-visible]:ring-3 has-[[data-slot][aria-invalid=true]]:ring-3
has-[>[data-align=block-end]]:h-auto has-[>[data-align=block-end]]:flex-col has-[>[data-align=block-
start]]:h-auto has-[>[data-align=block-start]]:flex-col has-[>[data-align=block-end]]:[&>input]:pt-3
has-[>[data-align=block-start]]:[&>input]:pb-3 has-[>[data-align=inline-end]]:[&>input]:pr-1.5
has-[>[data-align=inline-start]]:[&>input]:pl-1.5 group/input-group relative flex w-full min-w-0 items-
center outline-none has-[>textarea]:h-auto',
18       className,
19     )}
20     {...props}
21   />
22 );
23 }
24
25 const inputGroupAddonVariants = cva(
26   "text-muted-foreground h-auto gap-2 py-1.5 text-sm font-medium group-data-[disabled=true]/input-
group:opacity-50 [&>kbd]:rounded-[calc(var(--radius)-5px)] [&>svg:not([class*='size-'])]:size-4 flex
cursor-text items-center justify-center select-none",
27   {
28     variants: {
29       align: {
30         'inline-start':
31           'pl-2 has-[>button]:ml-[-0.3rem] has-[>kbd]:ml-[-0.15rem] order-first',
32         'inline-end':
33           'pr-2 has-[>button]:mr-[-0.3rem] has-[>kbd]:mr-[-0.15rem] order-last',
34         'block-start':
35           'px-2.5 pt-2 group-has-[>input]/input-group:pt-2 [.border-b]:pb-2 order-first w-full justify-
start',
36         'block-end':
37           'px-2.5 pb-2 group-has-[>input]/input-group:pb-2 [.border-t]:pt-2 order-last w-full justify-
start',
38       },
39     },
40     defaultVariants: {
41       align: 'inline-start',
42     },
43   },
44 );
45
46 function InputGroupAddon({
47   className,
48   align = 'inline-start',
49   ...props
50 }: React.ComponentProps<'div'> & VariantProps<typeof inputGroupAddonVariants>) {
51   return (
52     <div
53       role="group"
54       data-slot="input-group-addon"
55       data-align={align}
56       className={cn(inputGroupAddonVariants({ align }), className)}
57       onClick={(e) => {
58         if ((e.target as HTMLElement).closest('button')) {
59           return;
60         }
61         e.currentTarget.parentElement?.querySelector('input')?.focus();
62       }}
63       {...props}
64     />
65   );
66 }
67
```

```

68 const inputGroupButtonVariants = cva(
69   'gap-2 text-sm flex items-center shadow-none',
70   {
71     variants: {
72       size: {
73         xs: "h-6 gap-1 rounded-[calc(var(--radius)-3px)] px-1.5 [>svg:not([class*='size-'])]:size-3.5",
74         sm: '',
75         'icon-xs':
76           'size-6 rounded-[calc(var(--radius)-3px)] p-0 has-[>svg]:p-0',
77         'icon-sm': 'size-8 p-0 has-[>svg]:p-0',
78       },
79     },
80     defaultVariants: {
81       size: 'xs',
82     },
83   },
84 );
85
86 function InputGroupButton({
87   className,
88   type = 'button',
89   variant = 'ghost',
90   size = 'xs',
91   ...props
92 }: Omit<React.ComponentProps<typeof Button>, 'size' | 'type'> &
93   VariantProps<typeof inputGroupButtonVariants> & {
94     type?: 'button' | 'submit' | 'reset';
95   }) {
96   return (
97     <Button
98       type={type}
99       data-size={size}
100       variant={variant}
101       className={cn(inputGroupButtonVariants({ size }), className)}
102       {...props}
103     />
104   );
105 }
106
107 function InputGroupText({ className, ...props }: React.ComponentProps<'span'>) {
108   return (
109     <span
110       className={cn(
111         "text-muted-foreground gap-2 text-sm [&_svg:not([class*='size-'])]:size-4 flex items-center
112         [&_svg]:pointer-events-none",
113         className,
114       )}
115       {...props}
116     />
117   );
118 }
119
120 function InputGroupInput({
121   className,
122   ...props
123 }: React.ComponentProps<'input'>) {
124   return (
125     <Input
126       data-slot="input-group-control"
127       className={cn(
128         'rounded-none border-0 bg-transparent shadow-none ring-0 focus-visible:ring-0 disabled:bg-
129         transparent aria-invalid:ring-0 dark:bg-transparent dark:disabled:bg-transparent flex-1',
130         className,
131       )}
132       {...props}
133     />
134   );
135 }
136
137 function InputGroupTextarea({
138   className,
139   ...props
140 }: React.ComponentProps<'textarea'>) {
141   return (
142     <Textarea
143       data-slot="input-group-control"
144       className={cn(
145         'rounded-none border-0 bg-transparent py-2 shadow-none ring-0 focus-visible:ring-0 disabled:bg-
146         transparent aria-invalid:ring-0 dark:bg-transparent dark:disabled:bg-transparent flex-1 resize-none',
147         className,
148       )}
149       {...props}
150     />
151   );
152 }

```

```
150
151 export {
152   InputGroup,
153   InputGroupAddon,
154   InputGroupButton,
155   InputGroupText,
156   InputGroupInput,
157   InputGroupTextarea,
158 };
```

components/ui/input-otp.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { OTPInput, OTPInputContext } from 'input-otp';
5
6 import { cn } from '@lib/utils';
7 import { MinusIcon } from 'lucide-react';
8
9 function InputOTP({
10   className,
11   containerClassName,
12   ...props
13 }: React.ComponentProps<typeof OTPInput> & {
14   containerClassName?: string;
15 }) {
16   return (
17     <OTPInput
18       data-slot="input-otp"
19       containerClassName={cn(
20         'cn-input-otp flex items-center has-disabled:opacity-50',
21         containerClassName,
22       )}
23       spellCheck={false}
24       className={cn('disabled:cursor-not-allowed', className)}
25       {...props}
26     />
27   );
28 }
29
30 function InputOTPGroup({ className, ...props }: React.ComponentProps<'div'>) {
31   return (
32     <div
33       data-slot="input-otp-group"
34       className={cn(
35         'has-aria-invalid:ring-destructive/20 dark:has-aria-invalid:ring-destructive/40 has-aria-
36         invalid:border-destructive rounded-lg has-aria-invalid:ring-3 flex items-center',
37         className,
38       )}
39       {...props}
40     />
41   );
42 }
43
44 function InputOTPSlot({
45   index,
46   className,
47   ...props
48 }: React.ComponentProps<'div'> & {
49   index: number;
50 }) {
51   const inputOTPContext = React.useContext(OTPInputContext);
52   const { char, hasFakeCaret, isActive } = inputOTPContext?.slots[index] ?? {};
53
54   return (
55     <div
56       data-slot="input-otp-slot"
57       data-active={isActive}
58       className={cn(
59         'dark:bg-input/30 border-input data-[active=true]:border-ring data-[active=true]:ring-ring/50
60         data-[active=true]:aria-invalid:ring-destructive/20 dark:data-[active=true]:aria-invalid:ring-
61         destructive/40 aria-invalid:border-destructive data-[active=true]:aria-invalid:border-destructive size-8
62         border-y border-r text-sm transition-all outline-none first:rounded-l-lg first:border-l last:rounded-r-
63         lg data-[active=true]:ring-3 relative flex items-center justify-center data-[active=true]:z-10',
64         className,
65       )}
66       {...props}
67     >
68       {char}
69       {hasFakeCaret && (
70         <div className="pointer-events-none absolute inset-0 flex items-center justify-center">
71           <div className="animate-caret-blink bg-foreground h-4 w-px duration-1000" />
72         </div>
73       )}
74     </div>
75   );
76 }
77
78 function InputOTPSeparator({ ...props }: React.ComponentProps<'div'>) {
79   return (
80     <div
81       data-slot="input-otp-separator"
82       className="[&_svg:not([class*='size-']):size-4 flex items-center"
83     />
84   );
85 }
```

```
78     role="separator"
79     {...props}
80   >
81     <MinusIcon />
82   </div>
83 );
84 }
85
86 export { InputOTP, InputOTPGroup, InputOTPSlot, InputOTPSeparator };
```

components/ui/input.tsx

```
1 import * as React from 'react';
2 import { Input as InputPrimitive } from '@base-ui/react/input';
3
4 import { cn } from '@lib/utils';
5
6 function Input({ className, type, ...props }: React.ComponentProps<'input'>) {
7   return (
8     <InputPrimitive
9       type={type}
10       data-slot="input"
11       className={cn(
12         'h-8 w-full min-w-0 rounded-lg border border-input bg-transparent px-2.5 py-1 text-base
13         transition-colors outline-none file:inline-flex file:h-6 file:border-0 file:bg-transparent file:text-sm
14         file:font-medium file:text-foreground placeholder:text-muted-foreground focus-visible:border-ring focus-
15         visible:ring-3 focus-visible:ring-ring/50 disabled:pointer-events-none disabled:cursor-not-allowed
16         disabled:bg-input/50 disabled:opacity-50 aria-invalid:border-destructive aria-invalid:ring-3 aria-
17         invalid:ring-destructive/20 md:text-sm dark:bg-input/30 dark:disabled:bg-input/80 dark:aria-
18         invalid:border-destructive/50 dark:aria-invalid:ring-destructive/40',
19         className,
20       )}
21     >
22       {props.children}
23     </InputPrimitive>
24   );
25 }
26
27 export { Input };
```

components/ui/item.tsx

```
1 import * as React from 'react';
2 import { mergeProps } from '@base-ui/react/merge-props';
3 import { useRender } from '@base-ui/react/use-render';
4 import { cva, type VariantProps } from 'class-variance-authority';
5
6 import { cn } from '@lib/utils';
7 import { Separator } from '@components/ui/separator';
8
9 function ItemGroup({ className, ...props }: React.ComponentProps<'div'>) {
10   return (
11     <div
12       role="list"
13       data-slot="item-group"
14       className={cn(
15         'gap-4 has-data-[size=sm]:gap-2.5 has-data-[size=xs]:gap-2 group/item-group flex w-full flex-
16         col',
17         className,
18       )}
19       {...props}
20     />
21   );
22 }
23
24 function ItemSeparator({
25   className,
26   ...props
27 }: React.ComponentProps<typeof Separator>) {
28   return (
29     <Separator
30       data-slot="item-separator"
31       orientation="horizontal"
32       className={cn('my-2', className)}
33       {...props}
34     />
35   );
36 }
37
38 const itemVariants = cva(
39   '[a]:hover:bg-muted rounded-lg border text-sm group/item flex w-full flex-wrap items-center
40   transition-colors duration-100 outline-none focus-visible:border-ring focus-visible:ring-[3px] focus-
41   visible:ring-ring/50 [a]:transition-colors',
42   {
43     variants: {
44       variant: {
45         default: 'border-transparent',
46         outline: 'border-border',
47         muted: 'bg-muted/50 border-transparent',
48       },
49       size: {
50         default: 'gap-2.5 px-3 py-2.5',
51         sm: 'gap-2.5 px-3 py-2.5',
52         xs: 'gap-2 px-2.5 py-2 in-data-[slot=dropdown-menu-content]:p-0',
53       },
54     },
55     defaultVariants: {
56       variant: 'default',
57       size: 'default',
58     },
59   },
60 );
61
62 function Item({
63   className,
64   variant = 'default',
65   size = 'default',
66   render,
67   ...props
68 }: useRender.ComponentProps<'div'> & VariantProps<typeof itemVariants>) {
69   return useRender({
70     defaultTagName: 'div',
71     props: mergeProps<'div'>({
72       className: cn(itemVariants({ variant, size, className })),
73     }, props),
74     render,
75     state: {
76       slot: 'item',
77       variant,
78       size,
79     },
80   });
81 }
```

```

80 });
81 }
82
83 const itemMediaVariants = cva(
84   'gap-2 group-has-data-[slot=item-description]/item:translate-y-0.5 group-has-data-[slot=item-
      description]/item:self-start flex shrink-0 items-center justify-center [&_svg]:pointer-events-none',
85   {
86     variants: {
87       variant: {
88         default: 'bg-transparent',
89         icon: "[&_svg:not([class*='size-']):size-4",
90         image:
91           'size-10 overflow-hidden rounded-sm group-data-[size=sm]/item:size-8 group-
      data-[size=xs]/item:size-6 [&_img]:size-full [&_img]:object-cover',
92       },
93     },
94     defaultVariants: {
95       variant: 'default',
96     },
97   },
98 );
99
100 function ItemMedia({
101   className,
102   variant = 'default',
103   ...props
104 }: React.ComponentProps<'div'> & VariantProps<typeof itemMediaVariants>) {
105   return (
106     <div
107       data-slot="item-media"
108       data-variant={variant}
109       className={cn(itemMediaVariants({ variant, className }))}
110       {...props}
111     />
112   );
113 }
114
115 function ItemContent({ className, ...props }: React.ComponentProps<'div'>) {
116   return (
117     <div
118       data-slot="item-content"
119       className={cn(
120         'gap-1 group-data-[size=xs]/item:gap-0 flex flex-1 flex-col [&+[data-slot=item-content]]:flex-
      none',
121       className,
122     )}
123     {...props}
124   />
125 );
126 }
127
128 function ItemTitle({ className, ...props }: React.ComponentProps<'div'>) {
129   return (
130     <div
131       data-slot="item-title"
132       className={cn(
133         'gap-2 text-sm leading-snug font-medium underline-offset-4 line-clamp-1 flex w-fit items-
      center',
134       className,
135     )}
136     {...props}
137   />
138 );
139 }
140
141 function ItemDescription({ className, ...props }: React.ComponentProps<'p'>) {
142   return (
143     <p
144       data-slot="item-description"
145       className={cn(
146         'text-muted-foreground text-left text-sm leading-normal group-data-[size=xs]/item:text-xs line-
      clamp-2 font-normal [&>a]:underline [&>a]:underline-offset-4 [&>a:hover]:text-primary',
147       className,
148     )}
149     {...props}
150   />
151 );
152 }
153
154 function ItemActions({ className, ...props }: React.ComponentProps<'div'>) {
155   return (
156     <div
157       data-slot="item-actions"
158       className={cn('gap-2 flex items-center', className)}
159       {...props}

```

```

160     />
161   );
162 }
163
164 function ItemHeader({ className, ...props }: React.ComponentProps<'div'>) {
165   return (
166     <div
167       data-slot="item-header"
168       className={cn(
169         'gap-2 flex basis-full items-center justify-between',
170         className,
171       )}
172       {...props}
173     />
174   );
175 }
176
177 function ItemFooter({ className, ...props }: React.ComponentProps<'div'>) {
178   return (
179     <div
180       data-slot="item-footer"
181       className={cn(
182         'gap-2 flex basis-full items-center justify-between',
183         className,
184       )}
185       {...props}
186     />
187   );
188 }
189
190 export {
191   Item,
192   ItemMedia,
193   ItemContent,
194   ItemActions,
195   ItemGroup,
196   ItemSeparator,
197   ItemTitle,
198   ItemDescription,
199   ItemHeader,
200   ItemFooter,
201 };

```

components/ui/kbd.tsx

```
1 import { cn } from '@lib/utils';
2
3 function Kbd({ className, ...props }: React.ComponentProps<'kbd'>) {
4   return (
5     <kbd
6       data-slot="kbd"
7       className={cn(
8         "bg-muted text-muted-foreground in-data-[slot=tooltip-content]:bg-background/20 in-
9         data-[slot=tooltip-content]:text-background dark:in-data-[slot=tooltip-content]:bg-background/10 h-5
10        w-fit min-w-5 gap-1 rounded-sm px-1 font-sans text-xs font-medium [&svg:not([class*='size-']):size-3
11        pointer-events-none inline-flex items-center justify-center select-none",
12        className,
13      )}
14      {...props}
15    />
16  );
17 }
18
19 function KbdGroup({ className, ...props }: React.ComponentProps<'div'>) {
20   return (
21     <kbd
22       data-slot="kbd-group"
23       className={cn('gap-1 inline-flex items-center', className)}
24       {...props}
25     />
26   );
27 }
28
29 export { Kbd, KbdGroup };
```

components/ui/label.tsx

```
1 'use client';
2
3 import * as React from 'react';
4
5 import { cn } from '@lib/utils';
6
7 function Label({ className, ...props }: React.ComponentProps<'label'>) {
8   return (
9     <label
10       data-slot="label"
11       className={cn(
12         'flex items-center gap-2 text-sm leading-none font-medium select-none group-
13         data-[disabled=true]:pointer-events-none group-data-[disabled=true]:opacity-50 peer-disabled:cursor-not-
14         allowed peer-disabled:opacity-50',
15         className,
16       )}
17     >
18   );
19 }
20 export { Label };
```

components/ui/marker.tsx

```
1 import * as React from 'react';
2 import { mergeProps } from '@base-ui/react/merge-props';
3 import { useRender } from '@base-ui/react/use-render';
4 import { cva, type VariantProps } from 'class-variance-authority';
5
6 import { cn } from '@lib/utils';
7
8 const markerVariants = cva(
9   "gap-2 text-sm text-muted-foreground [a]:hover:text-foreground [a]:underline-offset-3 [a]:underline
10  [&_svg:not([class*='size-']):size-4 min-h-4 text-left group/marker relative flex w-full items-center",
11  {
12    variants: {
13      variant: {
14        default: '',
15        separator:
16          'before:h-py before:min-w-0 before:flex-1 before:bg-border after:h-py after:min-w-0
17          after:flex-1 after:bg-border before:mr-1 after:ml-1',
18        border: 'border-b border-border pb-2',
19      },
20    },
21  });
22
23 function Marker({
24   className,
25   variant = 'default',
26   render,
27   ...props
28 }: useRender.ComponentProps<'div'> & VariantProps<typeof markerVariants>) {
29   return useRender({
30     defaultTagName: 'div',
31     props: mergeProps<'div'>({
32       className: cn(markerVariants({ variant, className })),
33     }, props),
34     render,
35     state: {
36       slot: 'marker',
37       variant,
38     },
39   });
40 }
41
42 function MarkerIcon({ className, ...props }: React.ComponentProps<'span'>) {
43   return (
44     <span
45       data-slot="marker-icon"
46       aria-hidden="true"
47       className={cn(
48         "size-4 [&_svg:not([class*='size-']):size-4 shrink-0",
49         className,
50       )}
51       {...props}
52     />
53   );
54 }
55
56 function MarkerContent({ className, ...props }: React.ComponentProps<'span'>) {
57   return (
58     <span
59       data-slot="marker-content"
60       className={cn(
61         "group-data-[variant=separator]/marker:flex-none group-data-[variant=separator]/marker:text-
62         center *:[:a]:hover:text-foreground *:[:a]:underline *:[:a]:underline-offset-3 min-w-0 wrap-break-word",
63         className,
64       )}
65       {...props}
66     />
67   );
68 }
69
70 export { Marker, MarkerIcon, MarkerContent, markerVariants };
```

components/ui/menubar.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { Menu as MenuPrimitive } from '@base-ui/react/menu';
5 import { Menubar as MenubarPrimitive } from '@base-ui/react/menubar';
6
7 import { cn } from '@lib/utils';
8 import {
9   DropdownMenu,
10   DropdownMenuContent,
11   DropdownMenuGroup,
12   DropdownMenuItem,
13   DropdownMenuLabel,
14   DropdownMenuPortal,
15   DropdownMenuRadioGroup,
16   DropdownMenuSeparator,
17   DropdownMenuShortcut,
18   DropdownMenuSub,
19   DropdownMenuSubContent,
20   DropdownMenuSubTrigger,
21   DropdownMenuTrigger,
22 } from '@components/ui/dropdown-menu';
23 import { CheckIcon } from 'lucide-react';
24
25 function Menubar({ className, ...props }: MenubarPrimitive.Props) {
26   return (
27     <MenubarPrimitive
28       data-slot="menubar"
29       className={cn(
30         'h-8 gap-0.5 rounded-lg border p-[3px] flex items-center',
31         className,
32       )}
33       {...props}
34     />
35   );
36 }
37
38 function MenubarMenu({ ...props }: React.ComponentProps<typeof DropdownMenu>) {
39   return <DropdownMenu data-slot="menubar-menu" {...props} />;
40 }
41
42 function MenubarGroup({
43   ...props
44 }: React.ComponentProps<typeof DropdownMenuGroup>) {
45   return <DropdownMenuGroup data-slot="menubar-group" {...props} />;
46 }
47
48 function MenubarPortal({
49   ...props
50 }: React.ComponentProps<typeof DropdownMenuPortal>) {
51   return <DropdownMenuPortal data-slot="menubar-portal" {...props} />;
52 }
53
54 function MenubarTrigger({
55   className,
56   ...props
57 }: React.ComponentProps<typeof DropdownMenuTrigger>) {
58   return (
59     <DropdownMenuTrigger
60       data-slot="menubar-trigger"
61       className={cn(
62         'hover:bg-muted aria-expanded:bg-muted rounded-sm px-1.5 py-[2px] text-sm font-medium flex
63         items-center outline-hidden select-none',
64         className,
65       )}
66       {...props}
67     />
68   );
69 }
70
71 function MenubarContent({
72   className,
73   align = 'start',
74   alignOffset = -4,
75   sideOffset = 8,
76   ...props
77 }: React.ComponentProps<typeof DropdownMenuContent>) {
78   return (
79     <DropdownMenuContent
80       data-slot="menubar-content"
81       align={align}
82       alignOffset={alignOffset}
83     />
84   );
85 }
```

```

82     sideOffset={sideOffset}
83     className={cn(
84       'bg-popover text-popover-foreground data-open:animate-in data-open:fade-in-0 data-open:zoom-
in-95 data-[side=bottom]:slide-in-from-top-2 data-[side=left]:slide-in-from-right-2
data-[side=right]:slide-in-from-left-2 data-[side=top]:slide-in-from-bottom-2 ring-foreground/10
min-w-36 rounded-lg p-1 shadow-md ring-1 duration-100 data-[side=inline-start]:slide-in-from-right-2
data-[side=inline-end]:slide-in-from-left-2 cn-menu-target cn-menu-translucent',
85       className,
86     )}
87     {...props}
88   />
89 );
90 }
91
92 function MenubarItem({
93   className,
94   inset,
95   variant = 'default',
96   ...props
97 }: React.ComponentProps<typeof DropdownMenuItem>) {
98   return (
99     <DropdownMenuItem
100       data-slot="menubar-item"
101       data-inset={inset}
102       data-variant={variant}
103       className={cn(
104         "focus:bg-accent focus:text-accent-foreground data-[variant=destructive]:text-destructive
data-[variant=destructive]:focus:bg-destructive/10 dark:data-[variant=destructive]:focus:bg-
destructive/20 data-[variant=destructive]:focus:text-destructive
data-[variant=destructive]:*:[svg]:text-destructive! not-data-[variant=destructive]:focus:*:text-
accent-foreground gap-1.5 rounded-md px-1.5 py-1 text-sm data-disabled:opacity-50 data-inset:pl-7
[&_svg]:not([class*='size-'])]:size-4 group/menubar-item",
105         className,
106       )}
107       {...props}
108     />
109   );
110 }
111
112 function MenubarCheckboxItem({
113   className,
114   children,
115   checked,
116   inset,
117   ...props
118 }: MenuPrimitive.CheckboxItem.Props & {
119   inset?: boolean;
120 }) {
121   return (
122     <MenuPrimitive.CheckboxItem
123       data-slot="menubar-checkbox-item"
124       data-inset={inset}
125       className={cn(
126         'focus:bg-accent focus:text-accent-foreground focus:*:text-accent-foreground gap-1.5 rounded-md
py-1 pr-1.5 pl-7 text-sm data-inset:pl-7 relative flex cursor-default items-center outline-hidden
select-none data-disabled:pointer-events-none data-disabled:opacity-50 [&_svg]:pointer-events-none
[&_svg]:shrink-0',
127         className,
128       )}
129       checked={checked}
130       {...props}
131     >
132       <span className="left-1.5 size-4 [&_svg]:not([class*='size-'])]:size-4 pointer-events-none absolute
flex items-center justify-center">
133         <MenuPrimitive.CheckboxItemIndicator>
134           <CheckIcon />
135         </MenuPrimitive.CheckboxItemIndicator>
136       </span>
137       {children}
138     </MenuPrimitive.CheckboxItem>
139   );
140 }
141
142 function MenubarRadioGroup({
143   ...props
144 }: React.ComponentProps<typeof DropdownMenuRadioGroup>) {
145   return <DropdownMenuRadioGroup data-slot="menubar-radio-group" {...props} />;
146 }
147
148 function MenubarRadioItem({
149   className,
150   children,
151   inset,
152   ...props
153 }: MenuPrimitive.RadioItem.Props & {

```

```

154   inset?: boolean;
155 }) {
156   return (
157     <MenuPrimitive.RadioItem
158       data-slot="menubar-radio-item"
159       data-inset={inset}
160       className={cn(
161         "focus:bg-accent focus:text-accent-foreground focus:**:text-accent-foreground gap-1.5 rounded-md
py-1 pr-1.5 pl-7 text-sm data-disabled:opacity-50 data-inset:pl-7 [&_svg:not([class*='size-'])]:size-4
relative flex cursor-default items-center outline-hidden select-none data-disabled:pointer-events-none
[&_svg]:pointer-events-none [&_svg]:shrink-0",
162         className,
163       )}
164       {...props}
165     >
166       <span className="left-1.5 size-4 [&_svg:not([class*='size-'])]:size-4 pointer-events-none absolute
flex items-center justify-center">
167         <MenuPrimitive.RadioItemIndicator>
168           <CheckIcon />
169         </MenuPrimitive.RadioItemIndicator>
170       </span>
171       {children}
172     </MenuPrimitive.RadioItem>
173   );
174 }
175
176 function MenubarLabel({
177   className,
178   inset,
179   ...props
180 }: React.ComponentProps<typeof DropdownMenuLabel> & {
181   inset?: boolean;
182 }) {
183   return (
184     <DropdownMenuLabel
185       data-slot="menubar-label"
186       data-inset={inset}
187       className={cn(
188         'px-1.5 py-1 text-sm font-medium data-inset:pl-7',
189         className,
190       )}
191       {...props}
192     />
193   );
194 }
195
196 function MenubarSeparator({
197   className,
198   ...props
199 }: React.ComponentProps<typeof DropdownMenuSeparator>) {
200   return (
201     <DropdownMenuSeparator
202       data-slot="menubar-separator"
203       className={cn('bg-border -mx-1 my-1 h-px', className)}
204       {...props}
205     />
206   );
207 }
208
209 function MenubarShortcut({
210   className,
211   ...props
212 }: React.ComponentProps<typeof DropdownMenuShortcut>) {
213   return (
214     <DropdownMenuShortcut
215       data-slot="menubar-shortcut"
216       className={cn(
217         'text-muted-foreground group-focus/menubar-item:text-accent-foreground text-xs tracking-widest
ml-auto',
218         className,
219       )}
220       {...props}
221     />
222   );
223 }
224
225 function MenubarSub({
226   ...props
227 }: React.ComponentProps<typeof DropdownMenuSub>) {
228   return <DropdownMenuSub data-slot="menubar-sub" {...props} />;
229 }
230
231 function MenubarSubTrigger({
232   className,
233   inset,

```

```

234   ...props
235 }; React.ComponentProps<typeof DropdownMenuSubTrigger> & {
236   inset?: boolean;
237 }) {
238   return (
239     <DropdownMenuSubTrigger
240       data-slot="menubar-sub-trigger"
241       data-inset={inset}
242       className={cn(
243         "focus:bg-accent focus:text-accent-foreground data-open:bg-accent data-open:text-accent-
foreground gap-1.5 rounded-md px-1.5 py-1 text-sm data-inset:pl-7 [&svg:not([class*='size-'])]:size-4",
244         className,
245       )}
246       {...props}
247     />
248   );
249 }
250
251 function MenubarSubContent({
252   className,
253   ...props
254 }: React.ComponentProps<typeof DropdownMenuSubContent>) {
255   return (
256     <DropdownMenuSubContent
257       data-slot="menubar-sub-content"
258       className={cn(
259         'bg-popover text-popover-foreground data-open:animate-in data-closed:animate-out data-
closed:fade-out-0 data-open:fade-in-0 data-closed:zoom-out-95 data-open:zoom-in-95
data-[side=bottom]:slide-in-from-top-2 data-[side=left]:slide-in-from-right-2 data-[side=right]:slide-
in-from-left-2 data-[side=top]:slide-in-from-bottom-2 ring-foreground/10 min-w-32 rounded-lg p-1 shadow-
lg ring-1 duration-100 cn-menu-target cn-menu-translucent',
260         className,
261       )}
262       {...props}
263     />
264   );
265 }
266
267 export {
268   Menubar,
269   MenubarPortal,
270   MenubarMenu,
271   MenubarTrigger,
272   MenubarContent,
273   MenubarGroup,
274   MenubarSeparator,
275   MenubarLabel,
276   MenubarItem,
277   MenubarShortcut,
278   MenubarCheckboxItem,
279   MenubarRadioGroup,
280   MenubarRadioItem,
281   MenubarSub,
282   MenubarSubTrigger,
283   MenubarSubContent,
284 };

```

components/ui/message-scroller.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import {
5   MessageScroller as MessageScrollerPrimitive,
6   useMessageScroller,
7   useMessageScrollerScrollable,
8   useMessageScrollerVisibility,
9 } from '@shadcn/react/message-scroller';
10
11 import { cn } from '@lib/utils';
12 import { Button } from '@components/ui/button';
13 import { ArrowDownIcon } from 'lucide-react';
14
15 function MessageScrollerProvider(
16   props: React.ComponentProps<typeof MessageScrollerPrimitive.Provider>,
17 ) {
18   return <MessageScrollerPrimitive.Provider {...props} />;
19 }
20
21 function MessageScroller({
22   className,
23   ...props
24 }: React.ComponentProps<typeof MessageScrollerPrimitive.Root>) {
25   return (
26     <MessageScrollerPrimitive.Root
27       data-slot="message-scroller"
28       className={cn(
29         'group/message-scroller relative flex size-full min-h-0 flex-col overflow-hidden',
30         className,
31       )}
32       {...props}
33     />
34   );
35 }
36
37 function MessageScrollerViewport({
38   className,
39   ...props
40 }: React.ComponentProps<typeof MessageScrollerPrimitive.Viewport>) {
41   return (
42     <MessageScrollerPrimitive.Viewport
43       data-slot="message-scroller-viewport"
44       className={cn(
45         'size-full min-h-0 min-w-0 scroll-fade-b scrollbar-thin scrollbar-gutter-stable overflow-y-auto
46         overscroll-contain contain-content data-autoscrolling:scrollbar-thumb-transparent data-
47         autoscrolling:scrollbar-track-transparent',
48         className,
49       )}
50       {...props}
51     />
52   );
53 }
54
55 function MessageScrollerContent({
56   className,
57   ...props
58 }: React.ComponentProps<typeof MessageScrollerPrimitive.Content>) {
59   return (
60     <MessageScrollerPrimitive.Content
61       data-slot="message-scroller-content"
62       className={cn('gap-6 flex h-max min-h-full flex-col', className)}
63       {...props}
64     />
65   );
66 }
67
68 function MessageScrollerItem({
69   className,
70   scrollAnchor = false,
71   ...props
72 }: React.ComponentProps<typeof MessageScrollerPrimitive.Item>) {
73   return (
74     <MessageScrollerPrimitive.Item
75       data-slot="message-scroller-item"
76       scrollAnchor={scrollAnchor}
77       className={cn(
78         'min-w-0 shrink-0 [contain-intrinsic-size:auto_10rem] [content-visibility:auto]',
79         className,
80       )}
81       {...props}
82     />
83   );
84 }
```

```

81   });
82 }
83
84 function MessageScrollerButton({
85   direction = 'end',
86   className,
87   children,
88   render,
89   variant = 'secondary',
90   size = 'icon-sm',
91   ...props
92 }: React.ComponentProps<typeof MessageScrollerPrimitive.Button> &
93   Pick<React.ComponentProps<typeof Button>, 'variant' | 'size'>) {
94   return (
95     <MessageScrollerPrimitive.Button
96       data-slot="message-scroller-button"
97       data-direction={direction}
98       data-variant={variant}
99       data-size={size}
100     direction={direction}
101     className={cn(
102       'absolute inset-s-1/2 -translate-x-1/2 border-border bg-background text-foreground
transition-[translate,scale,opacity] duration-200 hover:bg-muted hover:text-foreground
data-[active=false]:pointer-events-none data-[active=false]:scale-95 data-[active=false]:opacity-0
data-[active=false]:duration-400 data-[active=false]:ease-[cubic-bezier(0.7,0,0.84,0)]
data-[active=true]:translate-y-0 data-[active=true]:scale-100 data-[active=true]:opacity-100
data-[active=true]:ease-[cubic-bezier(0.23,1,0.32,1)] data-[direction=end]:bottom-4
data-[direction=end]:data-[active=false]:translate-y-full data-[direction=start]:top-4
data-[direction=start]:data-[active=false]:-translate-y-full rtl:translate-x-1/2
data-[direction=start]:[&_svg]:rotate-180',
103       className,
104     )}
105     render={render ?? <Button variant={variant} size={size} />}
106     {...props}
107   >
108     {children ?? (
109       <>
110         <ArrowDownIcon />
111         <span className="sr-only">
112           {direction === 'end' ? 'Scroll to end' : 'Scroll to start'}
113         </span>
114       </>
115     )}
116   </MessageScrollerPrimitive.Button>
117 );
118 }
119
120 export {
121   MessageScrollerProvider,
122   MessageScroller,
123   MessageScrollerViewport,
124   MessageScrollerContent,
125   MessageScrollerItem,
126   MessageScrollerButton,
127   useMessageScroller,
128   useMessageScrollerScrollable,
129   useMessageScrollerVisibility,
130 };

```

components/ui/message.tsx

```
1 import * as React from 'react';
2
3 import { cn } from '@lib/utils';
4
5 function MessageGroup({ className, ...props }: React.ComponentProps<'div'>) {
6   return (
7     <div
8       data-slot="message-group"
9       className={cn('gap-2 flex min-w-0 flex-col', className)}
10      {...props}
11    />
12  );
13 }
14
15 function Message({
16   className,
17   align = 'start',
18   ...props
19 }: React.ComponentProps<'div'> & { align?: 'start' | 'end' }) {
20   return (
21     <div
22       data-slot="message"
23       data-align={align}
24       className={cn(
25         'text-sm gap-2 group/message relative flex w-full min-w-0 data-[align=end]:flex-row-reverse',
26         className,
27       )}
28       {...props}
29     />
30   );
31 }
32
33 function MessageAvatar({ className, ...props }: React.ComponentProps<'div'>) {
34   return (
35     <div
36       data-slot="message-avatar"
37       className={cn(
38         'min-w-8 group-has-data-[slot=message-footer]/message:-translate-y-8 flex w-fit shrink-0 items-
39         center justify-center self-end overflow-hidden rounded-full bg-muted',
40         className,
41       )}
42       {...props}
43     />
44   );
45 }
46
47 function MessageContent({ className, ...props }: React.ComponentProps<'div'>) {
48   return (
49     <div
50       data-slot="message-content"
51       className={cn(
52         'gap-2.5 group-data-[align=end]/message:*.data-slot:self-end flex w-full min-w-0 flex-col wrap-
53         break-word',
54         className,
55       )}
56       {...props}
57     />
58   );
59 }
60
61 function MessageHeader({ className, ...props }: React.ComponentProps<'div'>) {
62   return (
63     <div
64       data-slot="message-header"
65       className={cn(
66         'text-xs font-medium text-muted-foreground px-3 group-has-data-[variant=ghost]/message:px-0 flex
67         max-w-full min-w-0 items-center',
68         className,
69       )}
70       {...props}
71     />
72   );
73 }
74
75 function MessageFooter({ className, ...props }: React.ComponentProps<'div'>) {
76   return (
77     <div
78       data-slot="message-footer"
79       className={cn(
80         'text-xs font-medium text-muted-foreground px-3 group-has-data-[variant=ghost]/message:px-0 flex
81         max-w-full min-w-0 items-center group-data-[align=end]/message:justify-end',
82         className,
83       )}
84       {...props}
85     />
86   );
87 }
```

```
79     })}
80     {...props}
81   />
82   );
83 }
84
85 export {
86   MessageGroup,
87   Message,
88   MessageAvatar,
89   MessageContent,
90   MessageFooter,
91   MessageHeader,
92 };
```

components/ui/native-select.tsx

```
1 import * as React from 'react';
2
3 import { cn } from '@lib/utils';
4 import { ChevronDownIcon } from 'lucide-react';
5
6 type NativeSelectProps = Omit<React.ComponentProps<'select'>, 'size'> & {
7   size?: 'sm' | 'default';
8 };
9
10 function NativeSelect({
11   className,
12   size = 'default',
13   ...props
14 }: NativeSelectProps) {
15   return (
16     <div
17       className={cn(
18         'group/native-select relative w-fit has-[select:disabled]:opacity-50',
19         className,
20       )}
21       data-slot="native-select-wrapper"
22       data-size={size}
23     >
24       <select
25         data-slot="native-select"
26         data-size={size}
27         className="border-input placeholder:text-muted-foreground selection:bg-primary selection:text-
primary-foreground dark:bg-input/30 dark:hover:bg-input/50 focus-visible:border-ring focus-visible:ring-
ring/50 aria-invalid:ring-destructive/20 dark:aria-invalid:ring-destructive/40 aria-invalid:border-
destructive dark:aria-invalid:border-destructive/50 h-8 w-full min-w-0 appearance-none rounded-lg border
bg-transparent py-1 pr-8 pl-2.5 text-sm transition-colors select-none focus-visible:ring-3 aria-
invalid:ring-3 data-[size=sm]:h-7 data-[size=sm]:rounded-[min(var(--radius-md),10px)]
data-[size=sm]:py-0.5 outline-none disabled:pointer-events-none disabled:cursor-not-allowed"
28         {...props}
29       />
30       <ChevronDownIcon
31         className="text-muted-foreground top-1/2 right-2.5 size-4 -translate-y-1/2 pointer-events-none
absolute select-none"
32         aria-hidden="true"
33         data-slot="native-select-icon"
34       />
35     </div>
36   );
37 }
38
39 function NativeSelectOption({
40   className,
41   ...props
42 }: React.ComponentProps<'option'>) {
43   return (
44     <option
45       data-slot="native-select-option"
46       className={cn('bg-[Canvas] text-[CanvasText]', className)}
47       {...props}
48     />
49   );
50 }
51
52 function NativeSelectOptGroup({
53   className,
54   ...props
55 }: React.ComponentProps<'optgroup'>) {
56   return (
57     <optgroup
58       data-slot="native-select-optgroup"
59       className={cn('bg-[Canvas] text-[CanvasText]', className)}
60       {...props}
61     />
62   );
63 }
64
65 export { NativeSelect, NativeSelectOptGroup, NativeSelectOption };
```

components/ui/navigation-menu.tsx

```
1 import { NavigationMenu as NavigationMenuPrimitive } from '@base-ui/react/navigation-menu';
2 import { cva } from 'class-variance-authority';
3
4 import { cn } from '@lib/utils';
5 import { ChevronDownIcon } from 'lucide-react';
6
7 function NavigationMenu({
8   align = 'start',
9   className,
10  children,
11  ...props
12 }: NavigationMenuPrimitive.Root.Props &
13   Pick<NavigationMenuPrimitive.Positioner.Props, 'align'>) {
14   return (
15     <NavigationMenuPrimitive.Root
16       data-slot="navigation-menu"
17       className={cn(
18         'group/navigation-menu relative flex max-w-max flex-1 items-center justify-center',
19         className,
20       )}
21       {...props}
22     >
23       {children}
24       <NavigationMenuPositioner align={align} />
25     </NavigationMenuPrimitive.Root>
26   );
27 }
28
29 function NavigationMenuList({
30   className,
31   ...props
32 }: React.ComponentPropsWithRef<typeof NavigationMenuPrimitive.List>) {
33   return (
34     <NavigationMenuPrimitive.List
35       data-slot="navigation-menu-list"
36       className={cn(
37         'gap-0 group flex flex-1 list-none items-center justify-center',
38         className,
39       )}
40       {...props}
41     />
42   );
43 }
44
45 function NavigationMenuItem({
46   className,
47   ...props
48 }: React.ComponentPropsWithRef<typeof NavigationMenuPrimitive.Item>) {
49   return (
50     <NavigationMenuPrimitive.Item
51       data-slot="navigation-menu-item"
52       className={cn('relative', className)}
53       {...props}
54     />
55   );
56 }
57
58 const navigationMenuTriggerStyle = cva(
59   'hover:bg-muted focus:bg-muted data-open:hover:bg-muted data-open:focus:bg-muted data-open:bg-muted/50
60   focus-visible:ring-ring/50 data-popup-open:bg-muted/50 data-popup-open:hover:bg-muted rounded-lg px-2.5
61   py-1.5 text-sm font-medium transition-all focus-visible:ring-3 focus-visible:outline-1
62   disabled:opacity-50 group/navigation-menu-trigger inline-flex h-9 w-max items-center justify-center
63   outline-none disabled:pointer-events-none',
64 );
65
66 function NavigationMenuTrigger({
67   className,
68   children,
69   ...props
70 }: NavigationMenuPrimitive.Trigger.Props) {
71   return (
72     <NavigationMenuPrimitive.Trigger
73       data-slot="navigation-menu-trigger"
74       className={cn(navigationMenuTriggerStyle(), 'group', className)}
75       {...props}
76     >
77       {children}{' '}
78       <ChevronDownIcon
79         className="relative top-px ml-1 size-3 transition duration-300 group-data-open/navigation-menu-
80 trigger:rotate-180 group-data-popup-open/navigation-menu-trigger:rotate-180"
81         aria-hidden="true"
82       />
83     </NavigationMenuPrimitive.Trigger>
84   );
85 }
```

```

78     </NavigationMenuPrimitive.Trigger>
79   );
80 }
81
82 function NavigationMenuContent({
83   className,
84   ...props
85 }: NavigationMenuPrimitive.Content.Props) {
86   return (
87     <NavigationMenuPrimitive.Content
88       data-slot="navigation-menu-content"
89       className={cn(
90         'data-[motion^=from-]:animate-in data-[motion^=to-]:animate-out data-[motion^=from-]:fade-in
data-[motion^=to-]:fade-out data-[motion=from-end]:slide-in-from-right-52 data-[motion=from-
start]:slide-in-from-left-52 data-[motion=to-end]:slide-out-to-right-52 data-[motion=to-start]:slide-
out-to-left-52 group-data-[viewport=false]/navigation-menu:bg-popover group-
data-[viewport=false]/navigation-menu:text-popover-foreground group-data-[viewport=false]/navigation-
menu:data-open:animate-in group-data-[viewport=false]/navigation-menu:data-closed:animate-out group-
data-[viewport=false]/navigation-menu:data-closed:zoom-out-95 group-data-[viewport=false]/navigation-
menu:data-open:zoom-in-95 group-data-[viewport=false]/navigation-menu:data-open:fade-in-0 group-
data-[viewport=false]/navigation-menu:data-closed:fade-out-0 group-data-[viewport=false]/navigation-
menu:ring-foreground/10 p-1 ease-[cubic-bezier(0.22,1,0.36,1)] group-data-[viewport=false]/navigation-
menu:rounded-lg group-data-[viewport=false]/navigation-menu:shadow group-
data-[viewport=false]/navigation-menu:ring-1 group-data-[viewport=false]/navigation-menu:duration-300
data-ending-style:data-activation-direction=left:translate-x-[50%] data-ending-style:data-activation-
direction=right:translate-x-[-50%] data-starting-style:data-activation-direction=left:translate-x-[-50%]
data-starting-style:data-activation-direction=right:translate-x-[50%] h-full w-auto
transition-[opacity,transform,translate] duration-[0.35s] data-ending-style:opacity-0 data-starting-
style:opacity-0 **:data-[slot=navigation-menu-link]:focus:ring-0 **:data-[slot=navigation-menu-
link]:focus:outline-none',
91       className,
92     )}
93     {...props}
94   />
95 );
96 }
97
98 function NavigationMenuPositioner({
99   className,
100   side = 'bottom',
101   sideOffset = 8,
102   align = 'start',
103   alignOffset = 0,
104   ...props
105 }: NavigationMenuPrimitive.Positioner.Props) {
106   return (
107     <NavigationMenuPrimitive.Portal>
108       <NavigationMenuPrimitive.Positioner
109         side={side}
110         sideOffset={sideOffset}
111         align={align}
112         alignOffset={alignOffset}
113         className={cn(
114           'ease-[cubic-bezier(0.22,1,0.36,1)] data-[side=bottom]:before:top-[-10px]
data-[side=bottom]:before:right-0 data-[side=bottom]:before:left-0 isolate z-50 h-(--positioner-height)
w-(--positioner-width) max-w-(--available-width) transition-[top,left,right,bottom] duration-[0.35s]
data-instant:transition-none',
115         className,
116       )}
117       {...props}
118     >
119       <NavigationMenuPrimitive.Popup className="bg-popover text-popover-foreground ring-foreground/10
rounded-lg shadow ring-1 outline-none data-ending-style:scale-90 data-ending-style:opacity-0 data-
ending-style:duration-150 data-starting-style:scale-90 data-starting-style:opacity-0 data-[ending-
style]:easing-[ease] xs:w-(--popup-width) relative h-(--popup-height) w-(--popup-width)
origin-(--transform-origin) transition-[opacity,transform,width,height,scale,translate] duration-[0.35s]
ease-[cubic-bezier(0.22,1,0.36,1)]">
120         <NavigationMenuPrimitive.Viewport className="relative size-full overflow-hidden" />
121       </NavigationMenuPrimitive.Popup>
122     </NavigationMenuPrimitive.Positioner>
123   </NavigationMenuPrimitive.Portal>
124 );
125 }
126
127 function NavigationMenuLink({
128   className,
129   ...props
130 }: NavigationMenuPrimitive.Link.Props) {
131   return (
132     <NavigationMenuPrimitive.Link
133       data-slot="navigation-menu-link"
134       className={cn(
135         "data-active:focus:bg-muted data-active:hover:bg-muted data-active:bg-muted/50 focus-
visible:ring-ring/50 hover:bg-muted focus:bg-muted flex items-center gap-2 rounded-lg p-2 text-sm
transition-all outline-none focus-visible:ring-3 focus-visible:outline-1 in-data-[slot=navigation-menu-

```

```

        content]:rounded-md [&_svg:not([class*='size-']):size-4",
136         className,
137     )}
138     {...props}
139 />
140 );
141 }
142
143 function NavigationMenuIndicator({
144     className,
145     ...props
146 }: React.ComponentPropsWithRef<typeof NavigationMenuPrimitive.Icon>) {
147     return (
148         <NavigationMenuPrimitive.Icon
149             data-slot="navigation-menu-indicator"
150             className={cn(
151                 'data-[state=visible]:animate-in data-[state=hidden]:animate-out data-[state=hidden]:fade-out
data-[state=visible]:fade-in top-full z-1 flex h-1.5 items-end justify-center overflow-hidden',
152                 className,
153             )}
154             {...props}
155         >
156             <div className="bg-border rounded-tl-sm shadow-md relative top-[60%] h-2 w-2 rotate-45" />
157             </NavigationMenuPrimitive.Icon>
158         );
159     }
160
161 export {
162     NavigationMenu,
163     NavigationMenuContent,
164     NavigationMenuIndicator,
165     NavigationMenuItem,
166     NavigationMenuLink,
167     NavigationMenuList,
168     NavigationMenuTrigger,
169     navigationMenuTriggerStyle,
170     NavigationMenuPositioner,
171 };

```

components/ui/pagination.tsx

```
1 import * as React from 'react';
2
3 import { cn } from '@lib/utils';
4 import { Button } from '@components/ui/button';
5 import {
6   ChevronLeftIcon,
7   ChevronRightIcon,
8   MoreHorizontalIcon,
9 } from 'lucide-react';
10
11 function Pagination({ className, ...props }: React.ComponentProps<'nav'>) {
12   return (
13     <nav
14       role="navigation"
15       aria-label="pagination"
16       data-slot="pagination"
17       className={cn('mx-auto flex w-full justify-center', className)}
18       {...props}
19     />
20   );
21 }
22
23 function PaginationContent({
24   className,
25   ...props
26 }: React.ComponentProps<'ul'>) {
27   return (
28     <ul
29       data-slot="pagination-content"
30       className={cn('gap-0.5 flex items-center', className)}
31       {...props}
32     />
33   );
34 }
35
36 function PaginationItem({ ...props }: React.ComponentProps<'li'>) {
37   return <li data-slot="pagination-item" {...props} />;
38 }
39
40 type PaginationLinkProps = {
41   isActive?: boolean;
42 } & Pick<React.ComponentProps<typeof Button>, 'size'> &
43   React.ComponentProps<'a'>;
44
45 function PaginationLink({
46   className,
47   isActive,
48   size = 'icon',
49   ...props
50 }: PaginationLinkProps) {
51   return (
52     <Button
53       variant={isActive ? 'outline' : 'ghost'}
54       size={size}
55       className={cn(className)}
56       nativeButton={false}
57       render={
58         <a
59           aria-current={isActive ? 'page' : undefined}
60           data-slot="pagination-link"
61           data-active={isActive}
62           {...props}
63         />
64       }
65     />
66   );
67 }
68
69 function PaginationPrevious({
70   className,
71   text = 'Previous',
72   ...props
73 }: React.ComponentProps<typeof PaginationLink> & { text?: string }) {
74   return (
75     <PaginationLink
76       aria-label="Go to previous page"
77       size="default"
78       className={cn('pl-1.5!', className)}
79       {...props}
80     >
81     <ChevronLeftIcon data-icon="inline-start" className="cn-rtl-flip" />
82     <span className="hidden sm:block">{text}</span>

```

```

83     </PaginationLink>
84   );
85 }
86
87 function PaginationNext({
88   className,
89   text = 'Next',
90   ...props
91 }: React.ComponentProps<typeof PaginationLink> & { text?: string }) {
92   return (
93     <PaginationLink
94       aria-label="Go to next page"
95       size="default"
96       className={cn('pr-1.5!', className)}
97       {...props}
98     >
99       <span className="hidden sm:block">{text}</span>
100     <ChevronRightIcon data-icon="inline-end" className="cn-rtl-flip" />
101   </PaginationLink>
102 );
103 }
104
105 function PaginationEllipsis({
106   className,
107   ...props
108 }: React.ComponentProps<'span'>) {
109   return (
110     <span
111       aria-hidden
112       data-slot="pagination-ellipsis"
113       className={cn(
114         "size-8 [&_svg:not([class*='size-'])]:size-4 flex items-center justify-center",
115         className,
116       )}
117       {...props}
118     >
119       <MoreHorizontalIcon />
120       <span className="sr-only">More pages</span>
121     </span>
122 );
123 }
124
125 export {
126   Pagination,
127   PaginationContent,
128   PaginationEllipsis,
129   PaginationItem,
130   PaginationLink,
131   PaginationNext,
132   PaginationPrevious,
133 };

```

components/ui/popover.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { Popover as PopoverPrimitive } from '@base-ui/react/popover';
5
6 import { cn } from '@lib/utils';
7
8 function Popover({ ...props }: PopoverPrimitive.Root.Props) {
9   return <PopoverPrimitive.Root data-slot="popover" {...props} />;
10 }
11
12 function PopoverTrigger({ ...props }: PopoverPrimitive.Trigger.Props) {
13   return <PopoverPrimitive.Trigger data-slot="popover-trigger" {...props} />;
14 }
15
16 function PopoverContent({
17   className,
18   align = 'center',
19   alignOffset = 0,
20   side = 'bottom',
21   sideOffset = 4,
22   ...props
23 }: PopoverPrimitive.Popup.Props &
24   Pick<
25     PopoverPrimitive.Positioner.Props,
26     'align' | 'alignOffset' | 'side' | 'sideOffset'
27   > {
28   return (
29     <PopoverPrimitive.Portal>
30       <PopoverPrimitive.Positioner
31         align={align}
32         alignOffset={alignOffset}
33         side={side}
34         sideOffset={sideOffset}
35         className="isolate z-50"
36       >
37         <PopoverPrimitive.Popup
38           data-slot="popover-content"
39           className={cn(
40             'bg-popover text-popover-foreground data-open:animate-in data-closed:animate-out data-
41             closed:fade-out-0 data-open:fade-in-0 data-closed:zoom-out-95 data-open:zoom-in-95
42             data-[side=bottom]:slide-in-from-top-2 data-[side=left]:slide-in-from-right-2 data-[side=right]:slide-
43             in-from-left-2 data-[side=top]:slide-in-from-bottom-2 ring-foreground/10 flex flex-col gap-2.5 rounded-
44             lg p-2.5 text-sm shadow-md ring-1 duration-100 data-[side=inline-start]:slide-in-from-right-2
45             data-[side=inline-end]:slide-in-from-left-2 z-50 w-72 origin(--transform-origin) outline-hidden',
46             className,
47           )}
48           {...props}
49         />
50       </PopoverPrimitive.Positioner>
51     </PopoverPrimitive.Portal>
52   );
53 }
54
55 function PopoverHeader({ className, ...props }: React.ComponentProps<'div'>) {
56   return (
57     <div
58       data-slot="popover-header"
59       className={cn('flex flex-col gap-0.5 text-sm', className)}
60       {...props}
61     />
62   );
63 }
64
65 function PopoverTitle({ className, ...props }: PopoverPrimitive.Title.Props) {
66   return (
67     <PopoverPrimitive.Title
68       data-slot="popover-title"
69       className={cn('font-medium', className)}
70       {...props}
71     />
72   );
73 }
74
75 function PopoverDescription({
76   className,
77   ...props
78 }: PopoverPrimitive.Description.Props) {
79   return (
80     <PopoverPrimitive.Description
81       data-slot="popover-description"
82       className={cn('text-muted-foreground', className)}
83     />
84   );
85 }
```

```
78     {...props}
79   />
80 );
81 }
82
83 export {
84   Popover,
85   PopoverContent,
86   PopoverDescription,
87   PopoverHeader,
88   PopoverTitle,
89   PopoverTrigger,
90 };
```

components/ui/progress.tsx

```
1 'use client';
2
3 import { Progress as ProgressPrimitive } from '@base-ui/react/progress';
4
5 import { cn } from '@lib/utils';
6
7 function Progress({
8   className,
9   children,
10  value,
11  ...props
12 }: ProgressPrimitive.Root.Props) {
13   return (
14     <ProgressPrimitive.Root
15       value={value}
16       data-slot="progress"
17       className={cn('flex flex-wrap gap-3', className)}
18       {...props}
19     >
20       {children}
21       <ProgressTrack>
22         <ProgressIndicator />
23       </ProgressTrack>
24     </ProgressPrimitive.Root>
25   );
26 }
27
28 function ProgressTrack({ className, ...props }: ProgressPrimitive.Track.Props) {
29   return (
30     <ProgressPrimitive.Track
31       className={cn(
32         'bg-muted h-1 rounded-full relative flex w-full items-center overflow-x-hidden',
33         className,
34       )}
35       data-slot="progress-track"
36       {...props}
37     />
38   );
39 }
40
41 function ProgressIndicator({
42   className,
43   ...props
44 }: ProgressPrimitive.Indicator.Props) {
45   return (
46     <ProgressPrimitive.Indicator
47       data-slot="progress-indicator"
48       className={cn('bg-primary h-full transition-all', className)}
49       {...props}
50     />
51   );
52 }
53
54 function ProgressLabel({ className, ...props }: ProgressPrimitive.Label.Props) {
55   return (
56     <ProgressPrimitive.Label
57       className={cn('text-sm font-medium', className)}
58       data-slot="progress-label"
59       {...props}
60     />
61   );
62 }
63
64 function ProgressValue({ className, ...props }: ProgressPrimitive.Value.Props) {
65   return (
66     <ProgressPrimitive.Value
67       className={cn(
68         'text-muted-foreground ml-auto text-sm tabular-nums',
69         className,
70       )}
71       data-slot="progress-value"
72       {...props}
73     />
74   );
75 }
76
77 export {
78   Progress,
79   ProgressTrack,
80   ProgressIndicator,
81   ProgressLabel,
82   ProgressValue,
```


components/ui/radio-group.tsx

```
1 'use client';
2
3 import { Radio as RadioPrimitive } from '@base-ui/react/radio';
4 import { RadioGroup as RadioGroupPrimitive } from '@base-ui/react/radio-group';
5
6 import { cn } from '@lib/utils';
7
8 function RadioGroup({ className, ...props }: RadioGroupPrimitive.Props) {
9   return (
10     <RadioGroupPrimitive
11       data-slot="radio-group"
12       className={cn('grid gap-2 w-full', className)}
13       {...props}
14     />
15   );
16 }
17
18 function RadioGroupItem({ className, ...props }: RadioPrimitive.Root.Props) {
19   return (
20     <RadioPrimitive.Root
21       data-slot="radio-group-item"
22       className={cn(
23         'border-input dark:bg-input/30 data-checked:bg-primary data-checked:text-primary-foreground
24         dark:data-checked:bg-primary data-checked:border-primary aria-invalid:aria-checked:border-primary aria-
25         invalid:border-destructive focus-visible:border-ring focus-visible:ring-ring/50 aria-invalid:ring-
26         destructive/20 dark:aria-invalid:ring-destructive/40 dark:aria-invalid:border-destructive/50 flex size-4
27         rounded-full focus-visible:ring-3 aria-invalid:ring-3 group-has-[:focus-visible]/field-label:ring-0
28         group-has-[:focus-visible]/field-label:not-data-checked:border-input group-has-[:focus-visible]/field-
29         label:data-checked:border-primary group/radio-group-item peer relative aspect-square shrink-0 border
30         outline-none after:absolute after:-inset-x-3 after:-inset-y-2 disabled:cursor-not-allowed
31         disabled:opacity-50',
32         className,
33       )}
34       {...props}
35     >
36       <RadioPrimitive.Indicator
37         data-slot="radio-group-indicator"
38         className="flex size-4 items-center justify-center"
39       >
40         <span className="bg-primary-foreground absolute top-1/2 left-1/2 size-2 -translate-x-1/2
41         -translate-y-1/2 rounded-full" />
42       </RadioPrimitive.Indicator>
43     </RadioPrimitive.Root>
44   );
45 }
46
47 export { RadioGroup, RadioGroupItem };
```

components/ui/resizable.tsx

```
1 'use client';
2
3 import * as ResizablePrimitive from 'react-resizable-panels';
4
5 import { cn } from '@lib/utils';
6
7 function ResizablePanelGroup({
8   className,
9   ...props
10 }: ResizablePrimitive.GroupProps) {
11   return (
12     <ResizablePrimitive.Group
13       data-slot="resizable-panel-group"
14       className={cn(
15         'flex h-full w-full aria-[orientation=vertical]:flex-col',
16         className,
17       )}
18       {...props}
19     />
20   );
21 }
22
23 function ResizablePanel({ ...props }: ResizablePrimitive.PanelProps) {
24   return <ResizablePrimitive.Panel data-slot="resizable-panel" {...props} />;
25 }
26
27 function ResizableHandle({
28   withHandle,
29   className,
30   ...props
31 }: ResizablePrimitive.SeparatorProps & {
32   withHandle?: boolean;
33 }) {
34   return (
35     <ResizablePrimitive.Separator
36       data-slot="resizable-handle"
37       className={cn(
38         'relative flex w-px items-center justify-center bg-border ring-offset-background after:absolute
after:inset-y-0 after:left-1/2 after:w-1 after:-translate-x-1/2 focus-visible:ring-1 focus-visible:ring-
ring focus-visible:outline-hidden aria-[orientation=horizontal]:h-px
aria-[orientation=horizontal]:w-full aria-[orientation=horizontal]:after:left-0
aria-[orientation=horizontal]:after:h-1 aria-[orientation=horizontal]:after:w-full
aria-[orientation=horizontal]:after:translate-x-0 aria-[orientation=horizontal]:after:-translate-y-1/2
[&[aria-orientation=horizontal]>div]:rotate-90',
39         className,
40       )}
41       {...props}
42     >
43       {withHandle && (
44         <div className="bg-border h-6 w-1 rounded-lg z-10 flex shrink-0" />
45       )}
46     </ResizablePrimitive.Separator>
47   );
48 }
49
50 export { ResizableHandle, ResizablePanel, ResizablePanelGroup };
```

components/ui/scroll-area.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { ScrollArea as ScrollAreaPrimitive } from '@base-ui/react/scroll-area';
5
6 import { cn } from '@lib/utlis';
7
8 function ScrollArea({
9   className,
10  children,
11  ...props
12 }: ScrollAreaPrimitive.Root.Props) {
13   return (
14     <ScrollAreaPrimitive.Root
15       data-slot="scroll-area"
16       className={cn('relative', className)}
17       {...props}
18     >
19       <ScrollAreaPrimitive.Viewport
20         data-slot="scroll-area-viewport"
21         className="size-full rounded-[inherit] transition-[color,box-shadow] outline-none focus-
22         visible:ring-[3px] focus-visible:ring-ring/50 focus-visible:outline-1"
23       >
24         {children}
25       </ScrollAreaPrimitive.Viewport>
26       <ScrollBar />
27     </ScrollAreaPrimitive.Root>
28   );
29 }
30
31 function ScrollBar({
32   className,
33   orientation = 'vertical',
34   ...props
35 }: ScrollAreaPrimitive.Scrollbar.Props) {
36   return (
37     <ScrollAreaPrimitive.Scrollbar
38       data-slot="scroll-area-scrollbar"
39       data-orientation={orientation}
40       orientation={orientation}
41       className={cn(
42         'data-horizontal:h-2.5 data-horizontal:flex-col data-horizontal:border-t data-horizontal:border-
43         t-transparent data-vertical:h-full data-vertical:w-2.5 data-vertical:border-l data-vertical:border-l-
44         transparent flex touch-none p-px transition-colors select-none',
45         className,
46       )}
47       {...props}
48     >
49       <ScrollAreaPrimitive.Thumb
50         data-slot="scroll-area-thumb"
51         className="rounded-full relative flex-1 bg-border"
52       />
53     </ScrollAreaPrimitive.Scrollbar>
54   );
55 }
56
57 export { ScrollArea, ScrollBar };
```

components/ui/select.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { Select as SelectPrimitive } from '@base-ui/react/select';
5
6 import { cn } from '@lib/utils';
7 import { ChevronDownIcon, CheckIcon, ChevronUpIcon } from 'lucide-react';
8
9 const Select = SelectPrimitive.Root;
10
11 function SelectGroup({ className, ...props }: SelectPrimitive.Group.Props) {
12   return (
13     <SelectPrimitive.Group
14       data-slot="select-group"
15       className={cn('scroll-my-1 p-1', className)}
16       {...props}
17     />
18   );
19 }
20
21 function SelectValue({ className, ...props }: SelectPrimitive.Value.Props) {
22   return (
23     <SelectPrimitive.Value
24       data-slot="select-value"
25       className={cn('flex flex-1 text-left', className)}
26       {...props}
27     />
28   );
29 }
30
31 function SelectTrigger({
32   className,
33   size = 'default',
34   children,
35   ...props
36 }: SelectPrimitive.Trigger.Props & {
37   size?: 'sm' | 'default';
38 }) {
39   return (
40     <SelectPrimitive.Trigger
41       data-slot="select-trigger"
42       data-size={size}
43       className={cn(
44         "flex w-fit items-center justify-between gap-1.5 rounded-lg border border-input bg-transparent
45         py-2 pr-2 pl-2.5 text-sm whitespace-nowrap transition-colors outline-none select-none focus-
46         visible:border-ring focus-visible:ring-3 focus-visible:ring-ring/50 disabled:cursor-not-allowed
47         disabled:opacity-50 aria-invalid:border-destructive aria-invalid:ring-3 aria-invalid:ring-destructive/20
48         data-placeholder:text-muted-foreground data-[size=default]:h-8 data-[size=sm]:h-7
49         data-[size=sm]:rounded-[min(var(--radius-md),10px)] *:data-[slot=select-value]:line-clamp-1
50         *:data-[slot=select-value]:flex *:data-[slot=select-value]:items-center *:data-[slot=select-
51         value]:gap-1.5 dark:bg-input/30 dark:hover:bg-input/50 dark:aria-invalid:border-destructive/50
52         dark:aria-invalid:ring-destructive/40 [&_svg]:pointer-events-none [&_svg]:shrink-0
53         [&_svg]:not([class*='size-']):size-4",
54         className,
55       )}
56       {...props}
57     >
58       {children}
59       <SelectPrimitive.Icon
60         render={
61           <ChevronDownIcon className="pointer-events-none size-4 text-muted-foreground" />
62         }
63       />
64     </SelectPrimitive.Trigger>
65   );
66 }
67
68 function SelectContent({
69   className,
70   children,
71   side = 'bottom',
72   sideOffset = 4,
73   align = 'center',
74   alignOffset = 0,
75   alignItemWithTrigger = true,
76   ...props
77 }: SelectPrimitive.Popup.Props &
78 Pick<
79   SelectPrimitive.Positioner.Props,
80   'align' | 'alignOffset' | 'side' | 'sideOffset' | 'alignItemWithTrigger'
81 >) {
82   return (
```

```

74     <SelectPrimitive.Portal>
75     <SelectPrimitive.Positioner
76         side={side}
77         sideOffset={sideOffset}
78         align={align}
79         alignOffset={alignOffset}
80         alignItemWithTrigger={alignItemWithTrigger}
81         className="isolate z-50"
82     >
83     <SelectPrimitive.Popup
84         data-slot="select-content"
85         data-align-trigger={alignItemWithTrigger}
86         className={cn(
87             'relative isolate z-50 max-h(--available-height) w(--anchor-width) min-w-36
origin-(--transform-origin) overflow-x-hidden overflow-y-auto rounded-lg bg-popover text-popover-
foreground shadow-md ring-1 ring-foreground/10 duration-100 data-[align-trigger=true]:animate-none
data-[side=bottom]:slide-in-from-top-2 data-[side=inline-end]:slide-in-from-left-2 data-[side=inline-
start]:slide-in-from-right-2 data-[side=left]:slide-in-from-right-2 data-[side=right]:slide-in-from-
left-2 data-[side=top]:slide-in-from-bottom-2 data-open:animate-in data-open:fade-in-0 data-open:zoom-
in-95 data-closed:animate-out data-closed:fade-out-0 data-closed:zoom-out-95',
88             className,
89         )}
90         {...props}
91     >
92     <SelectScrollUpButton />
93     <SelectPrimitive.List>{children}</SelectPrimitive.List>
94     <SelectScrollDownButton />
95 </SelectPrimitive.Popup>
96 </SelectPrimitive.Positioner>
97 </SelectPrimitive.Portal>
98 );
99 }
100
101 function SelectLabel({
102     className,
103     ...props
104 }: SelectPrimitive.GroupLabel.Props) {
105     return (
106         <SelectPrimitive.GroupLabel
107             data-slot="select-label"
108             className={cn('px-1.5 py-1 text-xs text-muted-foreground', className)}
109             {...props}
110         />
111     );
112 }
113
114 function SelectItem({
115     className,
116     children,
117     ...props
118 }: SelectPrimitive.Item.Props) {
119     return (
120         <SelectPrimitive.Item
121             data-slot="select-item"
122             className={cn(
123                 "relative flex w-full cursor-default items-center gap-1.5 rounded-md py-1 pr-8 pl-1.5 text-sm
outline-hidden select-none focus:bg-accent focus:text-accent-foreground not-
data-[variant=destructive]:focus:*:text-accent-foreground data-disabled:pointer-events-none data-
disabled:opacity-50 [&_svg]:pointer-events-none [&_svg]:shrink-0 [&_svg]:not([class*='size-'])]:size-4
*:[span]:last:flex *:[span]:last:items-center *:[span]:last:gap-2",
124                 className,
125             )}
126             {...props}
127         >
128         <SelectPrimitive.ItemText className="flex flex-1 shrink-0 gap-2 whitespace-nowrap">
129             {children}
130         </SelectPrimitive.ItemText>
131         <SelectPrimitive.ItemIndicator>
132             render={
133                 <span className="pointer-events-none absolute right-2 flex size-4 items-center justify-center"
134                 />
135             }
136         </SelectPrimitive.ItemIndicator>
137     </SelectPrimitive.Item>
138 );
139 }
140 }
141
142 function SelectSeparator({
143     className,
144     ...props
145 }: SelectPrimitive.Separator.Props) {
146     return (
147         <SelectPrimitive.Separator

```

```

148     data-slot="select-separator"
149     className={cn('pointer-events-none -mx-1 my-1 h-px bg-border', className)}
150     {...props}
151   />
152 );
153 }
154
155 function SelectScrollUpButton({
156   className,
157   ...props
158 }): React.ComponentProps<typeof SelectPrimitive.ScrollUpArrow> {
159   return (
160     <SelectPrimitive.ScrollUpArrow
161       data-slot="select-scroll-up-button"
162       className={cn(
163         "top-0 z-10 flex w-full cursor-default items-center justify-center bg-popover py-1
164         [&_svg:not([class*='size-']):size-4",
165         className,
166       )}
167       {...props}
168     >
169     <ChevronUpIcon />
170   );
171 }
172
173 function SelectScrollDownButton({
174   className,
175   ...props
176 }): React.ComponentProps<typeof SelectPrimitive.ScrollDownArrow> {
177   return (
178     <SelectPrimitive.ScrollDownArrow
179       data-slot="select-scroll-down-button"
180       className={cn(
181         "bottom-0 z-10 flex w-full cursor-default items-center justify-center bg-popover py-1
182         [&_svg:not([class*='size-']):size-4",
183         className,
184       )}
185       {...props}
186     >
187     <ChevronDownIcon />
188   );
189 }
190
191 export {
192   Select,
193   SelectContent,
194   SelectGroup,
195   SelectItem,
196   SelectLabel,
197   SelectScrollDownButton,
198   SelectScrollUpButton,
199   SelectSeparator,
200   SelectTrigger,
201   SelectValue,
202 };

```

components/ui/separator.tsx

```
1 'use client';
2
3 import { Separator as SeparatorPrimitive } from '@base-ui/react/separator';
4
5 import { cn } from '@lib/utils';
6
7 function Separator({
8   className,
9   orientation = 'horizontal',
10  ...props
11 }: SeparatorPrimitive.Props) {
12   return (
13     <SeparatorPrimitive
14       data-slot="separator"
15       orientation={orientation}
16       className={cn(
17         'shrink-0 bg-border data-horizontal:h-py data-horizontal:w-full data-vertical:w-py data-
vertical:self-stretch',
18         className,
19       )}
20       {...props}
21     />
22   );
23 }
24
25 export { Separator };
```

components/ui/sheet.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { Dialog as SheetPrimitive } from '@base-ui/react/dialog';
5
6 import { cn } from '@lib/utls';
7 import { Button } from '@components/ui/button';
8 import { XIcon } from 'lucide-react';
9
10 function Sheet({ ...props }: SheetPrimitive.Root.Props) {
11   return <SheetPrimitive.Root data-slot="sheet" {...props} />;
12 }
13
14 function SheetTrigger({ ...props }: SheetPrimitive.Trigger.Props) {
15   return <SheetPrimitive.Trigger data-slot="sheet-trigger" {...props} />;
16 }
17
18 function SheetClose({ ...props }: SheetPrimitive.Close.Props) {
19   return <SheetPrimitive.Close data-slot="sheet-close" {...props} />;
20 }
21
22 function SheetPortal({ ...props }: SheetPrimitive.Portal.Props) {
23   return <SheetPrimitive.Portal data-slot="sheet-portal" {...props} />;
24 }
25
26 function SheetOverlay({ className, ...props }: SheetPrimitive.Backdrop.Props) {
27   return (
28     <SheetPrimitive.Backdrop
29       data-slot="sheet-overlay"
30       className={cn(
31         'bg-black/10 supports-backdrop-filter:backdrop-blur-xs fixed inset-0 z-50 transition-opacity
32         duration-150 data-ending-style:opacity-0 data-starting-style:opacity-0',
33         className,
34       )}
35     >
36     <SheetPrimitive.Backdrop data-slot="sheet-overlay" data-starting-style:opacity-0 data-ending-style:opacity-0
37       {...props}
38     />
39   );
40 }
41
42 function SheetContent({
43   className,
44   children,
45   side = 'right',
46   showCloseButton = true,
47   ...props
48 }: SheetPrimitive.Popup.Props & {
49   side?: 'top' | 'right' | 'bottom' | 'left';
50   showCloseButton?: boolean;
51 }) {
52   return (
53     <SheetPortal>
54       <SheetOverlay />
55       <SheetPrimitive.Popup
56         data-slot="sheet-content"
57         data-side={side}
58         className={cn(
59           'bg-popover text-popover-foreground fixed z-50 flex flex-col gap-4 bg-clip-padding text-sm
60           shadow-lg transition duration-200 ease-in-out data-[side=bottom]:inset-x-0 data-[side=bottom]:bottom-0
61           data-[side=bottom]:h-auto data-[side=bottom]:border-t data-[side=left]:inset-y-0 data-[side=left]:left-0
62           data-[side=left]:h-full data-[side=left]:w-3/4 data-[side=left]:border-r data-[side=right]:inset-y-0
63           data-[side=right]:right-0 data-[side=right]:h-full data-[side=right]:w-3/4 data-[side=right]:border-l
64           data-[side=top]:inset-x-0 data-[side=top]:top-0 data-[side=top]:h-auto data-[side=top]:border-b
65           data-[side=left]:sm:max-w-sm data-[side=right]:sm:max-w-sm data-ending-style:opacity-0 data-starting-
66           style:opacity-0 data-[side=bottom]:data-ending-style:translate-y-[2.5rem] data-[side=bottom]:data-
67           starting-style:translate-y-[-2.5rem] data-[side=left]:data-ending-style:translate-x-[-2.5rem]
68           data-[side=left]:data-starting-style:translate-x-[2.5rem] data-[side=right]:data-ending-
69           style:translate-x-[2.5rem] data-[side=right]:data-starting-style:translate-x-[-2.5rem]
70           data-[side=top]:data-ending-style:translate-y-[-2.5rem] data-[side=top]:data-starting-
71           style:translate-y-[2.5rem]',
72           className,
73         )}
74       >
75         {children}
76         {showCloseButton && (
77           <SheetPrimitive.Close
78             data-slot="sheet-close"
79             render={
80               <Button
81                 variant="ghost"
82                 className="absolute top-3 right-3"
83                 size="icon-sm"
84               />
85             )
86           }
87       )
88     </SheetPrimitive.Popup>
89   );
90 }
```

```

70         />
71     }
72     >
73     <XIcon />
74     <span className="sr-only">Close</span>
75     </SheetPrimitive.Close>
76   )}
77   </SheetPrimitive.Popup>
78 </SheetPortal>
79 );
80 }
81
82 function SheetHeader({ className, ...props }: React.ComponentProps<'div'>) {
83   return (
84     <div
85       data-slot="sheet-header"
86       className={cn('gap-0.5 p-4 flex flex-col', className)}
87       {...props}
88     />
89   );
90 }
91
92 function SheetFooter({ className, ...props }: React.ComponentProps<'div'>) {
93   return (
94     <div
95       data-slot="sheet-footer"
96       className={cn('gap-2 p-4 mt-auto flex flex-col', className)}
97       {...props}
98     />
99   );
100 }
101
102 function SheetTitle({ className, ...props }: SheetPrimitive.Title.Props) {
103   return (
104     <SheetPrimitive.Title
105       data-slot="sheet-title"
106       className={cn(
107         'text-foreground text-base font-medium cn-font-heading',
108         className,
109       )}
110       {...props}
111     />
112   );
113 }
114
115 function SheetDescription({
116   className,
117   ...props
118 }: SheetPrimitive.Description.Props) {
119   return (
120     <SheetPrimitive.Description
121       data-slot="sheet-description"
122       className={cn('text-muted-foreground text-sm', className)}
123       {...props}
124     />
125   );
126 }
127
128 export {
129   Sheet,
130   SheetTrigger,
131   SheetClose,
132   SheetContent,
133   SheetHeader,
134   SheetFooter,
135   SheetTitle,
136   SheetDescription,
137 };

```

components/ui/sidebar.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { mergeProps } from '@base-ui/react/merge-props';
5 import { useRender } from '@base-ui/react/use-render';
6 import { cva, type VariantProps } from 'class-variance-authority';
7
8 import { useIsMobile } from '@/hooks/use-mobile';
9 import { cn } from '@/lib/utils';
10 import { Button } from '@components/ui/button';
11 import { Input } from '@components/ui/input';
12 import { Separator } from '@components/ui/separator';
13 import {
14   Sheet,
15   SheetContent,
16   SheetDescription,
17   SheetHeader,
18   SheetTitle,
19 } from '@components/ui/sheet';
20 import { Skeleton } from '@components/ui/skeleton';
21 import {
22   Tooltip,
23   TooltipContent,
24   TooltipTrigger,
25 } from '@components/ui/tooltip';
26 import { PanelLeftIcon } from 'lucide-react';
27
28 const SIDEBAR_COOKIE_NAME = 'sidebar_state';
29 const SIDEBAR_COOKIE_MAX_AGE = 60 * 60 * 24 * 7;
30 const SIDEBAR_WIDTH = '16rem';
31 const SIDEBAR_WIDTH_MOBILE = '18rem';
32 const SIDEBAR_WIDTH_ICON = '3rem';
33 const SIDEBAR_KEYBOARD_SHORTCUT = 'b';
34
35 type SidebarContextProps = {
36   state: 'expanded' | 'collapsed';
37   open: boolean;
38   setOpen: (open: boolean) => void;
39   openMobile: boolean;
40   setOpenMobile: (open: boolean) => void;
41   isMobile: boolean;
42   toggleSidebar: () => void;
43 };
44
45 const SidebarContext = React.createContext<SidebarContextProps | null>(null);
46
47 function useSidebar() {
48   const context = React.useContext(SidebarContext);
49   if (!context) {
50     throw new Error('useSidebar must be used within a SidebarProvider.');
```

```

83     }
84
85     // This sets the cookie to keep the sidebar state.
86     document.cookie = `${SIDEBAR_COOKIE_NAME}=${openState}; path=/; max-
age=${SIDEBAR_COOKIE_MAX_AGE}`;
87   },
88   [setOpenProp, open],
89 );
90
91 // Helper to toggle the sidebar.
92 const toggleSidebar = React.useCallback(() => {
93   return isMobile ? setOpenMobile(open) => !open : setOpen((open) => !open);
94 }, [isMobile, setOpen, setOpenMobile]);
95
96 // Adds a keyboard shortcut to toggle the sidebar.
97 React.useEffect(() => {
98   const handleKeyDown = (event: KeyboardEvent) => {
99     if (
100       event.key === SIDEBAR_KEYBOARD_SHORTCUT &&
101       (event.metaKey || event.ctrlKey)
102     ) {
103       event.preventDefault();
104       toggleSidebar();
105     }
106   };
107
108   window.addEventListener('keydown', handleKeyDown);
109   return () => window.removeEventListener('keydown', handleKeyDown);
110 }, [toggleSidebar]);
111
112 // We add a state so that we can do data-state="expanded" or "collapsed".
113 // This makes it easier to style the sidebar with Tailwind classes.
114 const state = open ? 'expanded' : 'collapsed';
115
116 const contextValue = React.useMemo<SidebarContextProps>(
117   () => ({
118     state,
119     open,
120     setOpen,
121     isMobile,
122     openMobile,
123     setOpenMobile,
124     toggleSidebar,
125   }),
126   [state, open, setOpen, isMobile, openMobile, setOpenMobile, toggleSidebar],
127 );
128
129 return (
130   <SidebarContext.Provider value={contextValue}>
131     <div
132       data-slot="sidebar-wrapper"
133       style={
134         {
135           '--sidebar-width': SIDEBAR_WIDTH,
136           '--sidebar-width-icon': SIDEBAR_WIDTH_ICON,
137           ...style,
138         } as React.CSSProperties
139       >
140       <div
141         className={cn(
142           'group/sidebar-wrapper flex min-h-svh w-full has-data-[variant=inset]:bg-sidebar',
143           className,
144         )}
145         {...props}
146       >
147         {children}
148       </div>
149     </SidebarContext.Provider>
150   );
151
152 function Sidebar({
153   side = 'left',
154   variant = 'sidebar',
155   collapsible = 'offcanvas',
156   className,
157   children,
158   dir,
159   ...props
160 }: React.ComponentProps<'div'> & {
161   side?: 'left' | 'right';
162   variant?: 'sidebar' | 'floating' | 'inset';
163   collapsible?: 'offcanvas' | 'icon' | 'none';
164 }) {
165   const { isMobile, state, openMobile, setOpenMobile } = useSidebar();
166

```

```

167   if (collapsible === 'none') {
168     return (
169       <div
170         data-slot="sidebar"
171         className={cn(
172           'flex h-full w-(--sidebar-width) flex-col bg-sidebar text-sidebar-foreground',
173           className,
174         )}
175         {...props}
176       >
177         {children}
178       </div>
179     );
180   }
181
182   if (isMobile) {
183     return (
184       <Sheet open={openMobile} onOpenChange={setOpenMobile} {...props}>
185         <SheetContent
186           dir={dir}
187           data-sidebar="sidebar"
188           data-slot="sidebar"
189           data-mobile="true"
190           className="w-(--sidebar-width) bg-sidebar p-0 text-sidebar-foreground [&button]:hidden"
191           style={
192             {
193               '--sidebar-width': SIDEBAR_WIDTH_MOBILE,
194             } as React.CSSProperties
195           }
196           side={side}
197         >
198           <SheetHeader className="sr-only">
199             <SheetTitle>Sidebar</SheetTitle>
200             <SheetDescription>Displays the mobile sidebar.</SheetDescription>
201           </SheetHeader>
202           <div className="flex h-full w-full flex-col">{children}</div>
203         </SheetContent>
204       </Sheet>
205     );
206   }
207
208   return (
209     <div
210       className="group peer hidden text-sidebar-foreground md:block"
211       data-state={state}
212       data-collapsible={state === 'collapsed' ? collapsible : ''}
213       data-variant={variant}
214       data-side={side}
215       data-slot="sidebar"
216     >
217       {/* This is what handles the sidebar gap on desktop */}
218       <div
219         data-slot="sidebar-gap"
220         className={cn(
221           'transition-[width] duration-200 ease-linear relative w-(--sidebar-width) bg-transparent',
222           'group-data-[collapsible=offcanvas]:w-0',
223           'group-data-[side=right]:rotate-180',
224           variant === 'floating' || variant === 'inset'
225             ? 'group-data-[collapsible=icon]:w-[calc(var(--sidebar-width-icon)+(--spacing(4)))]'
226             : 'group-data-[collapsible=icon]:w-(--sidebar-width-icon)',
227         )}
228       >
229         <div
230           data-slot="sidebar-container"
231           data-side={side}
232           className={cn(
233             'fixed inset-y-0 z-10 hidden h-svh w-(--sidebar-width) transition-[left,right,width]
duration-200 ease-linear data-[side=left]:left-0 data-[side=left]:group-
data-[collapsible=offcanvas]:left-[calc(var(--sidebar-width)*-1)] data-[side=right]:right-0
data-[side=right]:group-data-[collapsible=offcanvas]:right-[calc(var(--sidebar-width)*-1)] md:flex',
234             // Adjust the padding for floating and inset variants.
235             variant === 'floating' || variant === 'inset'
236               ? 'p-2 group-data-[collapsible=icon]:w-[calc(var(--sidebar-width-icon)+(--spacing(4))+2px)]'
237               : 'group-data-[collapsible=icon]:w-(--sidebar-width-icon) group-data-[side=left]:border-r
group-data-[side=right]:border-l',
238             className,
239           )}
240         {...props}
241       >
242         <div
243           data-sidebar="sidebar"
244           data-slot="sidebar-inner"
245           className="bg-sidebar group-data-[variant=floating]:ring-sidebar-border group-
data-[variant=floating]:rounded-lg group-data-[variant=floating]:shadow-sm group-
data-[variant=floating]:ring-1 flex size-full flex-col"

```

```

246         >
247         {children}
248     </div>
249 </div>
250 </div>
251 );
252 }
253
254 function SidebarTrigger({
255     className,
256     onClick,
257     ...props
258 }: React.ComponentProps<typeof Button>) {
259     const { toggleSidebar } = useSidebar();
260
261     return (
262         <Button
263             data-sidebar="trigger"
264             data-slot="sidebar-trigger"
265             variant="ghost"
266             size="icon-sm"
267             className={cn(className)}
268             onClick={(event) => {
269                 onClick?.(event);
270                 toggleSidebar();
271             }}
272             {...props}
273         >
274             <PanelLeftIcon className="cn-rtl-flip" />
275             <span className="sr-only">Toggle Sidebar</span>
276         </Button>
277     );
278 }
279
280 function SidebarRail({ className, ...props }: React.ComponentProps<'button'>) {
281     const { toggleSidebar } = useSidebar();
282
283     return (
284         <button
285             data-sidebar="rail"
286             data-slot="sidebar-rail"
287             aria-label="Toggle Sidebar"
288             tabIndex={-1}
289             onClick={toggleSidebar}
290             title="Toggle Sidebar"
291             className={cn(
292                 'hover:after:bg-sidebar-border absolute inset-y-0 z-20 hidden w-4 transition-all ease-linear
group-data-[side=left]:-right-4 group-data-[side=right]:left-0 after:absolute after:inset-y-0
after:start-1/2 after:w-[2px] sm:flex ltr:-translate-x-1/2 rtl:-translate-x-1/2',
293                 'in-data-[side=left]:cursor-w-resize in-data-[side=right]:cursor-e-resize',
294                 '[[data-side=left][data-state=collapsed]_&]:cursor-e-resize [[data-side=right][data-
state=collapsed]_&]:cursor-w-resize',
295                 'group-data-[collapsible=offcanvas]:translate-x-0 group-data-[collapsible=offcanvas]:after:left-
full hover:group-data-[collapsible=offcanvas]:bg-sidebar',
296                 '[[data-side=left][data-collapsible=offcanvas]_&]:-right-2',
297                 '[[data-side=right][data-collapsible=offcanvas]_&]:-left-2',
298                 className,
299             )}
300             {...props}
301         />
302     );
303 }
304
305 function SidebarInset({ className, ...props }: React.ComponentProps<'main'>) {
306     return (
307         <main
308             data-slot="sidebar-inset"
309             className={cn(
310                 'bg-background md:peer-data-[variant=inset]:m-2 md:peer-data-[variant=inset]:ml-0 md:peer-
data-[variant=inset]:rounded-xl md:peer-data-[variant=inset]:shadow-sm md:peer-
data-[variant=inset]:peer-data-[state=collapsed]:ml-2 relative flex w-full flex-1 flex-col',
311                 className,
312             )}
313             {...props}
314         />
315     );
316 }
317
318 function SidebarInput({
319     className,
320     ...props
321 }: React.ComponentProps<typeof Input>) {
322     return (
323         <Input
324             data-slot="sidebar-input"

```

```

325     data-sidebar="input"
326     className={cn('bg-background h-8 w-full shadow-none', className)}
327     {...props}
328   />
329 );
330 }
331
332 function SidebarHeader({ className, ...props }: React.ComponentProps<'div'>) {
333   return (
334     <div
335       data-slot="sidebar-header"
336       data-sidebar="header"
337       className={cn('gap-2 p-2 flex flex-col', className)}
338       {...props}
339     />
340   );
341 }
342
343 function SidebarFooter({ className, ...props }: React.ComponentProps<'div'>) {
344   return (
345     <div
346       data-slot="sidebar-footer"
347       data-sidebar="footer"
348       className={cn('gap-2 p-2 flex flex-col', className)}
349       {...props}
350     />
351   );
352 }
353
354 function SidebarSeparator({
355   className,
356   ...props
357 }: React.ComponentProps<typeof Separator>) {
358   return (
359     <Separator
360       data-slot="sidebar-separator"
361       data-sidebar="separator"
362       className={cn('bg-sidebar-border mx-2 w-auto', className)}
363       {...props}
364     />
365   );
366 }
367
368 function SidebarContent({ className, ...props }: React.ComponentProps<'div'>) {
369   return (
370     <div
371       data-slot="sidebar-content"
372       data-sidebar="content"
373       className={cn(
374         'no-scrollbar gap-0 flex min-h-0 flex-1 flex-col overflow-auto group-
data-[collapsible=icon]:overflow-hidden',
375         className,
376       )}
377       {...props}
378     />
379   );
380 }
381
382 function SidebarGroup({ className, ...props }: React.ComponentProps<'div'>) {
383   return (
384     <div
385       data-slot="sidebar-group"
386       data-sidebar="group"
387       className={cn('p-2 relative flex w-full min-w-0 flex-col', className)}
388       {...props}
389     />
390   );
391 }
392
393 function SidebarGroupLabel({
394   className,
395   render,
396   ...props
397 }: useRender.ComponentProps<'div'> & React.ComponentProps<'div'>) {
398   return useRender({
399     defaultTagName: 'div',
400     props: mergeProps<'div'>({
401       {
402         className: cn(
403           'text-sidebar-foreground/70 ring-sidebar-ring h-8 rounded-md px-2 text-xs font-medium
transition-[margin,opacity] duration-200 ease-linear group-data-[collapsible=icon]:-mt-8 group-
data-[collapsible=icon]:opacity-0 focus-visible:ring-2 [&svg]:size-4 flex shrink-0 items-center
outline-hidden [&svg]:shrink-0',
404         className,
405       ),

```

```

406     },
407     props,
408   ),
409   render,
410   state: {
411     slot: 'sidebar-group-label',
412     sidebar: 'group-label',
413   },
414 });
415 }
416
417 function SidebarGroupAction({
418   className,
419   render,
420   ...props
421 }: useRender.ComponentProps<'button'> & React.ComponentProps<'button'>) {
422   return useRender({
423     defaultTagName: 'button',
424     props: mergeProps<'button'>({
425       {
426         className: cn(
427           'text-sidebar-foreground ring-sidebar-ring hover:bg-sidebar-accent hover:text-sidebar-accent-
foreground absolute top-3.5 right-3 w-5 rounded-md p-0 focus-visible:ring-2 [>svg]:size-4 flex aspect-
square items-center justify-center outline-hidden transition-transform group-
data-[collapsible=icon]:hidden after:absolute after:-inset-2 md:after:hidden [>svg]:shrink-0',
428         className,
429       ),
430     },
431     props,
432   ),
433   render,
434   state: {
435     slot: 'sidebar-group-action',
436     sidebar: 'group-action',
437   },
438 });
439 }
440
441 function SidebarGroupContent({
442   className,
443   ...props
444 }: React.ComponentProps<'div'>) {
445   return (
446     <div
447       data-slot="sidebar-group-content"
448       data-sidebar="group-content"
449       className={cn('text-sm w-full', className)}
450       {...props}
451     />
452   );
453 }
454
455 function SidebarMenu({ className, ...props }: React.ComponentProps<'ul'>) {
456   return (
457     <ul
458       data-slot="sidebar-menu"
459       data-sidebar="menu"
460       className={cn('gap-0 flex w-full min-w-0 flex-col', className)}
461       {...props}
462     />
463   );
464 }
465
466 function SidebarMenuItem({ className, ...props }: React.ComponentProps<'li'>) {
467   return (
468     <li
469       data-slot="sidebar-menu-item"
470       data-sidebar="menu-item"
471       className={cn('group/menu-item relative', className)}
472       {...props}
473     />
474   );
475 }
476
477 const sidebarMenuButtonVariants = cva(
478   'ring-sidebar-ring hover:bg-sidebar-accent hover:text-sidebar-accent-foreground active:bg-sidebar-
accent active:text-sidebar-accent-foreground data-active:bg-sidebar-accent data-active:text-sidebar-
accent-foreground data-open: hover:bg-sidebar-accent data-open: hover:text-sidebar-accent-foreground gap-2
rounded-md p-2 text-left text-sm transition-[width,height,padding] group-has-data-[sidebar=menu-
action]/menu-item:pr-8 group-data-[collapsible=icon]:size-8! group-data-[collapsible=icon]:p-2! focus-
visible:ring-2 data-active:font-medium peer/menu-button group/menu-button flex w-full items-center
overflow-hidden outline-hidden disabled:pointer-events-none disabled:opacity-50 aria-disabled:pointer-
events-none aria-disabled:opacity-50 [>svg]:size-4 [>svg]:shrink-0 [>span:last-child]:truncate',
479   {
480     variants: {

```

```

481     variant: {
482       default: 'hover:bg-sidebar-accent hover:text-sidebar-accent-foreground',
483       outline:
484         'bg-background hover:bg-sidebar-accent hover:text-sidebar-accent-foreground
shadow-[0_0_1px_var(--sidebar-border)] hover:shadow-[0_0_1px_var(--sidebar-accent)]',
485     },
486     size: {
487       default: 'h-8 text-sm',
488       sm: 'h-7 text-xs',
489       lg: 'h-12 text-sm group-data-[collapsible=icon]:p-0!',
490     },
491   },
492   defaultVariants: {
493     variant: 'default',
494     size: 'default',
495   },
496 },
497 );
498
499 function SidebarMenuButton({
500   render,
501   isActive = false,
502   variant = 'default',
503   size = 'default',
504   tooltip,
505   className,
506   ...props
507 }: useRender.ComponentProps<'button'> &
508   React.ComponentProps<'button'> & {
509     isActive?: boolean;
510     tooltip?: string | React.ComponentProps<typeof TooltipContent>;
511   } & VariantProps<typeof sidebarMenuButtonVariants>) {
512   const { isMobile, state } = useSidebar();
513   const comp = useRender({
514     defaultTagName: 'button',
515     props: mergeProps<'button'>({
516       {
517         className: cn(sidebarMenuButtonVariants({ variant, size }), className),
518       },
519       props,
520     },
521     render: !tooltip ? render : <TooltipTrigger render={render} />,
522     state: {
523       slot: 'sidebar-menu-button',
524       sidebar: 'menu-button',
525       size,
526       active: isActive,
527     },
528   });
529
530   if (!tooltip) {
531     return comp;
532   }
533
534   if (typeof tooltip === 'string') {
535     tooltip = {
536       children: tooltip,
537     };
538   }
539
540   return (
541     <Tooltip>
542       {comp}
543       <TooltipContent
544         side="right"
545         align="center"
546         hidden={state !== 'collapsed' || isMobile}
547         {...tooltip}
548       />
549     </Tooltip>
550   );
551 }
552
553 function SidebarMenuAction({
554   className,
555   render,
556   showOnHover = false,
557   ...props
558 }: useRender.ComponentProps<'button'> &
559   React.ComponentProps<'button'> & {
560     showOnHover?: boolean;
561   }) {
562   return useRender({
563     defaultTagName: 'button',
564     props: mergeProps<'button'>({

```

```

565     {
566         className: cn(
567             'text-sidebar-foreground ring-sidebar-ring hover:bg-sidebar-accent hover:text-sidebar-accent-
foreground peer-hover/menu-button:text-sidebar-accent-foreground absolute top-1.5 right-1 aspect-square
w-5 rounded-md p-0 peer-data-[size=default]/menu-button:top-1.5 peer-data-[size=lg]/menu-button:top-2.5
peer-data-[size=sm]/menu-button:top-1 focus-visible:ring-2 [>svg]:size-4 flex items-center justify-
center outline-hidden transition-transform group-data-[collapsible=icon]:hidden after:absolute
after:-inset-2 md:after:hidden [>svg]:shrink-0',
568             showOnHover &&
569             'group-focus-within/menu-item:opacity-100 group-hover/menu-item:opacity-100 peer-data-
active/menu-button:text-sidebar-accent-foreground aria-expanded:opacity-100 md:opacity-0',
570             className,
571         ),
572     },
573     props,
574 ),
575 render,
576 state: {
577     slot: 'sidebar-menu-action',
578     sidebar: 'menu-action',
579 },
580 });
581 }
582
583 function SidebarMenuBadge({
584     className,
585     ...props
586 }): React.ComponentProps<'div'> {
587     return (
588         <div
589             data-slot="sidebar-menu-badge"
590             data-sidebar="menu-badge"
591             className={cn(
592                 'text-sidebar-foreground peer-hover/menu-button:text-sidebar-accent-foreground peer-data-
active/menu-button:text-sidebar-accent-foreground pointer-events-none absolute right-1 h-5 min-w-5
rounded-md px-1 text-xs font-medium peer-data-[size=default]/menu-button:top-1.5 peer-
data-[size=lg]/menu-button:top-2.5 peer-data-[size=sm]/menu-button:top-1 flex items-center justify-
center tabular-nums select-none group-data-[collapsible=icon]:hidden',
593                 className,
594             )}
595             {...props}
596         />
597     );
598 }
599
600 function SidebarMenuSkeleton({
601     className,
602     showIcon = false,
603     ...props
604 }): React.ComponentProps<'div'> & {
605     showIcon?: boolean;
606 }) {
607     // Random width between 50 to 90%.
608     const [width] = React.useState(() => {
609         return `${Math.floor(Math.random() * 40) + 50}%`;
610     });
611
612     return (
613         <div
614             data-slot="sidebar-menu-skeleton"
615             data-sidebar="menu-skeleton"
616             className={cn('h-8 gap-2 rounded-md px-2 flex items-center', className)}
617             {...props}
618         >
619             {showIcon && (
620                 <Skeleton
621                     className="size-4 rounded-md"
622                     data-sidebar="menu-skeleton-icon"
623                 />
624             )}
625             <Skeleton
626                 className="h-4 max-w(--skeleton-width) flex-1"
627                 data-sidebar="menu-skeleton-text"
628                 style={
629                     {
630                         '--skeleton-width': width,
631                     } as React.CSSProperties
632                 }
633             />
634         </div>
635     );
636 }
637
638 function SidebarMenuSub({ className, ...props }): React.ComponentProps<'ul'> {
639     return (

```

```

640     <ul
641       data-slot="sidebar-menu-sub"
642       data-sidebar="menu-sub"
643       className={cn(
644         'border-sidebar-border mx-3.5 translate-x-py gap-1 border-l px-2.5 py-0.5 group-
data-[collapsible=icon]:hidden flex min-w-0 flex-col',
645         className,
646       )}
647       {...props}
648     />
649   );
650 }
651
652 function SidebarMenuSubItem({
653   className,
654   ...props
655 }: React.ComponentProps<'li'>) {
656   return (
657     <li
658       data-slot="sidebar-menu-sub-item"
659       data-sidebar="menu-sub-item"
660       className={cn('group/menu-sub-item relative', className)}
661       {...props}
662     />
663   );
664 }
665
666 function SidebarMenuSubButton({
667   render,
668   size = 'md',
669   isActive = false,
670   className,
671   ...props
672 }: useRender.ComponentProps<'a'> &
React.ComponentProps<'a'> & {
673   size?: 'sm' | 'md';
674   isActive?: boolean;
675 }) {
676   return useRender({
677     defaultTagName: 'a',
678     props: mergeProps<'a'>({
679       {
680         className: cn(
681           'text-sidebar-foreground ring-sidebar-ring hover:bg-sidebar-accent hover:text-sidebar-accent-
foreground active:bg-sidebar-accent active:text-sidebar-accent-foreground [&svg]:text-sidebar-accent-
foreground data-active:bg-sidebar-accent data-active:text-sidebar-accent-foreground h-7 gap-2 rounded-md
px-2 focus-visible:ring-2 data-[size=md]:text-sm data-[size=sm]:text-xs [&svg]:size-4 flex min-w-0
-translate-x-py items-center overflow-hidden outline-hidden group-data-[collapsible=icon]:hidden
disabled:pointer-events-none disabled:opacity-50 aria-disabled:pointer-events-none aria-
disabled:opacity-50 [&span:last-child]:truncate [&svg]:shrink-0',
682         className,
683       ),
684     },
685     props,
686   ),
687   render,
688   state: {
689     slot: 'sidebar-menu-sub-button',
690     sidebar: 'menu-sub-button',
691     size,
692     active: isActive,
693   },
694 });
695 }
696 }
697
698 export {
699   Sidebar,
700   SidebarContent,
701   SidebarFooter,
702   SidebarGroup,
703   SidebarGroupAction,
704   SidebarGroupContent,
705   SidebarGroupLabel,
706   SidebarHeader,
707   SidebarInput,
708   SidebarInset,
709   SidebarMenu,
710   SidebarMenuAction,
711   SidebarMenuBadge,
712   SidebarMenuButton,
713   SidebarMenuItem,
714   SidebarMenuSkeleton,
715   SidebarMenuSub,
716   SidebarMenuSubButton,
717   SidebarMenuSubItem,

```

```
718 SidebarProvider,  
719 SidebarRail,  
720 SidebarSeparator,  
721 SidebarTrigger,  
722 useSidebar,  
723 };
```

components/ui/skeleton.tsx

```
1 import { cn } from '@lib/utils';
2
3 function Skeleton({ className, ...props }: React.ComponentProps<'div'>) {
4   return (
5     <div
6       data-slot="skeleton"
7       className={cn('animate-pulse rounded-md bg-muted', className)}
8       {...props}
9     />
10  );
11 }
12
13 export { Skeleton };
```

components/ui/slider.tsx

```
1 import { Slider as SliderPrimitive } from '@base-ui/react/slider';
2
3 import { cn } from '@lib/utils';
4
5 function Slider({
6   className,
7   defaultValue,
8   value,
9   min = 0,
10  max = 100,
11  ...props
12 }: SliderPrimitive.Root.Props) {
13   const _values = Array.isArray(value)
14     ? value
15     : Array.isArray(defaultValue)
16       ? defaultValue
17       : [min, max];
18
19   return (
20     <SliderPrimitive.Root
21       className={cn('data-horizontal:w-full data-vertical:h-full', className)}
22       data-slot="slider"
23       defaultValue={defaultValue}
24       value={value}
25       min={min}
26       max={max}
27       thumbAlignment="edge"
28       {...props}
29     >
30       <SliderPrimitive.Control className="data-vertical:min-h-40 relative flex w-full touch-none items-center select-none data-disabled:opacity-50 data-vertical:h-full data-vertical:w-auto data-vertical:flex-col">
31         <SliderPrimitive.Track
32           data-slot="slider-track"
33           className="bg-muted rounded-full data-horizontal:h-1 data-horizontal:w-full data-vertical:h-full data-vertical:w-1 relative grow overflow-hidden select-none"
34         >
35           <SliderPrimitive.Indicator
36             data-slot="slider-range"
37             className="bg-primary select-none data-horizontal:h-full data-vertical:w-full"
38           />
39         </SliderPrimitive.Track>
40         {Array.from({ length: _values.length }, (_, index) => (
41           <SliderPrimitive.Thumb
42             data-slot="slider-thumb"
43             key={index}
44             className="border-ring ring-ring/50 relative size-3 rounded-full border bg-white transition-[color,box-shadow] after:absolute after:-inset-2 hover:ring-3 focus-visible:ring-3 focus-visible:outline-hidden active:ring-3 block shrink-0 select-none disabled:pointer-events-none disabled:opacity-50"
45           />
46         ))}
47       </SliderPrimitive.Control>
48     </SliderPrimitive.Root>
49   );
50 }
51
52 export { Slider };
```

components/ui/spinner.tsx

```
1 import { cn } from '@lib/utils';
2 import { Loader2Icon } from 'lucide-react';
3
4 function Spinner({ className, ...props }: React.ComponentProps<'svg'>) {
5   return (
6     <Loader2Icon
7       data-slot="spinner"
8       role="status"
9       aria-label="Loading"
10      className={cn('size-4 animate-spin', className)}
11      {...props}
12    />
13   );
14 }
15
16 export { Spinner };
```

components/ui/switch.tsx

```
1 'use client';
2
3 import { Switch as SwitchPrimitive } from '@base-ui/react/switch';
4
5 import { cn } from '@lib/utils';
6
7 function Switch({
8   className,
9   size = 'default',
10  ...props
11 }: SwitchPrimitive.Root.Props & {
12   size?: 'sm' | 'default';
13 }) {
14   return (
15     <SwitchPrimitive.Root
16       data-slot="switch"
17       data-size={size}
18       className={cn(
19         'peer group/switch relative inline-flex shrink-0 items-center rounded-full border border-
20         transparent transition-all outline-none group-has-[:focus-visible]/field-label:border-transparent group-
21         has-[:focus-visible]/field-label:ring-0 after:absolute after:-inset-x-3 after:-inset-y-2 focus-
22         visible:border-ring focus-visible:ring-3 focus-visible:ring-ring/50 aria-invalid:border-destructive
23         aria-invalid:ring-3 aria-invalid:ring-destructive/20 data-[size=default]:h-[18.4px]
24         data-[size=default]:w-[32px] data-[size=sm]:h-[14px] data-[size=sm]:w-[24px] dark:aria-invalid:border-
25         destructive/50 dark:aria-invalid:ring-destructive/40 data-checked:bg-primary data-unchecked:bg-input
26         dark:data-unchecked:bg-input/80 data-disabled:cursor-not-allowed data-disabled:opacity-50',
27         className,
28       )}
29     >
30       <SwitchPrimitive.Thumb
31         data-slot="switch-thumb"
32         className="pointer-events-none block rounded-full bg-background ring-0 transition-transform
33         group-data-[size=default]/switch:size-4 group-data-[size=sm]/switch:size-3 group-
34         data-[size=default]/switch:data-checked:translate-x-[calc(100%-2px)] group-data-[size=sm]/switch:data-
35         checked:translate-x-[calc(100%-2px)] dark:data-checked:bg-primary-foreground group-
36         data-[size=default]/switch:data-unchecked:translate-x-0 group-data-[size=sm]/switch:data-
37         unchecked:translate-x-0 dark:data-unchecked:bg-foreground"
38       />
39     </SwitchPrimitive.Root>
40   );
41 }
42
43 export { Switch };
```

components/ui/table.tsx

```
1 'use client';
2
3 import * as React from 'react';
4
5 import { cn } from '@lib/utils';
6
7 function Table({ className, ...props }: React.ComponentProps<'table'>) {
8   return (
9     <div
10       data-slot="table-container"
11       className="relative w-full overflow-x-auto"
12     >
13       <table
14         data-slot="table"
15         className={cn('w-full caption-bottom text-sm', className)}
16         {...props}
17       />
18     </div>
19   );
20 }
21
22 function TableHeader({ className, ...props }: React.ComponentProps<'thead'>) {
23   return (
24     <thead
25       data-slot="table-header"
26       className={cn('[&_tr]:border-b', className)}
27       {...props}
28     />
29   );
30 }
31
32 function TableBody({ className, ...props }: React.ComponentProps<'tbody'>) {
33   return (
34     <tbody
35       data-slot="table-body"
36       className={cn('[&_tr:last-child]:border-0', className)}
37       {...props}
38     />
39   );
40 }
41
42 function TableFooter({ className, ...props }: React.ComponentProps<'tfoot'>) {
43   return (
44     <tfoot
45       data-slot="table-footer"
46       className={cn(
47         'bg-muted/50 border-t font-medium [&>tr]:last:border-b-0',
48         className,
49       )}
50       {...props}
51     />
52   );
53 }
54
55 function TableRow({ className, ...props }: React.ComponentProps<'tr'>) {
56   return (
57     <tr
58       data-slot="table-row"
59       className={cn(
60         'hover:bg-muted/50 data-[state=selected]:bg-muted border-b transition-colors has-aria-expanded:bg-muted/50',
61         className,
62       )}
63       {...props}
64     />
65   );
66 }
67
68 function TableHead({ className, ...props }: React.ComponentProps<'th'>) {
69   return (
70     <th
71       data-slot="table-head"
72       className={cn(
73         'text-foreground h-10 px-2 text-left align-middle font-medium whitespace-nowrap [&:has([role=checkbox])]:pr-0',
74         className,
75       )}
76       {...props}
77     />
78   );
79 }
80
```

```

81 function TableCell({ className, ...props }: React.ComponentProps<'td'>) {
82   return (
83     <td
84       data-slot="table-cell"
85       className={cn(
86         'p-2 align-middle whitespace-nowrap [&:has([role=checkbox])]:pr-0',
87         className,
88       )}
89       {...props}
90     />
91   );
92 }
93
94 function TableCaption({
95   className,
96   ...props
97 }: React.ComponentProps<'caption'>) {
98   return (
99     <caption
100       data-slot="table-caption"
101       className={cn('text-muted-foreground mt-4 text-sm', className)}
102       {...props}
103     />
104   );
105 }
106
107 export {
108   Table,
109   TableHeader,
110   TableBody,
111   TableFooter,
112   TableHead,
113   TableRow,
114   TableCell,
115   TableCaption,
116 };

```

components/ui/tabs.tsx

```
1 'use client';
2
3 import { Tabs as TabsPrimitive } from '@base-ui/react/tabs';
4 import { cva, type VariantProps } from 'class-variance-authority';
5
6 import { cn } from '@lib/utls';
7
8 function Tabs({
9   className,
10  orientation = 'horizontal',
11  ...props
12 }: TabsPrimitive.Root.Props) {
13   return (
14     <TabsPrimitive.Root
15       data-slot="tabs"
16       data-orientation={orientation}
17       className={cn(
18         'group/tabs flex gap-2 data-horizontal:flex-col',
19         className,
20       )}
21       {...props}
22     />
23   );
24 }
25
26 const tabsListVariants = cva(
27   'group/tabs-list inline-flex w-fit items-center justify-center rounded-lg p-[3px] text-muted-foreground group-data-horizontal/tabs:h-8 group-data-vertical/tabs:h-fit group-data-vertical/tabs:flex-col data-[variant=line]:rounded-none',
28   {
29     variants: {
30       variant: {
31         default: 'bg-muted',
32         line: 'gap-1 bg-transparent',
33       },
34     },
35     defaultVariants: {
36       variant: 'default',
37     },
38   },
39 );
40
41 function TabsList({
42   className,
43   variant = 'default',
44   ...props
45 }: TabsPrimitive.List.Props & VariantProps<typeof tabsListVariants>) {
46   return (
47     <TabsPrimitive.List
48       data-slot="tabs-list"
49       data-variant={variant}
50       className={cn(tabsListVariants({ variant }), className)}
51       {...props}
52     />
53   );
54 }
55
56 function TabsTrigger({ className, ...props }: TabsPrimitive.Tab.Props) {
57   return (
58     <TabsPrimitive.Tab
59       data-slot="tabs-trigger"
60       className={cn(
61         "relative inline-flex h-[calc(100%-1px)] flex-1 items-center justify-center gap-1.5 rounded-md border border-transparent px-1.5 py-0.5 text-sm font-medium whitespace-nowrap text-foreground/60 transition-all group-data-vertical/tabs:w-full group-data-vertical/tabs:justify-start hover:text-foreground focus-visible:border-ring focus-visible:ring-[3px] focus-visible:ring-ring/50 focus-visible:outline-1 focus-visible:outline-ring disabled:pointer-events-none disabled:opacity-50 has-data-[icon=inline-end]:pr-1 has-data-[icon=inline-start]:pl-1 aria-disabled:pointer-events-none aria-disabled:opacity-50 dark:text-muted-foreground dark:hover:text-foreground group-data-[variant=default]/tabs-list:data-active:shadow-sm group-data-[variant=line]/tabs-list:data-active:shadow-none [&_svg]:pointer-events-none [&_svg]:shrink-0 [&_svg]:not([class*='size-']):size-4",
62         'group-data-[variant=line]/tabs-list:bg-transparent group-data-[variant=line]/tabs-list:data-active:bg-transparent dark:group-data-[variant=line]/tabs-list:data-active:border-transparent dark:group-data-[variant=line]/tabs-list:data-active:bg-transparent',
63         'data-active:bg-background data-active:text-foreground dark:data-active:border-input dark:data-active:bg-input/30 dark:data-active:text-foreground',
64         'after:absolute after:bg-foreground after:opacity-0 after:transition-opacity group-data-horizontal/tabs:after:inset-x-0 group-data-horizontal/tabs:after:bottom-[-5px] group-data-horizontal/tabs:after:h-0.5 group-data-vertical/tabs:after:inset-y-0 group-data-vertical/tabs:after:-right-1 group-data-vertical/tabs:after:w-0.5 group-data-[variant=line]/tabs-list:data-active:after:opacity-100',
65         className,

```

```
66     })
67     {...props}
68   />
69 );
70 }
71
72 function TabsContent({ className, ...props }: TabsPrimitive.Panel.Props) {
73   return (
74     <TabsPrimitive.Panel
75       data-slot="tabs-content"
76       className={cn('flex-1 text-sm outline-none', className)}
77       {...props}
78     />
79   );
80 }
81
82 export { Tabs, TabsList, TabsTrigger, TabsContent, tabsListVariants };
```

components/ui/textarea.tsx

```
1 import * as React from 'react';
2
3 import { cn } from '@lib/utils';
4
5 function Textarea({ className, ...props }: React.ComponentProps<'textarea'>) {
6   return (
7     <textarea
8       data-slot="textarea"
9       className={cn(
10        'flex field-sizing-content min-h-16 w-full rounded-lg border border-input bg-transparent px-2.5
11        py-2 text-base transition-colors outline-none placeholder:text-muted-foreground focus-visible:border-
12        ring focus-visible:ring-3 focus-visible:ring-ring/50 disabled:cursor-not-allowed disabled:bg-input/50
13        disabled:opacity-50 aria-invalid:border-destructive aria-invalid:ring-3 aria-invalid:ring-destructive/20
14        md:text-sm dark:bg-input/30 dark:disabled:bg-input/80 dark:aria-invalid:border-destructive/50
15        dark:aria-invalid:ring-destructive/40',
16        className,
17      )}
18      {...props}
19    />
20  );
21 }
22
23 export { Textarea };
```

components/ui/toast.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { Toast as ToastPrimitive } from '@base-ui/react/toast';
5
6 import { cn } from '@lib/utls';
7 import { Button } from '@components/ui/button';
8 import {
9   XIcon,
10  CircleCheckIcon,
11  InfoIcon,
12  TriangleAlertIcon,
13  OctagonXIcon,
14  Loader2Icon,
15 } from 'lucide-react';
16
17 const toast = ToastPrimitive.createToastManager();
18
19 function ToastProvider({ ...props }: ToastPrimitive.Provider.Props) {
20   return <ToastPrimitive.Provider {...props} />;
21 }
22
23 function ToastPortal({ ...props }: ToastPrimitive.Portal.Props) {
24   return <ToastPrimitive.Portal data-slot="toast-portal" {...props} />;
25 }
26
27 function ToastViewport({ className, ...props }: ToastPrimitive.Viewport.Props) {
28   return (
29     <ToastPrimitive.Viewport
30       data-slot="toast-viewport"
31       className={cn(
32         'pointer-events-none fixed inset-x-4 bottom-4 z-50 mx-auto w-auto max-w-sm outline-none
33         sm:right-4 sm:left-auto sm:mx-0 sm:w-full',
34         className,
35       )}
36       {...props}
37     />
38   );
39 }
40
41 function Toast({ className, ...props }: ToastPrimitive.Root.Props) {
42   return (
43     <ToastPrimitive.Root
44       data-slot="toast"
45       className={cn(
46         'rounded-2xl group/toast pointer-events-auto absolute right-0 bottom-0 z-[calc(1000-var(--toast-index))] w-full origin-bottom border bg-popover text-popover-foreground shadow-lg will-change-transform outline-none select-none focus-visible:border-ring focus-visible:ring-[3px] focus-visible:ring-ring/50',
47         '[--gap:0.75rem] [--height:var(--toast-frontmost-height,var(--toast-height))]',
48         '[--offset-y:calc(var(--toast-offset-y)*-1+calc(var(--toast-index)*var(--gap)*-1)+var(--toast-swipe-movement-y))]',
49         '[--peek:0.75rem] [--scale:calc(max(0,1-(var(--toast-index)*0.1)))]',
50         '[--shrink:calc(1-var(--scale))]',
51         'h-[--height] [transform:translateX(var(--toast-swipe-movement-x))_translateY(calc(var(--toast-swipe-movement-y)-(var(--toast-index)*var(--peek))-(var(--shrink)*var(--height))))_scale(var(--scale))]',
52         '[transition:transform_500ms_cubic-bezier(0.22,1,0.36,1),opacity_500ms,height_150ms]',
53         'after:absolute after:top-full after:left-0 after:h-[calc(var(--gap)+1px)] after:w-full after:content-[\'\']',
54         'data-expanded:h-[--toast-height] data-expanded:[transform:translateX(var(--toast-swipe-movement-x))_translateY(var(--offset-y))]',
55         'data-limited:opacity-0 data-starting-style:[transform:translateY(150%)]',
56         '[&[data-ending-style]:not([data-limited]):not([data-swipe-direction])]:[transform:translateY(150%)]',
57         'data-ending-style:data-[swipe-direction=down]:[transform:translateY(calc(var(--toast-swipe-movement-y)+150%))]',
58         'data-ending-style:data-[swipe-direction=left]:[transform:translateX(calc(var(--toast-swipe-movement-x)-150%))_translateY(var(--offset-y))]',
59         'data-ending-style:data-[swipe-direction=right]:[transform:translateX(calc(var(--toast-swipe-movement-x)+150%))_translateY(var(--offset-y))]',
60         'data-ending-style:data-[swipe-direction=up]:[transform:translateY(calc(var(--toast-swipe-movement-y)-150%))]',
61         'data-expanded:data-ending-style:data-[swipe-direction=down]:[transform:translateY(calc(var(--toast-swipe-movement-y)+150%))]',
62         'data-expanded:data-ending-style:data-[swipe-direction=left]:[transform:translateX(calc(var(--toast-swipe-movement-x)-150%))_translateY(var(--offset-y))]',
63         'data-expanded:data-ending-style:data-[swipe-direction=right]:[transform:translateX(calc(var(--toast-swipe-movement-x)+150%))_translateY(var(--offset-y))]',
64         'data-expanded:data-ending-style:data-[swipe-direction=up]:[transform:translateY(calc(var(--toast-swipe-movement-y)-150%))]',
65         className,
66       )}
67     />
68   );
69 }
```

```

62     {...props}
63   />
64   });
65 }
66
67 function ToastContent({ className, ...props }: ToastPrimitive.Content.Props) {
68   return (
69     <ToastPrimitive.Content
70       data-slot="toast-content"
71       className={cn(
72         'flex h-full items-center gap-3 overflow-hidden p-4 transition-opacity duration-250 ease-[cubic-
bezier(0.22,1,0.36,1)] data-behind:opacity-0 data-expanded:opacity-100',
73         className,
74       )}
75       {...props}
76     />
77   );
78 }
79
80 function ToastTitle({ className, ...props }: ToastPrimitive.Title.Props) {
81   return (
82     <ToastPrimitive.Title
83       data-slot="toast-title"
84       className={cn('text-sm font-medium', className)}
85       {...props}
86     />
87   );
88 }
89
90 function ToastDescription({
91   className,
92   ...props
93 }: ToastPrimitive.Description.Props) {
94   return (
95     <ToastPrimitive.Description
96       data-slot="toast-description"
97       className={cn('text-sm text-muted-foreground', className)}
98       {...props}
99     />
100   );
101 }
102
103 function ToastAction({
104   className,
105   render = <Button variant="outline" size="sm" />,
106   ...props
107 }: ToastPrimitive.Action.Props) {
108   return (
109     <ToastPrimitive.Action
110       data-slot="toast-action"
111       render={render}
112       className={cn('shrink-0', className)}
113       {...props}
114     />
115   );
116 }
117
118 function ToastClose({
119   className,
120   children,
121   render = <Button variant="ghost" size="icon-sm" />,
122   ...props
123 }: ToastPrimitive.Close.Props) {
124   return (
125     <ToastPrimitive.Close
126       data-slot="toast-close"
127       aria-label="Close toast"
128       render={render}
129       className={cn(
130         "relative shrink-0 text-muted-foreground after:absolute after:-inset-2 after:content-['']
hover:text-foreground",
131         className,
132       )}
133       {...props}
134     >
135       {children ?? <XIcon aria-hidden="true" />}
136     </ToastPrimitive.Close>
137   );
138 }
139
140 function ToastIcon({ type }: { type: string | undefined }) {
141   let icon: React.ReactNode = null;
142
143   if (type === 'success') {
144     icon = <CircleCheckIcon aria-hidden="true" />;

```

```

145 }
146
147 if (type === 'info') {
148   icon = <InfoIcon aria-hidden="true" />;
149 }
150
151 if (type === 'warning') {
152   icon = <TriangleAlertIcon aria-hidden="true" />;
153 }
154
155 if (type === 'error') {
156   icon = <OctagonXIcon className="text-destructive" aria-hidden="true" />;
157 }
158
159 if (type === 'loading') {
160   icon = <Loader2Icon className="animate-spin" aria-hidden="true" />;
161 }
162
163 if (!icon) {
164   return null;
165 }
166
167 return (
168   <span
169     data-slot="toast-icon"
170     className="shrink-0 [&svg]:pointer-events-none [&svg:not([class*='size-']):size-4"
171     >
172     {icon}
173   </span>
174 );
175 }
176
177 function ToastList() {
178   const { toasts } = ToastPrimitive.useToastManager();
179
180   return toasts.map((toastItem) => (
181     <Toast key={toastItem.id} toast={toastItem}>
182       <ToastContent>
183         <ToastIcon type={toastItem.type} />
184         <div className="flex min-w-0 flex-1 flex-col gap-1">
185           <ToastTitle />
186           <ToastDescription />
187         </div>
188         <ToastAction />
189         <ToastClose />
190       </ToastContent>
191     </Toast>
192   ));
193 }
194
195 function Toaster({
196   children,
197   toastManager = toast,
198   ...props
199 }: ToastPrimitive.Provider.Props) {
200   return (
201     <ToastProvider toastManager={toastManager} {...props}>
202       {children}
203     <ToastPortal>
204       <ToastViewport>
205         <ToastList />
206       </ToastViewport>
207     </ToastPortal>
208   </ToastProvider>
209 );
210 }
211
212 const createToastManager = ToastPrimitive.createToastManager;
213 const useToastManager = ToastPrimitive.useToastManager;
214
215 export {
216   Toaster,
217   Toast,
218   ToastAction,
219   ToastClose,
220   ToastContent,
221   ToastDescription,
222   ToastPortal,
223   ToastProvider,
224   ToastTitle,
225   ToastViewport,
226   createToastManager,
227   toast,
228   useToastManager,
229 };

```

components/ui/toggle-group.tsx

```
1 'use client';
2
3 import * as React from 'react';
4 import { Toggle as TogglePrimitive } from '@base-ui/react/toggle';
5 import { ToggleGroup as ToggleGroupPrimitive } from '@base-ui/react/toggle-group';
6 import { type VariantProps } from 'class-variance-authority';
7
8 import { cn } from '@lib/utils';
9 import { toggleVariants } from '@components/ui/toggle';
10
11 const ToggleGroupContext = React.createContext<
12   VariantProps<typeof toggleVariants> & {
13     spacing?: number;
14     orientation?: 'horizontal' | 'vertical';
15   }
16 >({
17   size: 'default',
18   variant: 'default',
19   spacing: 2,
20   orientation: 'horizontal',
21 });
22
23 function ToggleGroup({
24   className,
25   variant,
26   size,
27   spacing = 2,
28   orientation = 'horizontal',
29   children,
30   ...props
31 }: ToggleGroupPrimitive.Props &
32   VariantProps<typeof toggleVariants> & {
33     spacing?: number;
34     orientation?: 'horizontal' | 'vertical';
35   }) {
36   return (
37     <ToggleGroupPrimitive
38       data-slot="toggle-group"
39       data-variant={variant}
40       data-size={size}
41       data-spacing={spacing}
42       data-orientation={orientation}
43       style={{ '--gap': spacing } as React.CSSProperties}
44       className={cn(
45         'rounded-lg data-[size=sm]:rounded-[min(var(--radius-md),10px)] group/toggle-group flex w-fit
flex-row items-center gap-[--spacing(var(--gap))] data-vertical:flex-col data-vertical:items-stretch',
46         className,
47       )}
48       {...props}
49     >
50       <ToggleGroupContext.Provider
51         value={{ variant, size, spacing, orientation }}
52       >
53         {children}
54       </ToggleGroupContext.Provider>
55     </ToggleGroupPrimitive>
56   );
57 }
58
59 function ToggleGroupItem({
60   className,
61   children,
62   variant = 'default',
63   size = 'default',
64   ...props
65 }: TogglePrimitive.Props & VariantProps<typeof toggleVariants>) {
66   const context = React.useContext(ToggleGroupContext);
67
68   return (
69     <TogglePrimitive
70       data-slot="toggle-group-item"
71       data-variant={context.variant || variant}
72       data-size={context.size || size}
73       data-spacing={context.spacing}
74       className={cn(
75         'group-data-[spacing=0]/toggle-group:rounded-none group-data-[spacing=0]/toggle-group:px-2
group-data-[spacing=0]/toggle-group:has-data-[icon=inline-end]:pr-1.5 group-data-[spacing=0]/toggle-
group:has-data-[icon=inline-start]:pl-1.5 group-data-horizontal/toggle-
group:data-[spacing=0]:first:rounded-l-lg group-data-vertical/toggle-
group:data-[spacing=0]:first:rounded-t-lg group-data-horizontal/toggle-
group:data-[spacing=0]:last:rounded-r-lg group-data-vertical/toggle-group:data-[spacing=0]:last:rounded-
b-lg shrink-0 focus:z-10 focus-visible:z-10 group-data-horizontal/toggle-
```

```

group:data-[spacing=0]:data-[variant=outline]:border-l-0 group-data-vertical/toggle-
group:data-[spacing=0]:data-[variant=outline]:border-t-0 group-data-horizontal/toggle-
group:data-[spacing=0]:data-[variant=outline]:first:border-l group-data-vertical/toggle-
group:data-[spacing=0]:data-[variant=outline]:first:border-t',
76     toggleVariants({
77         variant: context.variant || variant,
78         size: context.size || size,
79     }),
80     className,
81     )}
82     {...props}
83   >
84     {children}
85   </TogglePrimitive>
86 );
87 }
88
89 export { ToggleGroup, ToggleGroupItem };

```

components/ui/toggle.tsx

```
1 'use client';
2
3 import { Toggle as TogglePrimitive } from '@base-ui/react/toggle';
4 import { cva, type VariantProps } from 'class-variance-authority';
5
6 import { cn } from '@lib/utls';
7
8 const toggleVariants = cva(
9   "hover:text-foreground aria-pressed:bg-muted focus-visible:border-ring focus-visible:ring-ring/50
  aria-invalid:ring-destructive/20 dark:aria-invalid:ring-destructive/40 aria-invalid:border-destructive
  data-[state=on]:bg-muted gap-1 rounded-lg text-sm font-medium transition-all
  [&_svg:not([class*='size-'])]:size-4 group/toggle inline-flex items-center justify-center whitespace-
  nowrap outline-none hover:bg-muted focus-visible:ring-[3px] disabled:pointer-events-none
  disabled:opacity-50 [&_svg]:pointer-events-none [&_svg]:shrink-0",
10  {
11    variants: {
12      variant: {
13        default: 'bg-transparent',
14        outline: 'border-input hover:bg-muted border bg-transparent',
15      },
16      size: {
17        default:
18          'h-8 min-w-8 px-2.5 has-data-[icon=inline-end]:pr-2 has-data-[icon=inline-start]:pl-2',
19        sm: "h-7 min-w-7 rounded-[min(var(--radius-md),12px)] px-2.5 text-[0.8rem] has-
  data-[icon=inline-end]:pr-1.5 has-data-[icon=inline-start]:pl-1.5
  [&_svg:not([class*='size-'])]:size-3.5",
20        lg: 'h-9 min-w-9 px-2.5 has-data-[icon=inline-end]:pr-2 has-data-[icon=inline-start]:pl-2',
21      },
22    },
23    defaultVariants: {
24      variant: 'default',
25      size: 'default',
26    },
27  },
28 );
29
30 function Toggle({
31   className,
32   variant = 'default',
33   size = 'default',
34   ...props
35 }: TogglePrimitive.Props & VariantProps<typeof toggleVariants>) {
36   return (
37     <TogglePrimitive
38       data-slot="toggle"
39       className={cn(toggleVariants({ variant, size, className }))}
40       {...props}
41     />
42   );
43 }
44
45 export { Toggle, toggleVariants };
```

components/ui/tooltip.tsx

```
1 'use client';
2
3 import { Tooltip as TooltipPrimitive } from '@base-ui/react/tooltip';
4
5 import { cn } from '@lib/utils';
6
7 function TooltipProvider({
8   delay = 0,
9   ...props
10 }: TooltipPrimitive.Provider.Props) {
11   return (
12     <TooltipPrimitive.Provider
13       data-slot="tooltip-provider"
14       delay={delay}
15       {...props}
16     />
17   );
18 }
19
20 function Tooltip({ ...props }: TooltipPrimitive.Root.Props) {
21   return <TooltipPrimitive.Root data-slot="tooltip" {...props} />;
22 }
23
24 function TooltipTrigger({ ...props }: TooltipPrimitive.Trigger.Props) {
25   return <TooltipPrimitive.Trigger data-slot="tooltip-trigger" {...props} />;
26 }
27
28 function TooltipContent({
29   className,
30   side = 'top',
31   sideOffset = 4,
32   align = 'center',
33   alignOffset = 0,
34   children,
35   ...props
36 }: TooltipPrimitive.Popup.Props &
37   Pick<
38     TooltipPrimitive.Positioner.Props,
39     'align' | 'alignOffset' | 'side' | 'sideOffset'
40   > ) {
41   return (
42     <TooltipPrimitive.Portal>
43       <TooltipPrimitive.Positioner
44         align={align}
45         alignOffset={alignOffset}
46         side={side}
47         sideOffset={sideOffset}
48         className="isolate z-50"
49       >
50         <TooltipPrimitive.Popup
51           data-slot="tooltip-content"
52           className={cn(
53             'z-50 inline-flex w-fit max-w-xs origin(--transform-origin) items-center gap-1.5 rounded-md
54             bg-foreground px-3 py-1.5 text-xs text-background has-data-[slot=kbd]:pr-1.5 data-[side=bottom]:slide-
55             in-from-top-2 data-[side=inline-end]:slide-in-from-left-2 data-[side=inline-start]:slide-in-from-right-2
56             data-[side=left]:slide-in-from-right-2 data-[side=right]:slide-in-from-left-2 data-[side=top]:slide-in-
57             from-bottom-2 **:data-[slot=kbd]:relative **:data-[slot=kbd]:isolate **:data-[slot=kbd]:z-50
58             **:data-[slot=kbd]:rounded-sm data-[state=delayed-open]:animate-in data-[state=delayed-open]:fade-in-0
59             data-[state=delayed-open]:zoom-in-95 data-open:animate-in data-open:fade-in-0 data-open:zoom-in-95 data-
60             closed:animate-out data-closed:fade-out-0 data-closed:zoom-out-95',
61             className,
62           )}
63           {...props}
64         >
65           {children}
66           <TooltipPrimitive.Arrow className="z-50 size-2.5 translate-y-[calc(-50%-2px)] rotate-45
67             rounded-[2px] bg-foreground fill-foreground data-[side=bottom]:top-1 data-[side=inline-end]:top-1/2!
68             data-[side=inline-end]:-left-1 data-[side=inline-end]:-translate-y-1/2 data-[side=inline-start]:top-1/2!
69             data-[side=inline-start]:-right-1 data-[side=inline-start]:-translate-y-1/2 data-[side=left]:top-1/2!
70             data-[side=left]:-right-1 data-[side=left]:-translate-y-1/2 data-[side=right]:top-1/2!
71             data-[side=right]:-left-1 data-[side=right]:-translate-y-1/2 data-[side=top]:-bottom-2.5" />
72         </TooltipPrimitive.Popup>
73       </TooltipPrimitive.Positioner>
74     </TooltipPrimitive.Portal>
75   );
76 }
77
78 export { Tooltip, TooltipTrigger, TooltipContent, TooltipProvider };
```

hooks/use-mobile.ts

```
1 import * as React from 'react';
2
3 const MOBILE_BREAKPOINT = 768;
4
5 export function useIsMobile() {
6   const [isMobile, setIsMobile] = React.useState<boolean | undefined>(
7     undefined,
8   );
9
10  React.useEffect(() => {
11    const mql = window.matchMedia(`(max-width: ${MOBILE_BREAKPOINT - 1}px)`);
12    const onChange = () => {
13      setIsMobile(window.innerWidth < MOBILE_BREAKPOINT);
14    };
15    mql.addEventListener('change', onChange);
16    setIsMobile(window.innerWidth < MOBILE_BREAKPOINT);
17    return () => mql.removeEventListener('change', onChange);
18  }, []);
19
20  return !!isMobile;
21 }
```

index.html

```
1 <!doctype html><html lang="en"><head><meta charset="UTF-8"/><meta name="viewport" content="width=device-  
width, initial-scale=1.0"/><meta name="description" content="Private on-device help finding affordable  
care and preparing for a medical visit. Educational US prototype."/><title>CareRoute – clearer next  
steps</title></head><body><div id="root"></div><script type="module"  
src="/main.tsx"></script></body></html>
```

lib/intent-model.mjs

```
1 // Multinomial naive Bayes trained on hand-written synthetic administrative intents.
2 // Never used for diagnosis, red flags, or medical urgency.
3 const samples = {
4   affordable: ['cannot afford care insurance bill cost expensive', 'uninsured low cost clinic financial
  assistance', 'cheap doctor without insurance money', 'sliding scale fees payment help', 'prescription
  costs budget medication assistance', 'transportation to free clinic', 'hospital bill too expensive',
  'need affordable health center'],
5   prepare: ['prepare appointment questions doctor visit', 'write summary for my appointment', 'what
  should i bring to the visit', 'organize notes before seeing clinician', 'list medications and allergies
  for doctor', 'remember questions at consultation', 'visit preparation checklist', 'explain my concern to
  a nurse'],
6   now: ['need help with a health concern', 'where can i get care today', 'call my clinic for advice',
  'find nearby primary care', 'not sure what to do next', 'need to talk with a nurse', 'looking for health
  information', 'general next steps for care'],
7 };
8 const stop = new Set('a an the to for my i me with of and at in do what should can get need'.split('
  '));
9 const tokens = text => (text.toLowerCase().match(/[a-z]+/g) || []).filter(w => !stop.has(w));
10 const vocabulary = new Set();
11 const classes = Object.entries(samples).map(([mode, phrases]) => {
12   const counts = new Map(); let total = 0;
13   for (const phrase of phrases) for (const word of tokens(phrase))
14     {counts.set(word, (counts.get(word) || 0) + 1); vocabulary.add(word); total++;}
15   return {mode, counts, total};
16 });
17 export function suggestIntent(text) {
18   const words = tokens(text).filter(w => vocabulary.has(w));
19   if (words.length < 2) return null;
20   const scores = classes.map(c => ({mode: c.mode, score: words.reduce((n, w) => n + Math.log(((c.counts.get(w) ||
  0) + 1) / (c.total + vocabulary.size)), 0)})).sort((a, b) => b.score - a.score);
21   if (scores[0].score - scores[1].score < 0.8) return null;
22   return scores[0].mode === 'now' ? null : scores[0].mode;
23 }
```

lib/utils.ts

```
1 import { clsx, type ClassValue } from 'clsx';
2 import { twMerge } from 'tailwind-merge';
3
4 export function cn(...inputs: ClassValue[]) {
5   return twMerge(clsx(inputs));
6 }
```

main.tsx

```
1 import { StrictMode } from 'react';
2 import { createRoot } from 'react-dom/client';
3 import Home from './app/page';
4 import './app/globals.css';
5 createRoot(document.getElementById('root')!).render(<StrictMode><Home /></StrictMode>);
```

package.json

```
1 {
2   "name": "careroute",
3   "version": "0.1.0",
4   "private": true,
5   "engines": {
6     "node": ">=22.13.0"
7   },
8   "scripts": {
9     "dev": "vite --host 0.0.0.0",
10    "build": "tsc --noEmit && vite build",
11    "preview": "vite preview --host 0.0.0.0",
12    "test": "node --test qa/intent-model.test.mjs"
13  },
14  "dependencies": {
15    "react": "19.2.6",
16    "react-dom": "19.2.6",
17    "@base-ui/react": "1.7.0",
18    "@shadcn/react": "0.3.0",
19    "class-variance-authority": "0.7.1",
20    "clsx": "2.1.1",
21    "cmdk": "1.1.1",
22    "date-fns": "4.1.0",
23    "embla-carousel-react": "8.5.2",
24    "input-otp": "1.4.2",
25    "lucide-react": "1.31.0",
26    "react-day-picker": "9.8.1",
27    "react-resizable-panels": "4.5.8",
28    "recharts": "3.8.0",
29    "shadcn": "4.18.0",
30    "tailwind-merge": "3.6.0",
31    "tw-animate-css": "1.4.0"
32  },
33  "devDependencies": {
34    "@tailwindcss/postcss": "4.2.1",
35    "@types/node": "22.19.19",
36    "@types/react": "19.2.14",
37    "@types/react-dom": "19.2.3",
38    "@vitejs/plugin-react": "6.0.2",
39    "oxfmt": "0.61.0",
40    "oxlint": "1.76.0",
41    "oxlint-tsgolint": "7.0.2001",
42    "tailwindcss": "4.2.1",
43    "typescript": "5.9.3",
44    "vite": "8.0.13"
45  },
46  "type": "module"
47 }
```

qa/intent-model.test.mjs

```
1 import { test } from 'node:test';
2 import assert from 'node:assert/strict';
3 import { suggestIntent } from '../lib/intent-model.mjs';
4 test('held-out administrative phrases select useful tools', () => {
5   for (const text of ['I do not have insurance and the clinic bill is expensive','my hospital bill costs
6     too much','I need financial assistance for a prescription'])
7     assert.equal(suggestIntent(text),'affordable');
8   for (const text of ['I want to organize questions for an appointment','help me write notes before my
9     visit','prepare a list of medications for my doctor']) assert.equal(suggestIntent(text),'prepare');
10 });
11 test('empty, unknown, mixed, and clinical-only wording abstains', () => {
12   for (const text of ['', 'hello', 'chest pain', 'insurance visit', '■', 'severe headache'])
13     assert.equal(suggestIntent(text),null);
14 });
```

tsconfig.json

```
1 {
2   "compilerOptions": {
3     "target": "ES2022",
4     "lib": [
5       "dom",
6       "dom.iterable",
7       "esnext"
8     ],
9     "allowJs": true,
10    "skipLibCheck": true,
11    "strict": true,
12    "noEmit": true,
13    "esModuleInterop": true,
14    "module": "esnext",
15    "moduleResolution": "bundler",
16    "types": [
17      "node",
18      "vite/client"
19    ],
20    "resolveJsonModule": true,
21    "isolatedModules": true,
22    "jsx": "react-jsx",
23    "incremental": true,
24    "paths": {
25      "@/*": [
26        "./*"
27      ]
28    }
29  },
30  "include": [
31    "main.tsx",
32    "app/page.tsx",
33    "lib/**/*",
34    "components/ui/**/*",
35    "vite.config.ts"
36  ],
37  "exclude": [
38    "node_modules"
39  ]
40 }
```

vercel.json

```
1 {
2   "framework": "vite",
3   "buildCommand": "npm run build",
4   "outputDirectory": "dist",
5   "headers": [
6     {
7       "source": "/*",
8       "headers": [
9         {
10          "key": "X-Content-Type-Options",
11          "value": "nosniff"
12        },
13        {
14          "key": "Referrer-Policy",
15          "value": "no-referrer"
16        },
17        {
18          "key": "Permissions-Policy",
19          "value": "camera=(), microphone=(), geolocation=()"
20        }
21      ]
22    }
23  ]
24 }
```

vite.config.ts

```
1 import { defineConfig } from 'vite';
2 import react from '@vitejs/plugin-react';
3 import tailwindcss from '@tailwindcss/postcss';
4 import { fileURLToPath, URL } from 'node:url';
5 export default defineConfig({plugins:[react()],css:{postcss:{plugins:[tailwindcss()]}}},resolve:{alias:{'
  @':fileURLToPath(new URL('.',import.meta.url))}}});
```