

=====FILE: src/App.tsx=====

```
1 import AdminLogout from "./admin/AdminLogout";
2 import AdminGuard from "./admin/AdminGuard";
3 import useThemeMode from "./hooks/useThemeMode";
4 import SubNavbar from "./components/SubNavbar";
5
6 import { Routes, Route, useLocation } from "react-router-dom";
7 import Navbar from "./components/Navbar";
8 import ScrollToTop from "./components/ScrollToTop";
9 import Footer from "./components/Footer";
10
11 import Home from "./pages/Home";
12 import Services from "./pages/Services";
13 import Schedule from "./pages/Schedule";
14 import StayLit from "./pages/StayLit";
15 import Contact from "./pages/Contact";
16 import Requests from "./pages/Requests";
17 import ServiceDetail from "./pages/ServiceDetail";
18
19 import Policies from "./pages/Policies";
20 import ServiceArea from "./pages/ServiceArea";
21 import FAQ from "./pages/FAQ";
22 import Reviews from "./pages/Reviews";
23 import Portfolio from "./pages/Portfolio";
24 import About from "./pages/About";
25
26 import CustomerPortal from "./pages/CustomerPortal";
27 import CustomerRequestDetail from "./pages/CustomerRequestDetail";
28
29 import Builds from "./pages/Builds";
30 import BuildDesigner from "./pages/BuildDesigner";
31 import BuildPreview from "./pages/BuildPreview";
32 import BuildPortal from "./pages/BuildPortal";
33
34 import AdminBuilds from "./pages/admin/AdminBuilds";
35 import AdminBuildDetail from "./pages/admin/AdminBuildDetail";
36 import AdminDashboard from "./pages/admin/AdminDashboard";
37 import RequestConfirmed from "./pages/RequestConfirmed";
38
39 import bg from "./assets/handyman-bg.png";
40 import staylitBg from "./assets/staylit-bg.png";
41 import ToastHost from "./components/ToastHost";
42 import MaintenanceDashboardPage from "./pages/admin/MaintenanceDashboard
Page";
43 import AdminRoute from "./components/AdminRoute";
44 import AdminLogin from "./pages/admin/AdminLogin";
45
46 export default function App() {
47   const location = useLocation();
48   const isStayLit = location.pathname.startsWith("/staylit");
49   useThemeMode(isStayLit);
50
51   return (
52     <div
53       style={{
54         minHeight: "100vh",
```

```

55      // Dark neon-friendly overlay so text stays readable over ANY ba
56      ckground image
57      backgroundImage: `
58      linear-gradient(
59          rgba(2,6,23,0.86),
60          rgba(2,6,23,0.82)
61      ),
62      radial-gradient(
63          circle at 20% 10%,
64          rgba(59,130,246,0.18),
65          rgba(0,0,0,0) 55%
66      ),
67      radial-gradient(
68          circle at 80% 15%,
69          rgba(168,85,247,0.14),
70          rgba(0,0,0,0) 60%
71      ),
72      url(${isStayLit ? staylitBg : bg})
73      `,
74      backgroundSize: "cover",
75      backgroundPosition: "center",
76      backgroundRepeat: "no-repeat",
77      backgroundAttachment: "fixed",
78
79      // Global readable text tone (prevents "washed out" text on imag
e backgrounds)
80      color: "#e5e7eb",
81      }
82      >
83      <Navbar />
84      <SubNavbar />
85
86      <ScrollToTop />
87
88      <main
89          key={location.pathname}
90          className="page-enter"
91          style={{
92              padding: "24px 16px",
93              maxWidth: 1200,
94              margin: "0 auto",
95              minHeight: "calc(100vh - 80px)",
96          }}
97      >
98          <Routes>
99
100         <Route path="/admin/logout" element={<AdminGuard><AdminLogout />
</AdminGuard>} />
101         <Route path="/" element={<Home />} />
102             <Route path="/services" element={<Services />} />
103             <Route path="/services/:id" element={<ServiceDetail />} />
104             <Route path="/schedule" element={<Schedule />} />
105             <Route path="/staylit/*" element={<StayLit />} />
106             <Route path="/contact" element={<Contact />} />
107
108             <Route path="/about" element={<About />} />
109             <Route path="/portfolio" element={<Portfolio />} />
110             <Route path="/reviews" element={<Reviews />} />
111             <Route path="/faq" element={<FAQ />} />
112             <Route path="/service-area" element={<ServiceArea />} />
113             <Route path="/policies" element={<Policies />} />

```

```

114      <Route path="/requests" element={<Requests />} />
115
116      <Route path="/customer" element={<CustomerPortal />} />
117      <Route path="/customer/requests/:id" element={<CustomerRequest
Detail />} />
118
119
120      {/* Custom Builds (customer) */}
121      <Route path="/builds" element={<Builds />} />
122      <Route path="/builds/new" element={<BuildDesigner />} />
123      <Route path="/builds/portal" element={<BuildPortal />} />
124      <Route path="/builds/:id" element={<BuildPreview />} />
125  <Route path="/admin/login" element={<AdminLogin />} />
126      <Route
127          path="/admin"
128          element={
129              <AdminGuard>
130                  <AdminDashboard />
131              </AdminGuard>
132          }
133      />
134
135
136      <Route
137          path="/admin/builds"
138          element={
139              <AdminGuard>
140                  <AdminBuilds />
141              </AdminGuard>
142          }
143      />
144      <Route
145          path="/admin/builds/:id"
146          element={
147              <AdminGuard>
148                  <AdminBuildDetail />
149              </AdminGuard>
150          }
151      />
152  {/* IMPORTANT: keep this ABOVE the catch-all 404 route */}
153  <Route path="/request-confirmed/:id" element={<RequestConfirme
d />} />
154
155      <Route
156          path="*"
157          element={
158              <div style={{ padding: 24, maxWidth: 900, margin: "0 auto"
}}>
159                  <h1 style={{ fontWeight: 950, margin: 0 }}>Page not foun
d (404)</h1>
160                  <p style={{ marginTop: 10, fontWeight: 800, opacity: 0.9
}}>
161                      If you expected Admin to load, this can mean the route
is missing or the app is not wrapped in
162                      BrowserRouter.
163                  </p>
164                  <p style={{ marginTop: 10, fontWeight: 900 }}>
165                      Try: <code>/admin/login</code> then <code>/admin</code
>
166                      </p>
167                  </div>
168          }

```

```

169          />
170          <Route path="/admin/maintenance" element={<AdminGuard><AdminRo
171      </Routes>
172      </main>
173
174      <Footer />
175      <ToastHost />
176      </div>
177  );
178 }

```

---

FILE: src/App.css

---

```

1  #root {
2      max-width: 1280px;
3      margin: 0 auto;
4      padding: 2rem;
5      text-align: center;
6  }
7
8  .logo {
9      height: 6em;
10     padding: 1.5em;
11     will-change: filter;
12     transition: filter 300ms;
13 }
14 .logo:hover {
15     filter: drop-shadow(0 0 2em #646cffaa);
16 }
17 .logo.react:hover {
18     filter: drop-shadow(0 0 2em #61dafbaa);
19 }
20
21 @keyframes logo-spin {
22     from {
23         transform: rotate(0deg);
24     }
25     to {
26         transform: rotate(360deg);
27     }
28 }
29
30 @media (prefers-reduced-motion: no-preference) {
31     a:nth-of-type(2) .logo {
32         animation: logo-spin infinite 20s linear;
33     }
34 }
35
36 .card {
37     padding: 2em;
38 }
39
40 .read-the-docs {
41     color: #888;
42 }
43
44 /* --- Splash fade-in animation (Home) --- */
45 .page-enter {
46     animation: pageEnter 520ms ease-out both;
47 }

```

```
48 @keyframes pageEnter {
49   from {
50     opacity: 0;
51     transform: translateY(10px);
52     filter: blur(2px);
53   }
54   to {
55     opacity: 1;
56     transform: translateY(0);
57     filter: blur(0);
58   }
59 }
60 }
61
62 /* Respect accessibility preferences */
63 @media (prefers-reduced-motion: reduce) {
64   .page-enter {
65     animation: none;
66   }
67 }
68
69
70 /* RV: Schedule item styles */
71 .schedList {
72   display: grid;
73   gap: 12px;
74   margin-top: 12px;
75 }
76
77 .schedItem {
78   width: 100%;
79   border-radius: 16px;
80   padding: 14px;
81   border: 1px solid rgba(15, 23, 42, 0.10);
82   background: rgba(255, 255, 255, 0.72);
83   box-shadow: 0 14px 34px rgba(2, 6, 23, 0.10);
84   backdrop-filter: blur(8px);
85 }
86
87 .schedItem:hover {
88   border-color: rgba(99, 102, 241, 0.25);
89   box-shadow: 0 18px 42px rgba(2, 6, 23, 0.14);
90 }
91
92 .schedItemTop {
93   display: flex;
94   justify-content: space-between;
95   align-items: flex-start;
96   gap: 12px;
97   flex-wrap: wrap;
98 }
99
100 .schedItemTitle {
101   display: flex;
102   align-items: center;
103   gap: 10px;
104   flex-wrap: wrap;
105   font-weight: 950;
106   color: #0f172a;
107 }
108
109 .schedItemMeta {
```

```
110     margin-top: 6px;  
111     font-weight: 850;  
112     opacity: 0.85;  
113 }  
114  
115 .schedPills {  
116     display: flex;  
117     gap: 8px;  
118     flex-wrap: wrap;  
119     margin-top: 8px;  
120 }  
121  
122 .schedPill {  
123     display: inline-flex;  
124     align-items: center;  
125     justify-content: center;  
126     padding: 6px 10px;  
127     border-radius: 999px;  
128     font-weight: 950;  
129     font-size: 12px;  
130     border: 1px solid rgba(15, 23, 42, 0.10);  
131     background: rgba(255,255,255,0.75);  
132 }  
133  
134 .schedPillService {  
135     background: rgba(79, 70, 229, 0.10);  
136     border-color: rgba(79, 70, 229, 0.22);  
137     color: #312e81;  
138 }  
139  
140 .schedPillAddOn {  
141     background: rgba(16, 185, 129, 0.10);  
142     border-color: rgba(16, 185, 129, 0.22);  
143     color: #064e3b;  
144 }  
145  
146 .schedEstimate {  
147     display: inline-flex;  
148     align-items: center;  
149     justify-content: center;  
150     padding: 8px 12px;  
151     border-radius: 12px;  
152     font-weight: 950;  
153     border: 1px solid rgba(15, 23, 42, 0.10);  
154     background: rgba(255,255,255,0.80);  
155 }  
156  
157 .schedQuote {  
158     background: rgba(245, 158, 11, 0.14);  
159     border-color: rgba(245, 158, 11, 0.22);  
160     color: #7c2d12;  
161 }  
162  
163 .schedFrom {  
164     background: rgba(14, 165, 233, 0.12);  
165     border-color: rgba(14, 165, 233, 0.22);  
166     color: #0c4a6e;  
167 }  
168  
169 .schedFixed {  
170     background: rgba(34, 197, 94, 0.12);  
171     border-color: rgba(34, 197, 94, 0.20);
```

```
172     color: #14532d;  
173 }
```

=====

FILE: src/index.css

=====

```
1  /* ====== RESET ====== */  
2  
3  *,  
4  *::before,  
5  *::after { box-sizing: border-box; }  
6  
7  html, body, #root {  
8    margin: 0;  
9    padding: 0;  
10   min-height: 100%;  
11   height: 100%;  
12 }  
13  
14 /* ====== THEME TOKENS ====== */  
15 /* Default theme = light (your app can toggle by setting html[data-theme  
="dark"])*/  
16  
17 :root {  
18   /* Text */  
19   --text-main: #050b18;  
20   --text-strong: #020617;  
21   --text-soft: #0f172a;  
22   --text-muted: #475569;  
23  
24   /* Borders + accents */  
25   --border: rgba(15, 23, 42, 0.14);  
26   --accent: #1d4ed8;  
27   --accent2: #60a5fa;  
28  
29   /* Surfaces */  
30   --panel-bg: rgba(255, 255, 255, 0.97);  
31   --field-bg: rgba(255, 255, 255, 0.98);  
32   --badge-bg: rgba(255, 255, 255, 0.96);  
33   --btn-bg: rgba(255, 255, 255, 0.98);  
34  
35   /* Radii + shadows */  
36   --radius-lg: 16px;  
37   --radius-md: 12px;  
38   --shadow-soft: 0 10px 26px rgba(0, 0, 0, 0.10);  
39   --shadow-hover: 0 16px 42px rgba(29, 78, 216, 0.22);  
40 }  
41  
42 /* Dark theme overrides */  
43 html[data-theme="dark"] {  
44   --text-main: #f8fafc;  
45   --text-strong: #ffffff;  
46   --text-soft: rgba(241, 245, 249, 0.92);  
47   --text-muted: rgba(226, 232, 240, 0.82);  
48  
49   --border: rgba(148, 163, 184, 0.22);  
50   --accent: #60a5fa;  
51   --accent2: #22d3ee;  
52  
53   --panel-bg: rgba(2, 6, 23, 0.86);  
54   --field-bg: rgba(2, 6, 23, 0.72);  
55   --badge-bg: rgba(2, 6, 23, 0.72);
```

```
56  --btn-bg:    rgba(2,6,23,0.72);
57
58  --shadow-soft: 0 12px 32px rgba(0,0,0,0.45);
59  --shadow-hover: 0 18px 52px rgba(34,211,238,0.20);
60 }
61
62 /* ===== BASE ===== */
63
64 body {
65   font-family: ui-sans-serif, system-ui, -apple-system, BlinkMacSystemFont,
66   "Segoe UI", Roboto, Inter, Arial;
67   -webkit-font-smoothing: antialiased;
68
69   color: var(--text-main);
70   background-color: #f8fbff;
71 }
72
73 html[data-theme="dark"] body {
74   background-color: #020617;
75 }
76
77 /* ===== TEXT ===== */
78
79 .h1 {
80   font-size: clamp(28px, 3vw, 44px);
81   font-weight: 900;
82   letter-spacing: -0.03em;
83   color: var(--text-strong);
84 }
85
86 .h2 {
87   font-size: clamp(20px, 2vw, 28px);
88   font-weight: 850;
89   color: var(--text-strong);
90 }
91
92 .h3 {
93   font-size: 16px;
94   font-weight: 900;
95   color: var(--text-strong);
96 }
97
98 .h1-accent {
99   background: linear-gradient(90deg, var(--accent), var(--accent2));
100  -webkit-background-clip: text;
101  color: transparent;
102 }
103
104 .lead {
105   font-size: 16px;
106   line-height: 1.6;
107   color: var(--text-soft);
108   font-weight: 600;
109 }
110
111 .body {
112   font-size: 14.5px;
113   line-height: 1.6;
114   color: var(--text-soft);
115   font-weight: 600;
116 }
```

```
117
118 .muted {
119   color: var(--text-muted);
120   font-weight: 700;
121 }
122
123 /* ===== LAYOUT ===== */
124
125 .stack {
126   display: grid;
127   gap: 14px;
128 }
129
130 .row {
131   display: flex;
132   align-items: center;
133   gap: 12px;
134   flex-wrap: wrap;
135 }
136
137 .center {
138   text-align: center;
139   align-items: center;
140   justify-content: center;
141 }
142
143 /* ===== CARDS / PANELS ===== */
144
145 .panel {
146   color: var(--text-main);
147   background: var(--panel-bg);
148   border: 1px solid var(--border);
149   border-radius: var(--radius-lg);
150   box-shadow: var(--shadow-soft);
151   backdrop-filter: blur(16px);
152   transition: transform 0.18s ease, box-shadow 0.18s ease, border-color
0.18s ease;
153 }
154
155 .card { padding: 18px; }
156
157 .card-center {
158   display: flex;
159   flex-direction: column;
160   align-items: center;
161   text-align: center;
162   gap: 8px;
163 }
164
165 /* Glow variant stays (optional class you can add anywhere) */
166 .panel.glow-panel{
167   box-shadow:
168     var(--shadow-soft),
169     0 0 28px rgba(34,211,238,0.45),
170     0 0 72px rgba(59,130,246,0.28),
171     0 0 120px rgba(167,139,250,0.18);
172   border-color: rgba(34,211,238,0.34);
173 }
174 .panel.glow-panel:hover{
175   box-shadow:
176     var(--shadow-soft),
177     0 0 34px rgba(34,211,238,0.55),
```

```
178      0 0 92px rgba(59,130,246,0.34),  
179      0 0 150px rgba(167,139,250,0.22);  
180  transform: translateY(-1px);  
181 }  
182  
183 /* Hover lift for desktop */  
184 @media (hover: hover) and (pointer: fine) {  
185   .panel:hover {  
186     transform: translateY(-2px);  
187     border-color: rgba(96,165,250,0.45);  
188     box-shadow:  
189       0 18px 40px rgba(29,78,216,0.18),  
190       0 0 0 1px rgba(96,165,250,0.18);  
191   }  
192 }  
193  
194 /* Tap feedback for touch devices */  
195 @media (hover: none) and (pointer: coarse) {  
196   .panel:active { transform: scale(0.99); }  
197 }  
198  
199 /* ====== BUTTONS ====== */  
200  
201 .btn {  
202   padding: 10px 14px;  
203   border-radius: var(--radius-md);  
204   border: 1px solid var(--border);  
205   background: var(--btn-bg);  
206   color: var(--text-main);  
207   font-weight: 900;  
208   cursor: pointer;  
209   transition: transform 0.12s ease, box-shadow 0.18s ease, border-color  
0.18s ease, filter 0.18s ease;  
210 }  
211  
212 .btn:hover {  
213   transform: translateY(-1px);  
214   border-color: rgba(96,165,250,0.6);  
215   box-shadow: var(--shadow-hover);  
216 }  
217  
218 .btn:active {  
219   transform: translateY(1px) scale(0.99);  
220   filter: brightness(0.98);  
221 }  
222  
223 .btn-primary {  
224   position: relative;  
225   overflow: hidden;  
226   background: linear-gradient(90deg, rgba(29,78,216,.95), rgba(96,165,25  
0,.95));  
227   color: #fff;  
228   border-color: rgba(29,78,216,.6);  
229 }  
230  
231 /* Subtle animated shine on primary buttons (desktop hover only) */  
232 .btn-primary::after {  
233   content: "";  
234   position: absolute;  
235   top: -40%;  
236   left: -60%;  
237   width: 50%;
```

```
238 height: 180%;  
239 transform: rotate(18deg);  
240 background: linear-gradient(  
241 to right,  
242 rgba(255,255,255,0) 0%,  
243 rgba(255,255,255,0.22) 50%,  
244 rgba(255,255,255,0) 100%  
245 );  
246 transition: left 0.75s ease;  
247 }  
248  
249 @media (hover: hover) and (pointer: fine) {  
250 .btn-primary:hover::after { left: 120%; }  
251 }  
252  
253 .btn-ghost {  
254 background: color-mix(in srgb, var(--btn-bg) 82%, transparent);  
255 }  
256  
257 /* ===== INPUTS ===== */  
258  
259 .field {  
260 width: 100%;  
261 padding: 10px 12px;  
262 border-radius: var(--radius-md);  
263 border: 1px solid var(--border);  
264 background: var(--field-bg);  
265 color: var(--text-main);  
266 font-weight: 600;  
267 }  
268  
269 .field:focus {  
270 outline: none;  
271 border-color: rgba(96,165,250,.7);  
272 box-shadow: 0 0 0 4px rgba(96,165,250,.22);  
273 }  
274  
275 /* ===== BADGE ===== */  
276  
277 .badge {  
278 padding: 5px 12px;  
279 border-radius: 999px;  
280 border: 1px solid var(--border);  
281 background: var(--badge-bg);  
282 font-size: 12px;  
283 font-weight: 900;  
284 color: var(--text-soft);  
285 }  
286  
287 /* ===== PAGE MOTION ===== */  
288  
289 .page {  
290 animation: pageFade 0.35s ease-out both;  
291 }  
292  
293 @keyframes pageFade {  
294 from { opacity: 0; transform: translateY(6px); }  
295 to { opacity: 1; transform: translateY(0); }  
296 }  
297  
298 /* Respect reduced motion */  
299 @media (prefers-reduced-motion: reduce) {
```

```
300 } * { animation: none !important; transition: none !important; }  
301 }
```

=====

FILE: src/styles/theme-override.css

=====

```
1 /* =====  
2  THEME OVERRIDE (loaded LAST)  
3  Purpose: ensure src/index.css theme tokens control the app,  
4  even when urban.css is present.  
5  ===== */  
6  
7 /* 1) Let index.css control base colors instead of urban.css */  
8 html, body {  
9  background: #f8fbff !important;  
10 color: var(--text-main) !important;  
11 }  
12  
13 /* Dark mode base (when html[data-theme="dark"] is set) */  
14 html[data-theme="dark"] body,  
15 html[data-theme="dark"] {  
16  background: #020617 !important;  
17  color: var(--text-main) !important;  
18 }  
19  
20 /* 2) Restore common text classes to theme tokens */  
21 .muted { color: var(--text-muted) !important; }  
22 .body, .lead, p, li, span { color: var(--text-soft) !important; }  
23  
24 /* 3) Restore panels/cards to theme tokens (beats urban.css .panel.card  
background) */  
25 .panel,  
26 .panel.card {  
27  background: var(--panel-bg) !important;  
28  border: 1px solid var(--border) !important;  
29  color: var(--text-main) !important;  
30 }  
31  
32 /* 4) Keep headings consistent with theme tokens */  
33 .h1, .h2, .h3, h1, h2, h3, h4 {  
34  color: var(--text-strong) !important;  
35 }  
36  
37 /* 5) Buttons/fields/badges use theme tokens (avoid urban glow forcing c  
olors) */  
38 .btn {  
39  background: var(--btn-bg) !important;  
40  color: var(--text-main) !important;  
41  border-color: var(--border) !important;  
42 }  
43  
44 .field, input, textarea, select {  
45  background: var(--field-bg) !important;  
46  color: var(--text-main) !important;  
47  border-color: var(--border) !important;  
48 }  
49  
50 .badge {  
51  background: var(--badge-bg) !important;  
52  color: var(--text-soft) !important;  
53  border-color: var(--border) !important;  
54 }
```

```

55  /*
56  =====
57  CLICK FIX: prevent overlays/pseudo-elements from stealing clicks
58  (Common issue with glass cards + .panel::before glow layers)
59  ===== */
60
61  /* Ensure panels establish a stacking context */
62  .panel, .panel.card { position: relative !important; }
63
64  /* Decorative overlays must NOT capture pointer events */
65  .panel::before,
66  .panel::after,
67  .card::before,
68  .card::after,
69  .page::before,
70  .page::after {
71    pointer-events: none !important;
72  }
73
74  /* Force interactive elements above overlays */
75  .panel a,
76  .panel button,
77  .panel .btn,
78  .panel [role="button"],
79  .panel input,
80  .panel select,
81  .panel textarea {
82    position: relative !important;
83    z-index: 2 !important;
84    pointer-events: auto !important;
85 }

```

---

FILE: src/components/Navbar.tsx

---

```

1 import React from "react";
2 import { NavLink, useNavigate } from "react-router-dom";
3 import rvLogo from "../assets/roberts-ventures-logo.png";
4
5 import { useCart } from "../data/requestCart";
6 import { getActiveTheme, toggleTheme } from "../lib/theme";
7
8 const linkStyleBase: React.CSSProperties = {
9   textDecoration: "none",
10  padding: "8px 12px",
11  borderRadius: 999,
12  fontWeight: 850,
13  fontSize: 14,
14  transition: "all .18s ease",
15  display: "inline-flex",
16  alignItems: "center",
17  gap: 8,
18  lineHeight: 1,
19};
20
21 const linkStyleActive: React.CSSProperties = {
22  background: "rgba(99,102,241,0.18)"
23  border: "1px solid rgba(99,102,241,0.25)"
24  boxShadow: "0 10px 26px rgba(2,6,23,0.18)",
25};
26
27 const linkStyleInactive: React.CSSProperties = {

```

```
28     background: "rgba(255,255,255,0.06)"  
29     border: "1px solid rgba(148,163,184,0.16)",  
30 };  
31  
32 function Pill({ children }: { children: React.ReactNode }) {  
33     return (  
34         <span  
35             style={{  
36                 display: "inline-flex",  
37                 alignItems: "center",  
38                 justifyContent: "center",  
39                 padding: "5px 10px",  
40                 borderRadius: 999,  
41                 fontSize: 12,  
42                 fontWeight: 950,  
43                 border: "1px solid rgba(148,163,184,0.18)",  
44                 background: "rgba(255,255,255,0.10)",  
45                 color: "rgba(248,250,252,0.88)",  
46             }}  
47         >  
48             {children}  
49         </span>  
50     );  
51 }  
52  
53 export default function Navbar() {  
54     // Shared NavLink style (kept inside Navbar() so it is always in-scope  
)  
55     const baseLinkStyle: React.CSSProperties = {  
56         textDecoration: "none",  
57         padding: "8px 12px",  
58         borderRadius: 999,  
59         fontWeight: 850,  
60         fontSize: 14,  
61         transition: "all .2s ease",  
62         display: "inline-flex",  
63         alignItems: "center",  
64         gap: 6,  
65     };  
66  
67     const navigate = useNavigate();  
68     const cart = useCart();  
69     const count = (cart as any)?.count ?? 0;  
70  
71     const [theme, setTheme] = React.useState(getActiveTheme());  
72  
73     const onToggleTheme = () => {  
74         toggleTheme();  
75         // setActiveTheme reads from your theme store (html attribute/locals  
76         // storage/etc.)  
77         setTheme(getActiveTheme());  
78     };  
79  
80     return (  
81         <header  
82             style={{  
83                 position: "sticky",  
84                 top: 0,  
85                 zIndex: 50,  
86                 backdropFilter: "blur(18px)",  
87                 background:
```

```
 88          "linear-gradient(90deg, rgba(10,10,15,.78), rgba(6,10,22,.82))  
 89      borderBottom: "1px solid rgba(148,163,184,.15)",  
 90  }  
 91 > <div  
 92     style={ {  
 93       maxWidth: 1200,  
 94       margin: "0 auto",  
 95       padding: "14px 18px",  
 96       display: "flex",  
 97       alignItems: "center",  
 98       justifyContent: "space-between",  
 99       gap: 16,  
100       flexWrap: "wrap",  
101     }  
102   }  
103 > /* Brand */  
104 <button  
105   type="button"  
106   onClick={() => navigate("/")}  
107   style={ {  
108     background: "transparent",  
109     border: "none",  
110     padding: 0,  
111     cursor: "pointer",  
112     display: "flex",  
113     alignItems: "center",  
114     gap: 10,  
115   }  
116   aria-label="Go to Home"  
117 >  
118 <img  
119   src={rvLogo}  
120   alt="Roberts Ventures LLC"  
121   style={ {  
122     width: 32,  
123     height: 32,  
124     objectFit: "contain",  
125     filter: "drop-shadow(0 0 14px rgba(56,189,248,.22))",  
126   }  
127 />  
128 <div style={ { display: "grid", lineHeight: 1.15 } }>  
129   <div  
130     style={ {  
131       fontSize: 18,  
132       fontWeight: 950,  
133       letterSpacing: ".06em",  
134       textTransform: "uppercase",  
135       background: "linear-gradient(90deg,#38bdf8,#a78bfa,#f472  
b6)",  
136       WebkitBackgroundClip: "text",  
137       WebkitTextFillColor: "transparent",  
138       filter: "drop-shadow(0 0 16px rgba(56,189,248,.25))",  
139     }  
140   }  
141 >   Roberts Ventures LLC  
142   </div>  
143   <div  
144     style={ {  
145       fontSize: 11,  
146       color: "rgba(248,250,252,.62)",  
147     }  
148   }  
149 </div>  
150 <div  
151   style={ {  
152     display: "grid",  
153     gridTemplateColumns: "1fr 1fr",  
154     gap: 16,  
155     padding: "14px 18px",  
156   }  
157 > <div  
158   style={ {  
159     display: "flex",  
160     gap: 16,  
161     flexWrap: "wrap",  
162   }  
163 > <div  
164   style={ {  
165     display: "grid",  
166     gridTemplateColumns: "1fr 1fr",  
167     gap: 16,  
168     padding: "14px 18px",  
169   }  
170 > <div  
171   style={ {  
172     display: "flex",  
173     gap: 16,  
174     flexWrap: "wrap",  
175   }  
176 > <div  
177   style={ {  
178     display: "grid",  
179     gridTemplateColumns: "1fr 1fr",  
180     gap: 16,  
181     padding: "14px 18px",  
182   }  
183 > <div  
184   style={ {  
185     display: "flex",  
186     gap: 16,  
187     flexWrap: "wrap",  
188   }  
189 > <div  
190   style={ {  
191     display: "grid",  
192     gridTemplateColumns: "1fr 1fr",  
193     gap: 16,  
194     padding: "14px 18px",  
195   }  
196 > <div  
197   style={ {  
198     display: "flex",  
199     gap: 16,  
200     flexWrap: "wrap",  
201   }  
202 > <div  
203   style={ {  
204     display: "grid",  
205     gridTemplateColumns: "1fr 1fr",  
206     gap: 16,  
207     padding: "14px 18px",  
208   }  
209 > <div  
210   style={ {  
211     display: "flex",  
212     gap: 16,  
213     flexWrap: "wrap",  
214   }  
215 > <div  
216   style={ {  
217     display: "grid",  
218     gridTemplateColumns: "1fr 1fr",  
219     gap: 16,  
220     padding: "14px 18px",  
221   }  
222 > <div  
223   style={ {  
224     display: "flex",  
225     gap: 16,  
226     flexWrap: "wrap",  
227   }  
228 > <div  
229   style={ {  
230     display: "grid",  
231     gridTemplateColumns: "1fr 1fr",  
232     gap: 16,  
233     padding: "14px 18px",  
234   }  
235 > <div  
236   style={ {  
237     display: "flex",  
238     gap: 16,  
239     flexWrap: "wrap",  
240   }  
241 > <div  
242   style={ {  
243     display: "grid",  
244     gridTemplateColumns: "1fr 1fr",  
245     gap: 16,  
246     padding: "14px 18px",  
247   }  
248 > <div  
249   style={ {  
250     display: "flex",  
251     gap: 16,  
252     flexWrap: "wrap",  
253   }  
254 > <div  
255   style={ {  
256     display: "grid",  
257     gridTemplateColumns: "1fr 1fr",  
258     gap: 16,  
259     padding: "14px 18px",  
260   }  
261 > <div  
262   style={ {  
263     display: "flex",  
264     gap: 16,  
265     flexWrap: "wrap",  
266   }  
267 > <div  
268   style={ {  
269     display: "grid",  
270     gridTemplateColumns: "1fr 1fr",  
271     gap: 16,  
272     padding: "14px 18px",  
273   }  
274 > <div  
275   style={ {  
276     display: "flex",  
277     gap: 16,  
278     flexWrap: "wrap",  
279   }  
280 > <div  
281   style={ {  
282     display: "grid",  
283     gridTemplateColumns: "1fr 1fr",  
284     gap: 16,  
285     padding: "14px 18px",  
286   }  
287 > <div  
288   style={ {  
289     display: "flex",  
290     gap: 16,  
291     flexWrap: "wrap",  
292   }  
293 > <div  
294   style={ {  
295     display: "grid",  
296     gridTemplateColumns: "1fr 1fr",  
297     gap: 16,  
298     padding: "14px 18px",  
299   }  
300 > <div  
301   style={ {  
302     display: "flex",  
303     gap: 16,  
304     flexWrap: "wrap",  
305   }  
306 > <div  
307   style={ {  
308     display: "grid",  
309     gridTemplateColumns: "1fr 1fr",  
310     gap: 16,  
311     padding: "14px 18px",  
312   }  
313 > <div  
314   style={ {  
315     display: "flex",  
316     gap: 16,  
317     flexWrap: "wrap",  
318   }  
319 > <div  
320   style={ {  
321     display: "grid",  
322     gridTemplateColumns: "1fr 1fr",  
323     gap: 16,  
324     padding: "14px 18px",  
325   }  
326 > <div  
327   style={ {  
328     display: "flex",  
329     gap: 16,  
330     flexWrap: "wrap",  
331   }  
332 > <div  
333   style={ {  
334     display: "grid",  
335     gridTemplateColumns: "1fr 1fr",  
336     gap: 16,  
337     padding: "14px 18px",  
338   }  
339 > <div  
340   style={ {  
341     display: "flex",  
342     gap: 16,  
343     flexWrap: "wrap",  
344   }  
345 > <div  
346   style={ {  
347     display: "grid",  
348     gridTemplateColumns: "1fr 1fr",  
349     gap: 16,  
350     padding: "14px 18px",  
351   }  
352 > <div  
353   style={ {  
354     display: "flex",  
355     gap: 16,  
356     flexWrap: "wrap",  
357   }  
358 > <div  
359   style={ {  
360     display: "grid",  
361     gridTemplateColumns: "1fr 1fr",  
362     gap: 16,  
363     padding: "14px 18px",  
364   }  
365 > <div  
366   style={ {  
367     display: "flex",  
368     gap: 16,  
369     flexWrap: "wrap",  
370   }  
371 > <div  
372   style={ {  
373     display: "grid",  
374     gridTemplateColumns: "1fr 1fr",  
375     gap: 16,  
376     padding: "14px 18px",  
377   }  
378 > <div  
379   style={ {  
380     display: "flex",  
381     gap: 16,  
382     flexWrap: "wrap",  
383   }  
384 > <div  
385   style={ {  
386     display: "grid",  
387     gridTemplateColumns: "1fr 1fr",  
388     gap: 16,  
389     padding: "14px 18px",  
390   }  
391 > <div  
392   style={ {  
393     display: "flex",  
394     gap: 16,  
395     flexWrap: "wrap",  
396   }  
397 > <div  
398   style={ {  
399     display: "grid",  
400     gridTemplateColumns: "1fr 1fr",  
401     gap: 16,  
402     padding: "14px 18px",  
403   }  
404 > <div  
405   style={ {  
406     display: "flex",  
407     gap: 16,  
408     flexWrap: "wrap",  
409   }  
410 > <div  
411   style={ {  
412     display: "grid",  
413     gridTemplateColumns: "1fr 1fr",  
414     gap: 16,  
415     padding: "14px 18px",  
416   }  
417 > <div  
418   style={ {  
419     display: "flex",  
420     gap: 16,  
421     flexWrap: "wrap",  
422   }  
423 > <div  
424   style={ {  
425     display: "grid",  
426     gridTemplateColumns: "1fr 1fr",  
427     gap: 16,  
428     padding: "14px 18px",  
429   }  
430 > <div  
431   style={ {  
432     display: "flex",  
433     gap: 16,  
434     flexWrap: "wrap",  
435   }  
436 > <div  
437   style={ {  
438     display: "grid",  
439     gridTemplateColumns: "1fr 1fr",  
440     gap: 16,  
441     padding: "14px 18px",  
442   }  
443 > <div  
444   style={ {  
445     display: "flex",  
446     gap: 16,  
447     flexWrap: "wrap",  
448   }  
449 > <div  
450   style={ {  
451     display: "grid",  
452     gridTemplateColumns: "1fr 1fr",  
453     gap: 16,  
454     padding: "14px 18px",  
455   }  
456 > <div  
457   style={ {  
458     display: "flex",  
459     gap: 16,  
460     flexWrap: "wrap",  
461   }  
462 > <div  
463   style={ {  
464     display: "grid",  
465     gridTemplateColumns: "1fr 1fr",  
466     gap: 16,  
467     padding: "14px 18px",  
468   }  
469 > <div  
470   style={ {  
471     display: "flex",  
472     gap: 16,  
473     flexWrap: "wrap",  
474   }  
475 > <div  
476   style={ {  
477     display: "grid",  
478     gridTemplateColumns: "1fr 1fr",  
479     gap: 16,  
480     padding: "14px 18px",  
481   }  
482 > <div  
483   style={ {  
484     display: "flex",  
485     gap: 16,  
486     flexWrap: "wrap",  
487   }  
488 > <div  
489   style={ {  
490     display: "grid",  
491     gridTemplateColumns: "1fr 1fr",  
492     gap: 16,  
493     padding: "14px 18px",  
494   }  
495 > <div  
496   style={ {  
497     display: "flex",  
498     gap: 16,  
499     flexWrap: "wrap",  
500   }  
501 > <div  
502   style={ {  
503     display: "grid",  
504     gridTemplateColumns: "1fr 1fr",  
505     gap: 16,  
506     padding: "14px 18px",  
507   }  
508 > <div  
509   style={ {  
510     display: "flex",  
511     gap: 16,  
512     flexWrap: "wrap",  
513   }  
514 > <div  
515   style={ {  
516     display: "grid",  
517     gridTemplateColumns: "1fr 1fr",  
518     gap: 16,  
519     padding: "14px 18px",  
520   }  
521 > <div  
522   style={ {  
523     display: "flex",  
524     gap: 16,  
525     flexWrap: "wrap",  
526   }  
527 > <div  
528   style={ {  
529     display: "grid",  
530     gridTemplateColumns: "1fr 1fr",  
531     gap: 16,  
532     padding: "14px 18px",  
533   }  
534 > <div  
535   style={ {  
536     display: "flex",  
537     gap: 16,  
538     flexWrap: "wrap",  
539   }  
540 > <div  
541   style={ {  
542     display: "grid",  
543     gridTemplateColumns: "1fr 1fr",  
544     gap: 16,  
545     padding: "14px 18px",  
546   }  
547 > <div  
548   style={ {  
549     display: "flex",  
550     gap: 16,  
551     flexWrap: "wrap",  
552   }  
553 > <div  
554   style={ {  
555     display: "grid",  
556     gridTemplateColumns: "1fr 1fr",  
557     gap: 16,  
558     padding: "14px 18px",  
559   }  
560 > <div  
561   style={ {  
562     display: "flex",  
563     gap: 16,  
564     flexWrap: "wrap",  
565   }  
566 > <div  
567   style={ {  
568     display: "grid",  
569     gridTemplateColumns: "1fr 1fr",  
570     gap: 16,  
571     padding: "14px 18px",  
572   }  
573 > <div  
574   style={ {  
575     display: "flex",  
576     gap: 16,  
577     flexWrap: "wrap",  
578   }  
579 > <div  
580   style={ {  
581     display: "grid",  
582     gridTemplateColumns: "1fr 1fr",  
583     gap: 16,  
584     padding: "14px 18px",  
585   }  
586 > <div  
587   style={ {  
588     display: "flex",  
589     gap: 16,  
590     flexWrap: "wrap",  
591   }  
592 > <div  
593   style={ {  
594     display: "grid",  
595     gridTemplateColumns: "1fr 1fr",  
596     gap: 16,  
597     padding: "14px 18px",  
598   }  
599 > <div  
600   style={ {  
601     display: "flex",  
602     gap: 16,  
603     flexWrap: "wrap",  
604   }  
605 > <div  
606   style={ {  
607     display: "grid",  
608     gridTemplateColumns: "1fr 1fr",  
609     gap: 16,  
610     padding: "14px 18px",  
611   }  
612 > <div  
613   style={ {  
614     display: "flex",  
615     gap: 16,  
616     flexWrap: "wrap",  
617   }  
618 > <div  
619   style={ {  
620     display: "grid",  
621     gridTemplateColumns: "1fr 1fr",  
622     gap: 16,  
623     padding: "14px 18px",  
624   }  
625 > <div  
626   style={ {  
627     display: "flex",  
628     gap: 16,  
629     flexWrap: "wrap",  
630   }  
631 > <div  
632   style={ {  
633     display: "grid",  
634     gridTemplateColumns: "1fr 1fr",  
635     gap: 16,  
636     padding: "14px 18px",  
637   }  
638 > <div  
639   style={ {  
640     display: "flex",  
641     gap: 16,  
642     flexWrap: "wrap",  
643   }  
644 > <div  
645   style={ {  
646     display: "grid",  
647     gridTemplateColumns: "1fr 1fr",  
648     gap: 16,  
649     padding: "14px 18px",  
650   }  
651 > <div  
652   style={ {  
653     display: "flex",  
654     gap: 16,  
655     flexWrap: "wrap",  
656   }  
657 > <div  
658   style={ {  
659     display: "grid",  
660     gridTemplateColumns: "1fr 1fr",  
661     gap: 16,  
662     padding: "14px 18px",  
663   }  
664 > <div  
665   style={ {  
666     display: "flex",  
667     gap: 16,  
668     flexWrap: "wrap",  
669   }  
670 > <div  
671   style={ {  
672     display: "grid",  
673     gridTemplateColumns: "1fr 1fr",  
674     gap: 16,  
675     padding: "14px 18px",  
676   }  
677 > <div  
678   style={ {  
679     display: "flex",  
680     gap: 16,  
681     flexWrap: "wrap",  
682   }  
683 > <div  
684   style={ {  
685     display: "grid",  
686     gridTemplateColumns: "1fr 1fr",  
687     gap: 16,  
688     padding: "14px 18px",  
689   }  
690 > <div  
691   style={ {  
692     display: "flex",  
693     gap: 16,  
694     flexWrap: "wrap",  
695   }  
696 > <div  
697   style={ {  
698     display: "grid",  
699     gridTemplateColumns: "1fr 1fr",  
700     gap: 16,  
701     padding: "14px 18px",  
702   }  
703 > <div  
704   style={ {  
705     display: "flex",  
706     gap: 16,  
707     flexWrap: "wrap",  
708   }  
709 > <div  
710   style={ {  
711     display: "grid",  
712     gridTemplateColumns: "1fr 1fr",  
713     gap: 16,  
714     padding: "14px 18px",  
715   }  
716 > <div  
717   style={ {  
718     display: "flex",  
719     gap: 16,  
720     flexWrap: "wrap",  
721   }  
722 > <div  
723   style={ {  
724     display: "grid",  
725     gridTemplateColumns: "1fr 1fr",  
726     gap: 16,  
727     padding: "14px 18px",  
728   }  
729 > <div  
730   style={ {  
731     display: "flex",  
732     gap: 16,  
733     flexWrap: "wrap",  
734   }  
735 > <div  
736   style={ {  
737     display: "grid",  
738     gridTemplateColumns: "1fr 1fr",  
739     gap: 16,  
740     padding: "14px 18px",  
741   }  
742 > <div  
743   style={ {  
744     display: "flex",  
745     gap: 16,  
746     flexWrap: "wrap",  
747   }  
748 > <div  
749   style={ {  
750     display: "grid",  
751     gridTemplateColumns: "1fr 1fr",  
752     gap: 16,  
753     padding: "14px 18px",  
754   }  
755 > <div  
756   style={ {  
757     display: "flex",  
758     gap: 16,  
759     flexWrap: "wrap",  
760   }  
761 > <div  
762   style={ {  
763     display: "grid",  
764     gridTemplateColumns: "1fr 1fr",  
765     gap: 16,  
766     padding: "14px 18px",  
767   }  
768 > <div  
769   style={ {  
770     display: "flex",  
771     gap: 16,  
772     flexWrap: "wrap",  
773   }  
774 > <div  
775   style={ {  
776     display: "grid",  
777     gridTemplateColumns: "1fr 1fr",  
778     gap: 16,  
779     padding: "14px 18px",  
780   }  
781 > <div  
782   style={ {  
783     display: "flex",  
784     gap: 16,  
785     flexWrap: "wrap",  
786   }  
787 > <div  
788   style={ {  
789     display: "grid",  
790     gridTemplateColumns: "1fr 1fr",  
791     gap: 16,  
792     padding: "14px 18px",  
793   }  
794 > <div  
795   style={ {  
796     display: "flex",  
797     gap: 16,  
798     flexWrap: "wrap",  
799   }  
800 > <div  
801   style={ {  
802     display: "grid",  
803     gridTemplateColumns: "1fr 1fr",  
804     gap: 16,  
805     padding: "14px 18px",  
806   }  
807 > <div  
808   style={ {  
809     display: "flex",  
810     gap: 16,  
811     flexWrap: "wrap",  
812   }  
813 > <div  
814   style={ {  
815     display: "grid",  
816     gridTemplateColumns: "1fr 1fr",  
817     gap: 16,  
818     padding: "14px 18px",  
819   }  
820 > <div  
821   style={ {  
822     display: "flex",  
823     gap: 16,  
824     flexWrap: "wrap",  
825   }  
826 > <div  
827   style={ {  
828     display: "grid",  
829     gridTemplateColumns: "1fr 1fr",  
830     gap: 16,  
831     padding: "14px 18px",  
832   }  
833 > <div  
834   style={ {  
835     display: "flex",  
836     gap: 16,  
837     flexWrap: "wrap",  
838   }  
839 > <div  
840   style={ {  
841     display: "grid",  
842     gridTemplateColumns: "1fr 1fr",  
843     gap: 16,  
844     padding: "14px 18px",  
845   }  
846 > <div  
847   style={ {  
848     display: "flex",  
849     gap: 16,  
850     flexWrap: "wrap",  
851   }  
852 > <div  
853   style={ {  
854     display: "grid",  
855     gridTemplateColumns: "1fr 1fr",  
856     gap: 16,  
857     padding: "14px 18px",  
858   }  
859 > <div  
860   style={ {  
861     display: "flex",  
862     gap: 16,  
863     flexWrap: "wrap",  
864   }  
865 > <div  
866   style={ {  
867     display: "grid",  
868     gridTemplateColumns: "1fr 1fr",  
869     gap: 16,  
870     padding: "14px 18px",  
871   }  
872 > <div  
873   style={ {  
874     display: "flex",  
875     gap: 16,  
876     flexWrap: "wrap",  
877   }  
878 > <div  
879   style={ {  
880     display: "grid",  
881     gridTemplateColumns: "1fr 1fr",  
882     gap: 16,  
883     padding: "14px 18px",  
884   }  
885 > <div  
886   style={ {  
887     display: "flex",  
888     gap: 16,  
889     flexWrap: "wrap",  
890   }  
891 > <div  
892   style={ {  
893     display: "grid",  
894     gridTemplateColumns: "1fr 1fr",  
895     gap: 16,  
896     padding: "14px 18px",  
897   }  
898 > <div  
899   style={ {  
900     display: "flex",  
901     gap: 16,  
902     flexWrap: "wrap",  
903   }  
904 > <div  
905   style={ {  
906     display: "grid",  
907     gridTemplateColumns: "1fr 1fr",  
908     gap: 16,  
909     padding: "14px 18px",  
910   }  
911 > <div  
912   style={ {  
913     display: "flex",  
914     gap: 16,  
915     flexWrap: "wrap",  
916   }  
917 > <div  
918   style={ {  
919     display: "grid",  
920     gridTemplateColumns: "1fr 1fr",  
921     gap: 16,  
922     padding: "14px 18px",  
923   }  
924 > <div  
925   style={ {  
926     display: "flex",  
927     gap: 16,  
928     flexWrap: "wrap",  
929   }  
930 > <div  
931   style={ {  
932     display: "grid",  
933     gridTemplateColumns: "1fr 1fr",  
934     gap: 16,  
935     padding: "14px 18px",  
936   }  
937 > <div  
938   style={ {  
939     display: "flex",  
940     gap: 16,  
941     flexWrap: "wrap",  
942   }  
943 > <div  
944   style={ {  
945     display: "grid",  
946     gridTemplateColumns: "1fr 1fr",  
947     gap: 16,  
948     padding: "14px 18px",  
949   }  
950 > <div  
951   style={ {  
952     display: "flex",  
953     gap: 16,  
954     flexWrap: "wrap",  
955   }  
956 > <div  
957   style={ {  
958     display: "grid",  
959     gridTemplateColumns: "1fr 1fr",  
960     gap: 16,  
961     padding: "14px 18px",  
962   }  
963 > <div  
964   style={ {  
965     display: "flex",  
966     gap: 16,  
967     flexWrap: "wrap",  
968   }  
969 > <div  
970   style={ {  
971     display: "grid",  
972     gridTemplateColumns: "1fr 1fr",  
973     gap: 16,  
974     padding: "14px 18px",  
975   }  
976 > <div  
977   style={ {  
978     display: "flex",  
979     gap: 16,  
980     flexWrap: "wrap",  
981   }  
982 > <div  
983   style={ {  
984     display: "grid",  
985     gridTemplateColumns: "1fr 1fr",  
986     gap: 16,  
987     padding: "14px 18px",  
988   }  
989 > <div  
990   style={ {  
991     display: "flex",  
992     gap: 16,  
993     flexWrap: "wrap",  
994   }  
995 > <div  
996   style={ {  
997     display: "grid",  
998     gridTemplateColumns: "1fr 1fr",  
999     gap: 16,  
1000     padding: "14px 18px",  
1001   }  
1002 > <div  
1003   style={ {  
1004     display: "flex",  
1005     gap: 16,  
1006     flexWrap: "wrap",  
1007   }  
1008 > <div  
1009   style={ {  
1010     display: "grid",  
1011     gridTemplateColumns: "1fr 1fr",  
1012     gap: 16,  
1013     padding: "14px 18px",  
1014   }  
1015 > <div  
1016   style={ {  
1017     display: "flex",  
1018     gap
```

```
148          fontWeight: 800,
149      })
150      > Services • Stay Lit Candle Co.
151      </div>
152      </div>
153  </button>
154
155  {/* Nav */}
156  <nav
157      style={{
158          display: "flex",
159          gap: 8,
160          flexWrap: "wrap",
161          justifyContent: "center",
162          alignItems: "center",
163      }}
164      aria-label="Primary"
165  >
166      <NavLink
167          to="/"
168          end
169          style={({ isActive }) => ({
170              ...linkStyleBase,
171              ...(isActive ? linkStyleActive : linkStyleInactive),
172              color: "rgba(248,250,252,0.92)",
173          })}
174      >
175          Home
176      </NavLink>
177
178      <NavLink
179          to="/services"
180          style={({ isActive }) => ({
181              ...linkStyleBase,
182              ...(isActive ? linkStyleActive : linkStyleInactive),
183              color: "rgba(248,250,252,0.92)",
184          })}
185      >
186          Services
187      </NavLink>
188
189      <NavLink
190          to="/schedule"
191          style={({ isActive }) => ({
192              ...linkStyleBase,
193              ...(isActive ? linkStyleActive : linkStyleInactive),
194              color: "rgba(248,250,252,0.92)",
195          })}
196      >
197          Schedule
198      </NavLink>
199
200
201
202
203      <NavLink
204          to="/staylit"
205          style={({ isActive }) => ({
206              ...linkStyleBase,
207              ...(isActive ? linkStyleActive : linkStyleInactive),
208              color: "rgba(248,250,252,0.92)",
209          })}
210  >
```

```

210      >
211          Stay Lit
212          {count > 0 ? <Pill>{count}</Pill> : null}
213      </NavLink>
214
215      {/* Theme toggle */}
216      <button
217          type="button"
218          onClick={onToggleTheme}
219          style={{
220              padding: "8px 12px",
221              borderRadius: 999,
222              fontWeight: 950,
223              border: "1px solid rgba(148,163,184,0.18)",
224              background: "rgba(255,255,255,0.08)",
225              color: "rgba(248,250,252,0.92)",
226              cursor: "pointer",
227          }}
228          aria-label="Toggle theme"
229          title="Toggle theme"
230      >
231          Theme: {theme === "dark" ? "Dark" : "Light"}
232      </button>
233
234      <NavLink
235          to="/admin/maintenance"
236          style={baseLinkStyle}
237          className={({ isActive }) => `nav-link${isActive ? " active" : ""}`}
238      >
239          Admin
240      </NavLink>
241
242      </nav>
243      </div>
244      </header>
245  );
246 }

```

---

FILE: src/admin/AdminGuard.tsx

---

```

1  import React from "react";
2  import { Navigate, useLocation } from "react-router-dom";
3  import { isAdminAuthed } from "./adminAuth";
4
5  type Props = { children: React.ReactElement };
6
7  export default function AdminGuard({ children }: Props) {
8      const loc = useLocation();
9      if (isAdminAuthed()) return children;
10
11      // Send them to /admin/login and remember where they were going
12      return <Navigate to="/admin/login" replace state={{ from: loc.pathname
+ loc.search }} />;
13  }

```

---

FILE: src/admin/adminAuth.ts

---

```

1  // src/admin/adminAuth.ts
2
3  const KEY = "rv_admin_session_v1";

```

```
4  // --- Lockout (brute-force protection) ---
5  const LOCK_KEY = "rv_admin_lock_v1";
6  const MAX_FAILS = 5;
7  const LOCK_MINUTES = 15;
8
9  type LockState = {
10    fails: number;
11    lockedUntil: number; // epoch ms; 0 means not locked
12  };
13
14  function now(): number {
15    return Date.now();
16  }
17
18  function safeParse<T>(raw: string | null): T | null {
19    if (!raw) return null;
20    try {
21      return JSON.parse(raw) as T;
22    } catch {
23      return null;
24    }
25  }
26
27
28 // =====
29 // Session auth (existing)
30 // =====
31
32  export function isAdminAuthed(): boolean {
33    try {
34      const raw = localStorage.getItem(KEY);
35      if (!raw) return false;
36      const data = JSON.parse(raw);
37      // session expires after 12 hours
38      return Boolean(data?.ok) && typeof data?.exp === "number" && now() <
data.exp;
39    } catch {
40      return false;
41    }
42  }
43
44  export function setAdminAuthed(): void {
45    const exp = now() + 12 * 60 * 60 * 1000;
46    localStorage.setItem(KEY, JSON.stringify({ ok: true, exp }));
47
48    // successful login should reset lock state
49    resetAdminLock();
50  }
51
52  export function clearAdminAuthed(): void {
53    localStorage.removeItem(KEY);
54  }
55
56
57 // =====
58 // Lockout helpers (NEW)
59 // =====
60
61  export function getAdminLock(): LockState {
62    const state = safeParse<LockState>(localStorage.getItem(LOCK_KEY));
63    if (!state) return { fails: 0, lockedUntil: 0 };
64
65    // If lock expired, normalize back to unlocked
66  }
```

```

65  if (state.lockedUntil && now() >= state.lockedUntil) {
66    const cleared = { fails: 0, lockedUntil: 0 };
67    localStorage.setItem(LOCK_KEY, JSON.stringify(cleared));
68    return cleared;
69  }
70
71  // Ensure fields exist
72  return {
73    fails: typeof state.fails === "number" ? state.fails : 0,
74    lockedUntil: typeof state.lockedUntil === "number" ? state.lockedUnt
il : 0,
75  };
76}
77
78 export function isAdminLocked(): boolean {
79  const s = getAdminLock();
80  return Boolean(s.lockedUntil && now() < s.lockedUntil);
81}
82
83 export function getAdminLockRemainingMs(): number {
84  const s = getAdminLock();
85  if (!s.lockedUntil) return 0;
86  return Math.max(0, s.lockedUntil - now());
87}
88
89 export function recordAdminFail(): LockState {
90  const s = getAdminLock();
91  const nextFails = (s.fails || 0) + 1;
92
93  // If you hit max fails, lock for LOCK_MINUTES
94  const lockedUntil =
95    nextFails >= MAX_FAILS ? now() + LOCK_MINUTES * 60 * 1000 : 0;
96
97  const next: LockState = { fails: nextFails, lockedUntil };
98  localStorage.setItem(LOCK_KEY, JSON.stringify(next));
99  return next;
100}
101
102 export function resetAdminLock(): void {
103  localStorage.setItem(LOCK_KEY, JSON.stringify({ fails: 0, lockedUntil:
0 }));
104}

```

---

FILE: src/admin/AdminLoginPage.tsx

---

```

1  import React from "react";
2  import { useLocation, useNavigate } from "react-router-dom";
3  import { setAdminAuthed } from "./adminAuth";
4
5  /**
6   * Admin Login Protection
7   * - Uses a simple passcode stored in an env var.
8   * - This is NOT true security by itself; it's route protection / UX pro
tection.
9   *
10  * Set in: .env (Vite) => VITE_ADMIN_PASSCODE="your-passcode"
11  */
12 export default function AdminLoginPage() {
13  const nav = useNavigate();
14  const loc = useLocation() as any;
15  const from = loc?.state?.from || "/admin";

```

```

16
17  const [code, setCode] = React.useState("");
18  const [err, setErr] = React.useState<string | null>(null);
19
20  const expected = (import.meta as any).env?.VITE_ADMIN_PASSCODE || "";
21
22  function submit(e: React.FormEvent) {
23    e.preventDefault();
24    setErr(null);
25
26    if (!expected) {
27      setErr("Admin passcode is not set. Add VITE_ADMIN_PASSCODE to your
.env file.");
28      return;
29    }
30
31    if (code.trim() !== String(expected).trim()) {
32      setErr("Invalid passcode.");
33      return;
34    }
35
36    setAdminAuthed();
37    nav(from, { replace: true });
38  }
39
40  return (
41    <div className="page" style={{ maxWidth: 520, margin: "0 auto", padding:
ing: "26px 16px" }}>
42      <div className="panel card">
43        <div className="h2" style={{ marginBottom: 8 }}>Admin Login</div
>
44        <div className="muted" style={{ marginBottom: 16 }}>
45          Enter the admin passcode to continue.
46        </div>
47
48        <form onSubmit={submit} className="stack">
49          <input
50            className="field"
51            type="password"
52            value={code}
53            onChange={(e) => setCode(e.target.value)}
54            placeholder="Passcode"
55            autoComplete="current-password"
56          />
57
58          {err ? <div style={{ color: "crimson", fontWeight: 800 }}>{err
59
60          <button className="btn btn-primary" type="submit">Unlock Admin
61        </form>
62      </div>
63    </div>
64  );
65}

```

---

FILE: src/pages/admin/AdminLogin.tsx

---

```

1  import { useEffect, useMemo, useState } from "react";
2  import { useNavigate } from "react-router-dom";
3  import { toast } from "../../lib/toast";

```

```

4 import {
5   setAdminAuthed,
6   isAdminLocked,
7   recordAdminFail,
8   getAdminLockRemainingMs,
9 } from "../../admin/adminAuth";
10
11 /**
12  * Admin PIN Login (with lockout)
13  * PIN source:
14  *   .env.local => VITE_ADMIN_PIN="1234"
15  */
16 export default function AdminLogin() {
17   const nav = useNavigate();
18   const [pin, setPin] = useState("");
19   const [err, setErr] = useState<string>("");
20
21   const correct = (import.meta as any).env?.VITE_ADMIN_PIN ?? "1234";
22
23   const [lockMs, setLockMs] = useState<number>(0);
24
25   // Tick lock timer while locked
26   useEffect(() => {
27     let t: any;
28     const tick = () => setLockMs(getAdminLockRemainingMs());
29     tick();
30     t = setInterval(tick, 1000);
31     return () => clearInterval(t);
32   }, []);
33
34   const locked = useMemo(() => isAdminLocked(), [lockMs]);
35
36   function submit(e: React.FormEvent) {
37     e.preventDefault();
38     setErr("");
39
40     // If locked, block attempts and show remaining time
41     if (locked) {
42       const s = Math.max(1, Math.ceil(lockMs / 1000));
43       const msg = `Locked. Try again in ${s}s.`;
44       setErr(msg);
45       toast(msg, "error", "Admin Locked", 2600);
46       return;
47     }
48
49     const entered = pin.trim();
50     if (!entered) {
51       setErr("Enter your admin PIN.");
52       return;
53     }
54
55     if (entered !== String(correct).trim()) {
56       const res = recordAdminFail(); // updates lock state / remaining f
ails internally
57       const remain = (res as any)?.remainingFails ?? undefined
58       const msg = remain !== undefined
59         ? `Incorrect PIN. ${remain} attempt(s) left.`
60         : "Incorrect PIN.";
61
62       setErr(msg);
63       setPin(""); // ✅ clear input after a bad attempt
64       toast("Incorrect admin PIN.", "error", "Access Denied", 2600);

```

```
65         return;
66     }
67
68     // ✅ Success
69     setAdminAuthed(); // stored session (and resets lock inside adminA
uth)
70     setPin(""); // ✅ clear input
71     setErr("");
72     toast("Welcome back.", "success", "Admin Access", 2200, "Go to Dashb
oard", "/admin");
73     nav("/admin", { replace: true });
74 }
75
76 const lockLabel =
77   locked ? `Locked: ${Math.max(1, Math.ceil(lockMs / 1000))}s remainin
g` : "";
78
79 return (
80   <div className="stack page">
81     <section className="panel card card-center" style={{ maxWidth: 720
, margin: "0 auto" }}>
82       <h1 className="h2">Admin Login</h1>
83       <p className="lead" style={{ maxWidth: 560 }}>
84         Enter your admin PIN to access customer requests and manage st
atuses.
85       </p>
86
87       {locked ? (
88         <div
89           className="panel"
90           style={{
91             padding: 10,
92             borderRadius: 12,
93             border: "1px solid rgba(245,158,11,0.35)",
94             background: "rgba(245,158,11,0.12)",
95             color: "#7c2d12",
96             fontWeight: 900,
97             width: "100%",
98             maxWidth: 420,
99             textAlign: "center",
100           }}
101         >
102           {lockLabel}
103         </div>
104       ) : null}
105
106       <form onSubmit={submit} style={{ width: "100%", maxWidth: 420, d
isplay: "grid", gap: 10 }}>
107         <label style={{ display: "grid", gap: 6 }}>
108           <span className="label">Admin PIN</span>
109           <input
110             className="field"
111             value={pin}
112             onChange={(e) => setPin(e.target.value)}
113             placeholder="••••"
114             inputMode="numeric"
115             autoFocus
116             disabled={locked}
117           />
118         </label>
119
120       {err ? (
```

```

121         <div
122             className="panel"
123             style={{
124                 padding: 10,
125                 borderRadius: 12,
126                 border: "1px solid rgba(220,38,38,0.25)",
127                 background: "rgba(220,38,38,0.08)",
128                 color: "#7f1d1d",
129                 fontWeight: 900,
130             }}
131         >
132             {err}
133         </div>
134     ) : null
135
136     <button className="btn btn-primary" type="submit" disabled={lo
cked}>
137         Unlock Admin
138     </button>
139
140     <button className="btn btn-ghost" type="button" onClick={() =>
nav("/")}>
141         Back to Home
142     </button>
143
144     <div className="muted" style={{ fontWeight: 900, marginTop: 8
}}>
145         PIN is stored in <code>.env.local</code> as <code>VITE_ADMIN
_PIN</code>.
146     </div>
147     </form>
148     </section>
149     </div>
150   );
151 }

```

---

FILE: src/pages/admin/AdminDashboard.tsx

---

```

1  import { useEffect, useMemo, useState } from "react";
2  import { toast } from "../../lib/toast";
3  import { clearAdminAuthed } from "../../../admin/adminAuth";
4
5  type CartItem = { serviceId: string; qty: number; note: string };
6
7  type QuoteLine = {
8    id: string;
9    label: string;
10   qty: number;
11   unitPrice: number; // dollars
12 };
13
14 type QuoteDraft = {
15   lines: QuoteLine[];
16   taxRatePct: number; // 0-100
17   notes: string;
18   createdAt: string;
19   updatedAt: string;
20   sentAt?: string;
21 };
22
23 type RequestRecord = {

```

```

24     createdAt: string;
25     appointmentStart: string;
26     customer: { name: string; phone: string; address?: string; notes?: str
ing };
27     items: CartItem[];
28     photos?: { name: string; type: string; dataUrl: string }[];
29     status?: "new" | "in_progress" | "complete"; // may be missing on olde
r records
30     quote?: QuoteDraft;
31   };
32
33   const KEY = "rv_requests";
34
35   function safeStatus(s: any): "new" | "in_progress" | "complete" {
36     if (s === "in_progress" || s === "complete" || s === "new") return s;
37     return "new";
38   }
39
40   function uid() {
41     return Math.random().toString(16).slice(2) + Date.now().toString(16);
42   }
43
44   function readRequests(): RequestRecord[] {
45     try {
46       const raw = JSON.parse(localStorage.getItem(KEY) ?? "[]");
47       const arr = Array.isArray(raw) ? raw : [];
48       return arr.map(r: any) => ({ ...r, status: safeStatus(r?.status) })
);
49     } catch {
50       return [];
51     }
52   }
53
54   function writeRequests(reqs: RequestRecord[]) {
55     localStorage.setItem(KEY, JSON.stringify(reqs));
56   }
57
58   function fmt(dtIso: string) {
59     const d = new Date(dtIso);
60     if (Number.isNaN(d.getTime())) return dtIso;
61     return d.toLocaleString();
62   }
63
64   function money(n: number) {
65     return new Intl.NumberFormat("en-US", { style: "currency", currency: "USD" }).format(n);
66   }
67
68   function clampNum(x: any, fallback: number) {
69     const n = Number(x);
70     return Number.isFinite(n) ? n : fallback;
71   }
72
73   function badgeStyle(status: string) {
74     if (status === "complete")
75       return { background: "rgba(34,197,94,0.12)", color: "#14532d", border: "1px solid rgba(34,197,94,0.20)" };
76     if (status === "in_progress")
77       return { background: "rgba(14,165,233,0.12)", color: "#0c4a6e", border: "1px solid rgba(14,165,233,0.20)" };
78     return { background: "rgba(245,158,11,0.14)", color: "#7c2d12", border: "1px solid rgba(245,158,11,0.22)" };

```

```

79  }
80
81  export default function AdminDashboard() {
82    const [reqs, setReqs] = useState<RequestRecord[]>([]);
83    const [q, setQ] = useState("");
84    const [statusFilter, setStatusFilter] = useState<"all" | "new" | "in_p
rogress" | "complete">("all");
85
86    // which request has the quote panel open
87    const [openQuoteKey, setOpenQuoteKey] = useState<string | null>(null);
88
89    function refresh() {
90      setReqs(readRequests());
91    }
92
93    useEffect(() => {
94      refresh();
95    }, []);
96
97    const filtered = useMemo(() => {
98      const term = q.trim().toLowerCase();
99      return reqs.filter((r) => {
100        const st = safeStatus(r.status);
101        if (statusFilter !== "all" && st !== statusFilter) return false;
102        if (!term) return true;
103
104        const hay = [
105          r.customer?.name,
106          r.customer?.phone,
107          r.customer?.address,
108          r.customer?.notes,
109          st,
110          r.createdAt,
111          r.appointmentStart,
112          ...(r.items ?? []).map((i) => i.serviceId),
113        ]
114        .filter(Boolean)
115        .join(" ")
116        .toLowerCase();
117
118        return hay.includes(term);
119      });
120    }, [reqs, q, statusFilter]);
121
122    function updateRequestByIndex(index: number, updater: (r: RequestRecor
d) => RequestRecord) {
123      const next = [...reqs];
124      if (!next[index]) return;
125      next[index] = updater(next[index]);
126      setReqs(next);
127      writeRequests(next);
128    }
129
130    function setStatus(index: number, status: "new" | "in_progress" | "com
plete") {
131      updateRequestByIndex(index, (r) => ({ ...r, status }));
132      toast("Status updated.", "success", "Saved", 1600);
133    }
134
135    function clearAll() {
136      if (!confirm("Delete ALL saved requests from this browser? This cann
ot be undone.")) return;

```

```

137     localStorage.setItem(KEY, "[]");
138     refresh();
139     toast("All requests cleared (local only).", "warning", "Cleared", 24
00);
140   }
141
142   function logout() {
143     clearAdminAuthed();
144     toast("Logged out.", "info", "Admin", 1600, "Go Home", "/");
145     window.location.href = "/";
146   }
147
148 // -----
149 // Quote builder functions
150 // -----
151 function ensureQuote(index: number) {
152   updateRequestByIndex(index, (r) => {
153     if (r.quote) return r;
154     const now = new Date().toISOString();
155     const seeded: QuoteLine[] = (r.items ?? []).map((it) => ({
156       id: uid(),
157       label: it.serviceId, // can be swapped to friendly name later
158       qty: clampNum(it.qty, 1),
159       unitPrice: 0,
160     }));
161     return {
162       ...r,
163       quote: {
164         lines: seeded,
165         taxRatePct: 0,
166         notes: "",
167         createdAt: now,
168         updatedAt: now,
169       },
170       status: safeStatus(r.status),
171     };
172   });
173 }
174
175 function setQuoteField(index: number, patch: Partial<QuoteDraft>) {
176   updateRequestByIndex(index, (r) => {
177     const now = new Date().toISOString();
178     const qd: QuoteDraft = r.quote ?? { lines: [], taxRatePct: 0, note
s: "", createdAt: now, updatedAt: now };
179     return { ...r, quote: { ...qd, ...patch, updatedAt: now } };
180   });
181 }
182
183 function setQuoteLine(index: number, lineId: string, patch: Partial<Qu
oteLine>) {
184   updateRequestByIndex(index, (r) => {
185     if (!r.quote) return r;
186     const now = new Date().toISOString();
187     return {
188       ...r,
189       quote: {
190         ...r.quote,
191         updatedAt: now,
192         lines: r.quote.lines.map((ln) => (ln.id === lineId ? { ...ln,
...patch } : ln)),
193       },
194     };

```

```

195     });
196   }
197
198   function addQuoteLine(index: number) {
199     updateRequestByIndex(index, (r) => {
200       const now = new Date().toISOString();
201       const qd = r.quote ?? { lines: [], taxRatePct: 0, notes: "", createdAt: now, updatedAt: now };
202       return {
203         ...r,
204         quote: {
205           ...qd,
206           updatedAt: now,
207           lines: [...qd.lines, { id: uid(), label: "Custom line item", quantity: 1, unitPrice: 0 }],
208         },
209       };
210     });
211   }
212
213   function removeQuoteLine(index: number, lineId: string) {
214     updateRequestByIndex(index, (r) => {
215       if (!r.quote) return r;
216       const now = new Date().toISOString();
217       return { ...r, quote: { ...r.quote, updatedAt: now, lines: r.quote.lines.filter((ln) => ln.id !== lineId) } };
218     });
219   }
220
221   function calcTotals(qd?: QuoteDraft) {
222     const lines = qd?.lines ?? [];
223     const subtotal = lines.reduce((sum, ln) => sum + clampNum(ln.quantity, 0) * clampNum(ln.unitPrice, 0));
224     const taxRate = Math.max(0, Math.min(100, clampNum(qd?.taxRatePct, 0)));
225     const tax = subtotal * (taxRate / 100);
226     const total = subtotal + tax;
227     return { subtotal, taxRate, tax, total };
228   }
229
230   function buildQuoteText(r: RequestRecord) {
231     const qd = r.quote;
232     const { subtotal, taxRate, tax, total } = calcTotals(qd);
233
234     const lines = (qd?.lines ?? []).map((ln) => {
235       const lineTotal = clampNum(ln.quantity, 0) * clampNum(ln.unitPrice, 0);
236       return ` - ${ln.label} - ${ln.quantity} x ${money(ln.unitPrice)} = ${money(lineTotal)}`;
237     });
238
239     const appt = r.appointmentStart ? fmt(r.appointmentStart) : "-";
240
241     return [
242       `Roberts Ventures LLC - Quote`,
243       `Customer: ${r.customer?.name || "-"},`,
244       `Phone: ${r.customer?.phone || "-"},`,
245       r.customer?.address ? `Address: ${r.customer.address}` : null,
246       `Requested Appointment: ${appt},`,
247       `Line Items:`,
248       ... (lines.length ? lines : [` - (No line items yet)`]),
249     ];
250   }

```

```

251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
```
`Subtotal: ${money(subtotal)}` ,
taxRate > 0 ? `Tax (${money(tax)})` : null,
`Total: ${money(total)}` ,
qd?.notes?.trim() ? `Notes: ${qd.notes.trim()}` : null,
`Reply to confirm and we'll get you scheduled.`,
]
.filter(Boolean)
.join("\n");
}
}

async function copyQuote(index: number) {
  const r = reqs[index];
  if (!r?.quote) return toast("Create a quote first.", "warning", "No
Quote", 1800);
  const text = buildQuoteText(r);
  try {
    await navigator.clipboard.writeText(text);
    toast("Quote copied to clipboard.", "success", "Copied", 1800);
  } catch {
    toast("Clipboard blocked. Select the preview text and copy.", "war
ning", "Clipboard", 2400);
  }
}

function openEmail(index: number) {
  const r = reqs[index];
  if (!r?.quote) return toast("Create a quote first.", "warning", "No
Quote", 1800);
  const subject = encodeURIComponent(`Roberts Ventures LLC Quote - ${r
.customer?.name || "Customer"}`);
  const body = encodeURIComponent(buildQuoteText(r));
  window.location.href = `mailto:?subject=${subject}&body=${body}`;
}

function markQuoteSent(index: number) {
  const now = new Date().toISOString();
  updateRequestByIndex(index, (r) => {
    if (!r.quote) return r;
    const st = safeStatus(r.status);
    return {
      ...r,
      status: st === "new" ? "in_progress" : st,
      quote: { ...r.quote, sentAt: now, updatedAt: now },
    };
  });
  toast("Marked quote as sent.", "success", "Saved", 1600);
}

// -----
// Render
// -----
return (
  <div className="stack page">
    <section className="panel card card-center" style={{ maxWidth: 110
0, margin: "0 auto" }}>
      <div className="row" style={{ justifyContent: "space-between", a
lignItems: "center", width: "100%", flexWrap: "wrap", gap: 10 }}>
        <div style={{ display: "grid", gap: 6 }}>
          <h1 className="h2" style={{ margin: 0 }}>Admin Dashboard</h1

```

```

>      307          <div className="muted" style={{ fontWeight: 900 }}>Requests
  stored locally in this browser (Firebase later).</div>
  308      </div>
  309
  310          <div className="row" style={{ gap: 10, flexWrap: "wrap" }}>
  311              <button className="btn btn-ghost" onClick={refresh}>Refresh<
/button>
  312              <button className="btn btn-ghost" onClick={clearAll}>Clear A
  ll</button>
  313          <button className="btn btn-primary admin-quote-btn" style={{ box
  shadow: "0 8px 22px rgba(29,78,216,0.28)", border: "1px solid rgba(29,78,216,0.35)", fontWeight: 950 }} onClick={logout}>Logout</button>
  314      </div>
  315      </div>
  316
  317          <div className="row" style={{ width: "100%", marginTop: 12, gap:
  10, flexWrap: "wrap" }}>
  318              <input
  319                  className="field"
  320                  style={{ flex: 1, minWidth: 240 }}
  321                  value={q}
  322                  onChange={(e) => setQ(e.target.value)}
  323                  placeholder="Search name, phone, service id, notes...">
  324          />
  325
  326          <select
  327              className="field"
  328              style={{ width: 220 }}
  329              value={statusFilter}
  330              onChange={(e) => setStatusFilter(e.target.value as any)}>
  331              >
  332                  <option value="all">All statuses</option>
  333                  <option value="new">New</option>
  334                  <option value="in_progress">In progress</option>
  335                  <option value="complete">Complete</option>
  336          </select>
  337
  338          <span className="badge" style={{ justifyContent: "center" }}>T
  otal: {filtered.length}</span>
  339      </div>
  340      </section>
  341
  342          {filtered.length === 0 ? (
  343              <section className="panel card card-center" style={{ maxWidth: 1
  100, margin: "0 auto" }}>
  344                  <h3 className="h3">No requests found</h3>
  345                  <p className="body" style={{ maxWidth: 720 }}>
  346                      If you submitted a request earlier, it's stored in <code>loc
  alStorage</code> on the same device/browser.
  347                  </p>
  348              </section>
  349          ) : (
  350              <section className="stack" style={{ maxWidth: 1100, margin: "0 a
  uto" }}>
  351                  {filtered.map((r, idx) => {
  352                      const key = (r.createdAt || "x") + ":" + idx;
  353                      const quoteOpen = openQuoteKey === key;
  354                      const st = safeStatus(r.status);
  355                      const totals = calcTotals(r.quote);
  356
  357                      return (

```

```

358 <article key={key} className="panel card" style={{ padding
: 16 }}>
359     <div className="row" style={{ justifyContent: "space-bet
ween", alignItems: "flex-start", gap: 10, flexWrap: "wrap" }}>
360         <div style={{ display: "grid", gap: 6 }}>
361             <div className="h3" style={{ margin: 0 }}>
362                 {r.customer?.name || "Customer"}`}
363                 <span className="badge" style={{ marginLeft: 8, ..
364                     {st.replace("_", " ")).toUpperCase()}
365                 </span>
366             </div>
367
368             <div className="muted" style={{ fontWeight: 900 }}>
369                 Created: {fmt(r.createdAt)} · Appointment: {fmt(r.
appointmentStart)}
370
371
372             <div className="body" style={{ marginTop: 6 }}>
373                 <strong>Phone:</strong> {r.customer?.phone || "-"}
374                 {r.customer?.address ? (
375                     <>
376                         . <strong>Address:</strong> {r.customer.addres
s}
377                     </>
378                 ) : null}
379             </div>
380
381             {r.customer?.notes ? (
382                 <div className="body" style={{ marginTop: 6 }}>
383                     <strong>Notes:</strong> {r.customer.notes}
384                 </div>
385             ) : null}
386             </div>
387
388             <div className="row" style={{ gap: 8, flexWrap: "wrap"
}}>
389                 <button className="btn btn-ghost" onClick={() => set
Status(idx, "new")}>New</button>
390                 <button className="btn btn-ghost" onClick={() => set
Status(idx, "in_progress")}>In Progress</button>
391                 <button className="btn btn-ghost" onClick={() => set
Status(idx, "complete")}>Complete</button>
392
393                 <button
394                     className="btn btn-primary"
395                     onClick={() => {
396                         ensureQuote(idx);
397                         setOpenQuoteKey((prev) => (prev === key ? null :
key));
398                     }}
399                 >
400                     {quoteOpen ? "Close Quote" : "Edit Price / Send Qu
ote"}
401                 </button>
402             </div>
403         </div>
404
405         <div style={{ marginTop: 14 }}>
406             <div className="label">Requested items</div>
407             <div className="body" style={{ marginTop: 8, display:

```

```

"grid", gap: 6 }}>
  408
  409
0, borderRadius: 12 }}>
  410
  411
it.qty}</span>
  412
  413
  414
0, marginTop: 4 }}>
  415
  416
  417
  418
  419
  420
  421
  422
  423
  424
  425
- between", alignItems: "flex-start", gap: 10, flexWrap: "wrap" }}>
  426
  427
Builder</div>
  428
}>
  429
or email the quote.
  430
  431
  432
  433
  434
  435
  436
  437
  438
  439
0 }}>
  440
  441
  442
  443
  444
  445
  446
  447
  448
  449
  {{(r.items ?? []).map((it, j) => (
    <div key={j} className="panel" style={{ padding: 1
      <div style={{ fontWeight: 950 }}>
        {it.serviceId} <span className="muted">· qty {it.qty}</span>
      </div>
      {it.note ? (
        <div className="muted" style={{ fontWeight: 80
          {it.note}
        </div>
      ) : null}
    )})
  </div>
</div>
  {quoteOpen ? (
    <div style={{ marginTop: 16 }} className="panel card">
      <div className="row" style={{ justifyContent: "space
      <div style={{ display: "grid", gap: 6 }}>
        <div className="h3" style={{ margin: 0 }}>Quote
          <div className="muted" style={{ fontWeight: 900
            Edit unit prices, add custom lines, then copy
          </div>
        </div>
      </div>
      <div className="row" style={{ gap: 8, flexWrap: "w
        <span className="badge rate-bright" style={{ jus
        tifyContent: "center" }}>Subtotal: {money(totals.subtotal)}</span>
        <span className="badge rate-bright" style={{ jus
        tifyContent: "center" }}>Total: {money(totals.total)}</span>
      </div>
      </div>
      <div style={{ marginTop: 12, display: "grid", gap: 1
        {{(r.quote?.lines ?? []).map((ln) => (
          <div key={ln.id} className="panel" style={{ padd
          ing: 12, borderRadius: 12, display: "grid", gap: 10 }}>
            <div style={{ display: "grid", gridTemplateCol
            umns: "1fr 110px 140px 90px", gap: 10, alignItems: "center" }}>
              <input className="field" value={ln.label} on
              Change={(e) => setQuoteLine(idx, ln.id, { label: e.target.value })} />
              <input className="field" inputMode="numeric"
              value={ln.qty} onChange={(e) => setQuoteLine(idx, ln.id, { qty: clampNum(e.targ
              et.value, 1) })} />
              <input className="field" inputMode="decimal"
              value={ln.unitPrice} onChange={(e) => setQuoteLine(idx, ln.id, { unitPrice: cla
              mpNum(e.target.value, 0) })} />
            <button className="btn btn-ghost" onClick={(
              ) => removeQuoteLine(idx, ln.id)}>Remove</button>
          </div>
        </div>
        <div className="muted" style={{ fontWeight: 90
  
```

```
0 }>
450   Line total: {money(clampNum(ln.qty, 0) * cla
mpNum(ln.unitPrice, 0))}>
451   </div>
452   </div>
453   )})
454
455   <div className="row" style={{ justifyContent: "cen
ter", gap: 10, flexWrap: "wrap" }}>
456   <button className="btn btn-ghost" onClick={() =>
addQuoteLine(idx)}>+ Add Custom Line</button>
457
458   <div className="row" style={{ gap: 8, alignItems
: "center" }}>
459   <span className="label">Tax %</span>
460   <input
461   className="field"
462   style={{ width: 120 }}
463   inputMode="decimal"
464   value={r.quote?.taxRatePct ?? 0}
465   onChange={(e) => setQuoteField(idx, { taxRat
ePct: Math.max(0, Math.min(100, clampNum(e.target.value, 0))) })}
466   />
467   </div>
468   </div>
469
470   <label style={{ display: "grid", gap: 6 }}>
471   <span className="label">Quote Notes (optional)</
span>
472   <textarea
473   className="field"
474   rows={4}
475   value={r.quote?.notes ?? ""}
476   onChange={(e) => setQuoteField(idx, { notes: e
.target.value })}
477   placeholder="Example: Price valid for 7 days.
Includes standard hardware. Materials not included unless listed above."
478   />
479   </label>
480
481   <div className="row" style={{ justifyContent: "cen
ter", gap: 10, flexWrap: "wrap" }}>
482   <button className="btn btn-primary" onClick={() =>
copyQuote(idx)}>Copy Quote</button>
483   <button className="btn btn-primary" onClick={() =>
openEmail(idx)}>Open Email Draft</button>
484   <button className="btn btn-ghost" onClick={() =>
markQuoteSent(idx)}>Mark Quote Sent</button>
485   </div>
486
487   <div className="panel" style={{ padding: 12, borde
rRadius: 12 }}>
488   <div className="label">Preview</div>
489   <pre style={{ marginTop: 10, whiteSpace: "pre-wr
ap", fontWeight: 800, color: "#0f172a" }}>
490   {buildQuoteText(r)}
491   </pre>
492   </div>
493
494   <div className="muted" style={{ fontWeight: 900 }}>
495   {r.quote?.sentAt ? `Quote marked sent: ${fmt(r.q
```

```

uote.sentAt})` : "Quote not marked as sent yet."}
496                                     </div>
497                                     </div>
498                                     </div>
499                                     ) : null}
500
501                                     {(r.photos?.length ?? 0) > 0 ? (
502   <div style={{ marginTop: 14 }}>
503   <div className="label">Photos</div>
504   <div style={{ marginTop: 8, display: "grid", gridTem
plateColumns: "repeat(auto-fit, minmax(140px, 1fr))", gap: 10 }}>
505   {r.photos!.map((p, k) => (
506   <div key={k} className="panel" style={{ padding:
10, borderRadius: 12 }}>
507   <img
508   src={p.dataUrl}
509   alt={p.name}
510   style={{
511   width: "100%",
512   height: 120,
513   objectFit: "cover",
514   borderRadius: 10,
515   border: "1px solid rgba(15,23,42,0.15)",
516   }}
517   />
518   <div className="muted" style={{ fontSize: 11,
fontWeight: 900, marginTop: 6, wordBreak: "break-word" }}>
519   {p.name}
520   </div>
521   </div>
522   ))}
523   </div>
524   </div>
525   ) : null}
526   </article>
527   );
528   );
529   </section>
530   );
531   </div>
532   );
533   }

```

---

FILE: src/pages/admin/AdminBuilds.tsx

---

```

1  import { useEffect, useMemo, useState } from "react";
2  import { Link } from "react-router-dom";
3  import { readBuilds, type BuildSubmission } from "../../lib/buildsStore"
;
4  import { toast } from "../../lib/toast";
5
6  function safe(s: any) {
7    return String(s || "").toLowerCase();
8  }
9
10 function fmt(iso: string) {
11  const d = new Date(iso);
12  if (Number.isNaN(d.getTime())) return iso;
13  return d.toLocaleString();
14}
15

```

```

16  export default function AdminBuilds() {
17    const [all, setAll] = useState<BuildSubmission[]>([]);
18    const [q, setQ] = useState("");
19
20    function refresh() {
21      setAll(readBuilds());
22    }
23
24    useEffect(() => refresh(), []);
25
26    const filtered = useMemo(() => {
27      const term = q.trim().toLowerCase();
28      if (!term) return all;
29      return all.filter((b) => {
30        const hay = [
31          b.id,
32          b.status,
33          b.createdAt,
34          b.updatedAt,
35          b.customer?.name,
36          b.customer?.phone,
37          b.customer?.email,
38          b.project?.type,
39          b.project?.notes,
40        ]
41        .filter(Boolean)
42        .join(" ")
43        .toLowerCase();
44        return hay.includes(term);
45      });
46    }, [all, q]);
47
48    function clearAll() {
49      if (!confirm("Delete ALL build submissions stored in this browser? T
his cannot be undone.")) return;
50      localStorage.setItem("rv_build_submissions", "[]");
51      refresh();
52      toast("All build submissions cleared (local only).", "warning", "Cle
ared", 2400);
53    }
54
55    return (
56      <div className="stack page" style={{ gap: 16 }}>
57        <section className="panel card card-center" style={{ maxWidth: 110
0, margin: "0 auto", padding: 18 }}>
58          <div className="row" style={{ justifyContent: "space-between", a
lignItems: "center", gap: 10, flexWrap: "wrap" }}>
59            <div style={{ display: "grid", gap: 6 }}>
60              <h1 className="h2" style={{ margin: 0 }}>Admin – Build Submi
ssions</h1>
61              <div className="muted" style={{ fontWeight: 900 }}>
62                Stored locally in this browser (Firebase later). Customers
see renders + public estimate only.
63              </div>
64            </div>
65
66            <div className="row" style={{ gap: 10, flexWrap: "wrap" }}>
67              <button className="btn btn-ghost" onClick={refresh}>Refresh<
/button>
68              <button className="btn btn-ghost" onClick={clearAll}>Clear A
ll</button>
69              <Link className="btn btn-primary" to="/admin">Back to Admin

```

```

Dashboard</Link>
  70      </div>
  71      </div>
  72
  73      <div className="row" style={{ width: "100%", marginTop: 12, gap:
  74        10, flexWrap: "wrap" }}>
  75        <input className="field" style={{ flex: 1, minWidth: 240 }} va
lue={q} onChange={(e) => setQ(e.target.value)} placeholder="Search name, phone,
  76        email, status, type..." />
  77        <span className="badge" style={{ justifyContent: "center" }}>T
otal: {filtered.length}</span>
  78      </div>
  79      </section>
  80
  81      {filtered.length === 0 ? (
  82        <section className="panel card card-center" style={{ maxWidth: 1
100, margin: "0 auto" }}>
  83          <h3 className="h3">No build submissions found</h3>
  84          <p className="body" style={{ maxWidth: 780 }}>
  85            Builds are stored in <code>localStorage</code> on the same d
evice/browser. Once Firebase is connected, these will sync.
  86          </p>
  87          <Link className="btn btn-ghost" to="/builds/new">Create a test
  88        build</Link>
  89      ) : (
  90        <section className="stack" style={{ maxWidth: 1100, margin: "0 a
uto" }}>
  91          <filtered
  92            .slice()
  93            .sort((a, b) => safe(b.createdAt).localeCompare(safe(a.creat
edAt)))
  94            .map((b) => (
  95              <article key={b.id} className="panel card" style={{ paddin
g: 16 }}>
  96                <div className="row" style={{ justifyContent: "space-bet
ween", alignItems: "flex-start", gap: 10, flexWrap: "wrap" }}>
  97                  <div style={{ display: "grid", gap: 6 }}>
  98                    <div style={{ fontWeight: 950, color: "#0f172a" }}>
  99                      Build <span className="badge" style={{ marginLeft:
8, fontSize: 12 }}>{String(b.id).slice(-8).toUpperCase()}</span>
  100                     <span className="badge" style={{ marginLeft: 8 }}>
  101                      {String(b.status || "draft").toUpperCase()}</span>
  102                    </div>
  103
  104
  105                    <div className="muted" style={{ fontWeight: 850 }}>
  106                      Created: {fmt(b.createdAt)} • Updated: {fmt(b.upda
tedAt)}
  107                    </div>
  108
  109
  110                    <div className="muted" style={{ fontWeight: 850 }}>
  111                      Customer: <strong>{b.customer?.name || "-"</strong
> • {b.customer?.phone || "-"} • {b.customer?.email || "-"}</div>
  112
  113                      <div className="muted" style={{ fontWeight: 850 }}>
  114                        Type: {b.project?.type || "-"} • Dims: {b.project?
.dims?.lengthIn ?? "-"}x{b.project?.dims?.widthIn ?? "-"}x{b.project?.dims?.heig
htIn ?? "-"} in
  115                        </div>
  116                      </div>
  117                    </div>
  118
  119
  120
  121
  122
  123
  124
  125
  126
  127
  128
  129
  130
  131
  132
  133
  134
  135
  136
  137
  138
  139
  140
  141
  142
  143
  144
  145
  146
  147
  148
  149
  150
  151
  152
  153
  154
  155
  156
  157
  158
  159
  160
  161
  162
  163
  164
  165
  166
  167
  168
  169
  170
  171
  172
  173
  174
  175
  176
  177
  178
  179
  180
  181
  182
  183
  184
  185
  186
  187
  188
  189
  190
  191
  192
  193
  194
  195
  196
  197
  198
  199
  200
  201
  202
  203
  204
  205
  206
  207
  208
  209
  210
  211
  212
  213
  214
  215
  216
  217
  218
  219
  220
  221
  222
  223
  224
  225
  226
  227
  228
  229
  230
  231
  232
  233
  234
  235
  236
  237
  238
  239
  240
  241
  242
  243
  244
  245
  246
  247
  248
  249
  250
  251
  252
  253
  254
  255
  256
  257
  258
  259
  260
  261
  262
  263
  264
  265
  266
  267
  268
  269
  270
  271
  272
  273
  274
  275
  276
  277
  278
  279
  280
  281
  282
  283
  284
  285
  286
  287
  288
  289
  290
  291
  292
  293
  294
  295
  296
  297
  298
  299
  300
  301
  302
  303
  304
  305
  306
  307
  308
  309
  310
  311
  312
  313
  314
  315
  316
  317
  318
  319
  320
  321
  322
  323
  324
  325
  326
  327
  328
  329
  330
  331
  332
  333
  334
  335
  336
  337
  338
  339
  340
  341
  342
  343
  344
  345
  346
  347
  348
  349
  350
  351
  352
  353
  354
  355
  356
  357
  358
  359
  360
  361
  362
  363
  364
  365
  366
  367
  368
  369
  370
  371
  372
  373
  374
  375
  376
  377
  378
  379
  380
  381
  382
  383
  384
  385
  386
  387
  388
  389
  390
  391
  392
  393
  394
  395
  396
  397
  398
  399
  400
  401
  402
  403
  404
  405
  406
  407
  408
  409
  410
  411
  412
  413
  414
  415
  416
  417
  418
  419
  420
  421
  422
  423
  424
  425
  426
  427
  428
  429
  430
  431
  432
  433
  434
  435
  436
  437
  438
  439
  440
  441
  442
  443
  444
  445
  446
  447
  448
  449
  450
  451
  452
  453
  454
  455
  456
  457
  458
  459
  460
  461
  462
  463
  464
  465
  466
  467
  468
  469
  470
  471
  472
  473
  474
  475
  476
  477
  478
  479
  480
  481
  482
  483
  484
  485
  486
  487
  488
  489
  490
  491
  492
  493
  494
  495
  496
  497
  498
  499
  500
  501
  502
  503
  504
  505
  506
  507
  508
  509
  510
  511
  512
  513
  514
  515
  516
  517
  518
  519
  520
  521
  522
  523
  524
  525
  526
  527
  528
  529
  530
  531
  532
  533
  534
  535
  536
  537
  538
  539
  540
  541
  542
  543
  544
  545
  546
  547
  548
  549
  550
  551
  552
  553
  554
  555
  556
  557
  558
  559
  560
  561
  562
  563
  564
  565
  566
  567
  568
  569
  570
  571
  572
  573
  574
  575
  576
  577
  578
  579
  580
  581
  582
  583
  584
  585
  586
  587
  588
  589
  590
  591
  592
  593
  594
  595
  596
  597
  598
  599
  600
  601
  602
  603
  604
  605
  606
  607
  608
  609
  610
  611
  612
  613
  614
  615
  616
  617
  618
  619
  620
  621
  622
  623
  624
  625
  626
  627
  628
  629
  630
  631
  632
  633
  634
  635
  636
  637
  638
  639
  640
  641
  642
  643
  644
  645
  646
  647
  648
  649
  650
  651
  652
  653
  654
  655
  656
  657
  658
  659
  660
  661
  662
  663
  664
  665
  666
  667
  668
  669
  670
  671
  672
  673
  674
  675
  676
  677
  678
  679
  680
  681
  682
  683
  684
  685
  686
  687
  688
  689
  690
  691
  692
  693
  694
  695
  696
  697
  698
  699
  700
  701
  702
  703
  704
  705
  706
  707
  708
  709
  710
  711
  712
  713
  714
  715
  716
  717
  718
  719
  720
  721
  722
  723
  724
  725
  726
  727
  728
  729
  730
  731
  732
  733
  734
  735
  736
  737
  738
  739
  740
  741
  742
  743
  744
  745
  746
  747
  748
  749
  750
  751
  752
  753
  754
  755
  756
  757
  758
  759
  760
  761
  762
  763
  764
  765
  766
  767
  768
  769
  770
  771
  772
  773
  774
  775
  776
  777
  778
  779
  779
  780
  781
  782
  783
  784
  785
  786
  787
  788
  789
  790
  791
  792
  793
  794
  795
  796
  797
  798
  799
  800
  801
  802
  803
  804
  805
  806
  807
  808
  809
  809
  810
  811
  812
  813
  814
  815
  816
  817
  818
  819
  819
  820
  821
  822
  823
  824
  825
  826
  827
  828
  829
  829
  830
  831
  832
  833
  834
  835
  836
  837
  838
  839
  839
  840
  841
  842
  843
  844
  845
  846
  847
  848
  849
  849
  850
  851
  852
  853
  854
  855
  856
  857
  858
  859
  859
  860
  861
  862
  863
  864
  865
  866
  867
  868
  869
  869
  870
  871
  872
  873
  874
  875
  876
  877
  878
  879
  879
  880
  881
  882
  883
  884
  885
  886
  887
  888
  889
  889
  890
  891
  892
  893
  894
  895
  896
  897
  898
  899
  899
  900
  901
  902
  903
  904
  905
  906
  907
  908
  909
  909
  910
  911
  912
  913
  914
  915
  916
  917
  918
  919
  919
  920
  921
  922
  923
  924
  925
  926
  927
  928
  929
  929
  930
  931
  932
  933
  934
  935
  936
  937
  938
  939
  939
  940
  941
  942
  943
  944
  945
  946
  947
  948
  949
  949
  950
  951
  952
  953
  954
  955
  956
  957
  958
  959
  959
  960
  961
  962
  963
  964
  965
  966
  967
  968
  969
  969
  970
  971
  972
  973
  974
  975
  976
  977
  978
  979
  979
  980
  981
  982
  983
  984
  985
  986
  987
  988
  989
  989
  990
  991
  992
  993
  994
  995
  996
  997
  998
  999
  999
  1000
  1000
  1001
  1002
  1003
  1004
  1005
  1006
  1007
  1008
  1009
  1009
  1010
  1011
  1012
  1013
  1014
  1015
  1016
  1017
  1018
  1019
  1019
  1020
  1021
  1022
  1023
  1024
  1025
  1026
  1027
  1028
  1029
  1029
  1030
  1031
  1032
  1033
  1034
  1035
  1036
  1037
  1038
  1039
  1039
  1040
  1041
  1042
  1043
  1044
  1045
  1046
  1047
  1048
  1049
  1049
  1050
  1051
  1052
  1053
  1054
  1055
  1056
  1057
  1058
  1059
  1059
  1060
  1061
  1062
  1063
  1064
  1065
  1066
  1067
  1068
  1069
  1069
  1070
  1071
  1072
  1073
  1074
  1075
  1076
  1077
  1078
  1079
  1079
  1080
  1081
  1082
  1083
  1084
  1085
  1086
  1087
  1088
  1089
  1089
  1090
  1091
  1092
  1093
  1094
  1095
  1096
  1097
  1098
  1099
  1099
  1100
  1101
  1102
  1103
  1104
  1105
  1106
  1107
  1108
  1109
  1109
  1110
  1111
  1112
  1113
  1114
  1115
  1116
  1117
  1118
  1119
  1119
  1120
  1121
  1122
  1123
  1124
  1125
  1126
  1127
  1128
  1129
  1129
  1130
  1131
  1132
  1133
  1134
  1135
  1136
  1137
  1138
  1139
  1139
  1140
  1141
  1142
  1143
  1144
  1145
  1146
  1147
  1148
  1149
  1149
  1150
  1151
  1152
  1153
  1154
  1155
  1156
  1157
  1158
  1159
  1159
  1160
  1161
  1162
  1163
  1164
  1165
  1166
  1167
  1168
  1169
  1169
  1170
  1171
  1172
  1173
  1174
  1175
  1176
  1177
  1178
  1178
  1179
  1180
  1181
  1182
  1183
  1184
  1185
  1186
  1187
  1188
  1188
  1189
  1190
  1191
  1192
  1193
  1194
  1195
  1196
  1197
  1198
  1198
  1199
  1199
  1200
  1201
  1202
  1203
  1204
  1205
  1206
  1207
  1208
  1209
  1209
  1210
  1211
  1212
  1213
  1214
  1215
  1216
  1217
  1218
  1219
  1219
  1220
  1221
  1222
  1223
  1224
  1225
  1226
  1227
  1228
  1229
  1229
  1230
  1231
  1232
  1233
  1234
  1235
  1236
  1237
  1238
  1239
  1239
  1240
  1241
  1242
  1243
  1244
  1245
  1246
  1247
  1248
  1249
  1249
  1250
  1251
  1252
  1253
  1254
  1255
  1256
  1257
  1258
  1259
  1259
  1260
  1261
  1262
  1263
  1264
  1265
  1266
  1267
  1268
  1269
  1269
  1270
  1271
  1272
  1273
  1274
  1275
  1276
  1277
  1278
  1278
  1279
  1280
  1281
  1282
  1283
  1284
  1285
  1286
  1287
  1288
  1288
  1289
  1290
  1291
  1292
  1293
  1294
  1295
  1296
  1297
  1297
  1298
  1299
  1299
  1300
  1301
  1302
  1303
  1304
  1305
  1306
  1307
  1308
  1309
  1309
  1310
  1311
  1312
  1313
  1314
  1315
  1316
  1317
  1318
  1319
  1319
  1320
  1321
  1322
  1323
  1324
  1325
  1326
  1327
  1328
  1329
  1329
  1330
  1331
  1332
  1333
  1334
  1335
  1336
  1337
  1338
  1339
  1339
  1340
  1341
  1342
  1343
  1344
  1345
  1346
  1347
  1348
  1349
  1349
  1350
  1351
  1352
  1353
  1354
  1355
  1356
  1357
  1358
  1359
  1359
  1360
  1361
  1362
  1363
  1364
  1365
  1366
  1367
  1368
  1369
  1369
  1370
  1371
  1372
  1373
  1374
  1375
  1376
  1377
  1378
  1378
  1379
  1380
  1381
  1382
  1383
  1384
  1385
  1386
  1387
  1388
  1388
  1389
  1390
  1391
  1392
  1393
  1394
  1395
  1396
  1397
  1398
  1398
  1399
  1400
  1401
  1402
  1403
  1404
  1405
  1406
  1407
  1408
  1409
  1409
  1410
  1411
  1412
  1413
  1414
  1415
  1416
  1417
  1418
  1419
  1419
  1420
  1421
  1422
  1423
  1424
  1425
  1426
  1427
  1428
  1429
  1429
  1430
  1431
  1432
  1433
  1434
  1435
  1436
  1437
  1438
  1438
  1439
  1440
  1441
  1442
  1443
  1444
  1445
  1446
  1447
  1448
  1449
  1449
  1450
  1451
  1452
  1453
  1454
  1455
  1456
  1457
  1458
  1459
  1459
  1460
  1461
  1462
  1463
  1464
  1465
  1466
  1467
  1468
  1469
  1469
  1470
  1471
  1472
  1473
  1474
  1475
  1476
  1477
  1478
  1478
  1479
  1480
  1481
  1482
  1483
  1484
  1485
  1486
  1487
  1488
  1488
  1489
  1490
  1491
  1492
  1493
  1494
  1495
  1496
  1497
  1498
  1498
  1499
  1500
  1501
  1502
  1503
  1504
  1505
  1506
  1507
  1508
  1509
  1509
  1510
  1511
  1512
  1513
  1514
  1515
  1516
  1517
  1518
  1519
  1519
  1520
  1521
  1522
  1523
  1524
  1525
  1526
  1527
  1528
  1529
  1529
  1530
  1531
  1532
  1533
  1534
  1535
  1536
  1537
  1538
  1539
  1539
  1540
  1541
  1542
  1543
  1544
  1545
  1546
  1547
  1548
  1549
  1549
  1550
  1551
  1552
  1553
  1554
  1555
  1556
  1557
  1558
  1559
  1559
  1560
  1561
  1562
  1563
  1564
  1565
  1566
  1567
  1568
  1569
  1569
  1570
  1571
  1572
  1573
  1574
  1575
  1576
  1577
  1578
  1578
  1579
  1580
  1581
  1582
  1583
  1584
  1585
  1586
  1587
  1588
  1588
  1589
  1590
  1591
  1592
  1593
  1594
  1595
  1596
  1597
  1598
  1598
  1599
  1600
  1601
  1602
  1603
  1604
  1605
  1606
  1607
  1608
  1609
  1609
  1610
  1611
  1612
  1613
  1614
  1615
  1616
  1617
  1618
  1619
  1619
  1620
  1621
  1622
  1623
  1624
  1625
  1626
  1627
  1628
  1629
  1629
  1630
  1631
  1632
  1633
  1
```

```

114             <Link className="btn btn-primary" to={`/admin/builds/${b.id}`} style={{ fontWeight: 950 }}>
115                 View Details →
116             </Link>
117         </div>
118     </article>
119     )})
120 </section>
121 )
122 </div>
123 );
124

```

=====

FILE: src/pages/admin/AdminBuildDetail.tsx

=====

```

1 import { useEffect, useMemo, useState } from "react";
2 import { Link, useParams } from "react-router-dom";
3 import { getBuild, upsertBuild, removeLastCustomerNote, compileNotes, ty
pe BuildSubmission } from "../../lib/buildsStore";
4 import { estimateBuild } from "../../lib/buildPricing";
5 import { toast } from "../../lib/toast";
6
7 function fmt(iso: string) {
8     const d = new Date(iso);
9     if (Number.isNaN(d.getTime())) return iso;
10    return d.toLocaleString();
11}
12
13 function money(n: number) {
14    return new Intl.NumberFormat("en-US", { style: "currency", currency: "U
SD" }).format(n);
15}
16
17 export default function AdminBuildDetail() {
18    const { id } = useParams();
19    const [build, setBuild] = useState<BuildSubmission | null>(null);
20
21    useEffect(() => {
22        if (!id) return;
23        setBuild(getBuild(id));
24    }, [id]);
25
26    const v = useMemo(() => build?.versions?.[0] ?? null, [build]);
27
28    if (!build || !v) {
29        return (
30            <div className="panel card card-center" style={{ maxWidth: 980, ma
rgin: "0 auto" }}>
31                <h3 className="h3">Build not found</h3>
32                <Link className="btn btn-primary" to="/admin/builds">Back</Link>
33            </div>
34        );
35    }
36
37    const b = build;
38    const vv = v;
39
40    function setStatus(status: BuildSubmission["status"]) {
41        const next: BuildSubmission = { ...b, status, updatedAt: new Date().t
oISOString() };
42        upsertBuild(next);

```

```

43     setBuild(next);
44     toast("Build status updated.", "success", "Saved", 1600);
45   }
46
47   function recalcEstimate() {
48     const est = estimateBuild(vv.inputsSnapshot.dims, vv.inputsSnapshot.
options);
49
50     const nextV = {
51       ...vv,
52       estimatePublic: {
53         total: est.total,
54         rangeLow: est.rangeLow,
55         rangeHigh: est.rangeHigh,
56         materials: est.materials,
57         labor: est.labor,
58         overhead: est.overhead,
59         finish: est.finish,
60       },
61     };
62
63     const next: BuildSubmission = { ...b, updatedAt: new Date().toISOString(),
64     versions: [nextV, ...b.versions.slice(1)] };
65     upsertBuild(next);
66     setBuild(next);
67     toast("Estimate recalculated.", "success", "Saved", 1600);
68   }
69
70   function removeLastCustomerNoteAdmin() {
71     if (!confirm("Remove the most recent customer note and regenerate pr
eviews?")) return;
72     const next = removeLastCustomerNote(b.id);
73     if (!next) {
74       toast("Could not remove note.", "error", "Error", 1800);
75     }
76     setBuild(next);
77     toast("Removed last customer note. Previews re-queued.", "success",
"Saved", 1800);
78   }
79
80   const notesLog = ((vv.inputsSnapshot as any).notesLog || []) as any[];
81   const compiledNotes = compileNotes(notesLog, vv.inputsSnapshot.notes);
82
83   return (
84     <div className="stack page" style={{ gap: 16 }}>
85       <section className="panel card card-center" style={{ maxWidth: 110
0, margin: "0 auto", padding: 18 }}>
86         <div className="row" style={{ justifyContent: "space-between", a
lignItems: "flex-start", gap: 10, flexWrap: "wrap" }}>
87           <div style={{ display: "grid", gap: 6 }}>
88             <h1 className="h2" style={{ margin: 0 }}>Admin - Build Detai
l</h1>
89             <div className="muted" style={{ fontWeight: 850 }}>
90               Build: <span className="badge">{String(build.id).slice(-8)
.toUpperCase()}</span> • Created: {fmt(build.createdAt)} • Updated: {fmt(build.u
pdatedAt)}
91             </div>
92             <div className="muted" style={{ fontWeight: 850 }}>
93               Customer: <strong>{build.customer?.name || "-"}</strong> •
{build.customer?.phone || "-"} • {build.customer?.email || "-"}
94             </div>

```

```
95          <div className="muted" style={{ fontWeight: 850 }}>
96              Type: {build.project?.type || "-"} • Dims: {build.project?
97      .dims?.lengthIn ?? "-" }x{build.project?.dims?.widthIn ?? "-" }x{build.project?.di
98      ms?.heightIn ?? "-" } in
99          </div>
100         </div>
101         <div className="row" style={{ gap: 10, flexWrap: "wrap" }}>
102             <Link className="btn btn-ghost" to="/admin/builds">Back</Lin
k>
103             <Link className="btn btn-ghost" to={`/builds/${build.id}`}>0
104             </div>
105         </div>
106         <div className="row" style={{ gap: 8, flexWrap: "wrap", marginTo
p: 12, justifyContent: "center" }}>
107             <span className="badge" style={{ fontWeight: 950 }}>Status: {S
tring(build.status).toUpperCase()}</span>
108             <button className="btn btn-ghost" onClick={() => setStatus("re
viewing")}>Reviewing</button>
109             <button className="btn btn-ghost" onClick={() => setStatus("qu
ote_sent")}>Quote Sent</button>
110             <button className="btn btn-ghost" onClick={() => setStatus("ap
proved")}>Approved</button>
111             <button className="btn btn-ghost" onClick={() => setStatus("in
_build")}>In Build</button>
112             <button className="btn btn-ghost" onClick={() => setStatus("co
mplete")}>Complete</button>
113             </div>
114
115         <div className="panel" style={{ padding: 14, borderRadius: 14, m
arginTop: 12, width: "100%", maxWidth: 1000 }}>
116             <div style={{ fontWeight: 950, color: "#0f172a" }}>Notes (comp
iled)</div>
117             {compiledNotes ? (
118                 <div className="muted" style={{ fontWeight: 850, whiteSpace:
"pre-wrap", marginTop: 8 }}>{compiledNotes}</div>
119             ) : (
120                 <div className="muted" style={{ fontWeight: 850, marginTop:
8 }}>No notes</div>
121             )}
122
123             <div className="row" style={{ gap: 10, flexWrap: "wrap", justi
fyContent: "center", marginTop: 12 }}>
124                 <button className="btn btn-ghost" onClick={removeLastCustome
rNoteAdmin} style={{ fontWeight: 950 }}>
125                     Remove last customer note
126                 </button>
127             </div>
128         </div>
129
130         <div className="panel" style={{ padding: 14, borderRadius: 14, m
arginTop: 12, width: "100%", maxWidth: 1000 }}>
131             <div style={{ fontWeight: 950, color: "#0f172a" }}>Public esti
mate (what customer sees)</div>
132             {!v.estimatePublic ? (
133                 <div className="muted" style={{ fontWeight: 850, marginTop:
8 }}>
134                     No estimate saved yet. Click "Recalculate Estimate".
135                 </div>
136             ) : (
```

```

137         <div style={{ display: "grid", gap: 10, marginTop: 10 }}>
138             <div className="row" style={{ gap: 10, flexWrap: "wrap" }}>
>
139                 <span className="badge rate-bright">Total: {money(v.estimatePublic.total)}</span>
140                 <span className="badge">Range: {money(v.estimatePublic.rangeLow || v.estimatePublic.total)} - {money(v.estimatePublic.rangeHigh || v.estimatePublic.total)}</span>
141             </div>
142             <div className="muted" style={{ fontWeight: 850 }}>
143                 Materials {money(v.estimatePublic.materials)} • Labor {money(v.estimatePublic.labor)} • Finish {money(v.estimatePublic.finish)} • Overhead {money(v.estimatePublic.overhead)}
144             </div>
145         </div>
146     </div>
147
148     <div className="row" style={{ gap: 10, flexWrap: "wrap", justifyContent: "center", marginTop: 12 }}>
149         <button className="btn btn-primary" onClick={recalcEstimate}>Recalculate Estimate</button>
150     </div>
151 </div>
152
153     <div className="panel" style={{ padding: 14, borderRadius: 14, marginTop: 12, width: "100%", maxWidth: 1000 }}>
154         <div style={{ fontWeight: 950, color: "#0f172a" }}>Renders (latest version)</div>
155         <div style={{ marginTop: 10, display: "grid", gridTemplateColumns: "repeat(auto-fit, minmax(220px, 1fr))", gap: 10 }}>
156             {v.renders || []}.map((r) => (
157                 <div key={r.renderId} className="panel" style={{ padding: 10, borderRadius: 12 }}>
158                     <div style={{ fontWeight: 950 }}>{String(r.view).toUpperCase()}</div>
159                     <span className="badge" style={{ marginLeft: 8 }}>{String(r.status).toUpperCase()}</span>
160                     <div style={{ marginTop: 8, width: "100%", height: 140, borderRadius: 12, overflow: "hidden", border: "1px solid rgba(15,23,42,0.16)" }}>
>
161                         {r.imageUrl ? (
162                             <img src={r.imageUrl} alt="render" style={{ width: "100%", height: "100%", objectFit: "cover" }} />
163                         ) : (
164                             <div className="card-center" style={{ width: "100%", height: "100%" }}>
165                                 <div className="muted" style={{ fontWeight: 850 }}>
166                                     </div>
167                                 </div>
168                                 <div className="muted" style={{ fontWeight: 850, marginTop: 8 }}>
169                                     {r.estimatePublic ? `Render est: ${money(r.estimatePublic.total)}` : "Render est: -"}
170                                 </div>
171                                 </div>
172                                 )>
173                             </div>
174                         </div>
175                     </div>
176                 </div>
177             );

```

=====FILE: src/pages/admin/MaintenanceDashboardPage.tsx=====

```
1 import React from "react";
2
3 type StorageEstimate = {
4   quota?: number;
5   usage?: number;
6 };
7
8 function bytes(n: number) {
9   if (!Number.isFinite(n)) return "-";
10  const units = ["B", "KB", "MB", "GB", "TB"];
11  let i = 0;
12  let v = n;
13  while (v >= 1024 && i < units.length - 1) {
14    v /= 1024;
15    i++;
16  }
17  return `${v.toFixed(i === 0 ? 0 : 2)} ${units[i]}`;
18}
19
20 function getLocalStorageBytes(): number {
21  try {
22    let total = 0;
23    for (let i = 0; i < localStorage.length; i++) {
24      const k = localStorage.key(i);
25      if (!k) continue;
26      const v = localStorage.getItem(k) ?? "";
27      total += k.length + v.length;
28    }
29    // Rough UTF-16 byte estimate: 2 bytes per char
30    return total * 2;
31  } catch {
32    return NaN;
33  }
34}
35
36 function downloadJson(filename: string, data: unknown) {
37  const blob = new Blob([JSON.stringify(data, null, 2)], { type: "application/json" });
38  const url = URL.createObjectURL(blob);
39  const a = document.createElement("a");
40  a.href = url;
41  a.download = filename;
42  document.body.appendChild(a);
43  a.click();
44  a.remove();
45  URL.revokeObjectURL(url);
46}
47
48 export default function MaintenanceDashboardPage() {
49  const [est, setEst] = React.useState<StorageEstimate>({});
```

```
57     let stop = false;
58
59     (async () => {
60       try {
61         // Supported in most modern browsers; safe to try/catch
62         // @ts-ignore
63         const r = (await navigator.storage?.estimate?.()) as StorageEsti
mate | undefined;
64         if (!stop && r) setEst(r);
65       } catch {
66         // ignore
67       }
68     })());
69
70     const on = () => setOnline(true);
71     const off = () => setOnline(false);
72     window.addEventListener("online", on);
73     window.addEventListener("offline", off);
74
75     const t = window.setInterval(() => setNow(new Date()), 1000);
76
77     return () => {
78       stop = true;
79       window.removeEventListener("online", on);
80       window.removeEventListener("offline", off);
81       window.clearInterval(t);
82     };
83   }, []);
84
85   const theme =
86     document.documentElement.getAttribute("data-theme") ||
87     (window.matchMedia?.("(prefers-color-scheme: dark)")?.matches ? "dar
k (system)" : "light (system)");
88
89   const report = {
90     generatedAt: now.toISOString(),
91     online,
92     theme,
93     location: window.location.href,
94     userAgent: navigator.userAgent,
95     language: navigator.language,
96     timeZone: Intl.DateTimeFormat().resolvedOptions().timeZone,
97     storage: {
98       localStorageApproxBytes: 1sBytes,
99       estimate: {
100         quota: est.quota,
101         usage: est.usage,
102       },
103     },
104   };
105
106   return (
107     <div className="page" style={{ maxWidth: 1100, margin: "0 auto", pad
ding: "18px 16px" }}>
108       <div className="stack">
109         <div className="panel card">
110           <div className="stack">
111             <div className="h2">Admin • Maintenance Dashboard</div>
112             <div className="muted">
113               Live system snapshot for troubleshooting and health checks
114             <span style={{ fontWeight: 900 }}>URL:</span> <code>{windo
```

```
w.location.pathname}</code>
  115          </div>
  116
  117          <div className="row">
  118              <span className="badge">{online ? "Online" : "Offline"}</s
pan>
  119          <span className="badge">Theme: {theme}</span>
  120          <span className="badge">Time: {now.toLocaleString()}</span
>
  121      </div>
  122
  123      <div className="row" style={{ gap: 10 }}>
  124          <button
  125              className="btn btn-primary"
  126              onClick={() => downloadJson(`maintenance-report-${now.to
ISOString().replace(/[:.]/g, "-")}.json`, report)}
  127          >
  128              Download Diagnostic Report (JSON)
  129          </button>
  130
  131          <button className="btn btn-ghost" onClick={() => window.lo
cation.reload()}>
  132              Refresh Snapshot
  133          </button>
  134      </div>
  135  </div>
  136 </div>
  137
  138  <div className="panel card">
  139      <div className="h3">Storage Health</div>
  140      <div className="stack" style={{ marginTop: 10 }}>
  141          <div className="row" style={{ justifyContent: "space-between
" }}>
  142              <div className="body">localStorage (approx)</div>
  143              <div className="body" style={{ fontWeight: 900 }}>
  144                  {Number.isFinite(lsBytes) ? bytes(lsBytes) : "Not availa
ble"}
  145              </div>
  146          </div>
  147
  148          <div className="row" style={{ justifyContent: "space-between
" }}>
  149              <div className="body">Browser storage usage</div>
  150              <div className="body" style={{ fontWeight: 900 }}>
  151                  {Number.isFinite(est.usage ?? NaN) ? bytes(est.usage as
number) : "Not available"}
  152              </div>
  153          </div>
  154
  155          <div className="row" style={{ justifyContent: "space-between
" }}>
  156              <div className="body">Browser storage quota</div>
  157              <div className="body" style={{ fontWeight: 900 }}>
  158                  {Number.isFinite(est.quota ?? NaN) ? bytes(est.quota as
number) : "Not available"}
  159              </div>
  160          </div>
  161
  162          <div className="muted" style={{ marginTop: 6 }}>
  163              Note: This dashboard shows browser/runtime health. File-ba
sed backup health (ZIP/PDF) is checked via your backup scripts (CLI).
  164      </div>
```

```

165      </div>
166  </div>
167
168  <div className="panel card">
169    <div className="h3">Environment</div>
170    <div className="stack" style={{ marginTop: 10 }}>
171      <div className="row" style={{ justifyContent: "space-between" }}>
172        <div className="body">Time zone</div>
173        <div className="body" style={{ fontWeight: 900 }}>
174          {report.timeZone}
175        </div>
176      </div>
177
178      <div className="row" style={{ justifyContent: "space-between" }}>
179        <div className="body">Language</div>
180        <div className="body" style={{ fontWeight: 900 }}>
181          {report.language}
182        </div>
183      </div>
184
185      <div className="row" style={{ justifyContent: "space-between" }}>
186        <div className="body">User agent</div>
187        <div className="body" style={{ fontWeight: 900, maxWidth: 680, textAlign: "right" }}>
188          {report.userAgent}
189        </div>
190      </div>
191    </div>
192  </div>
193
194  <div className="panel card">
195    <div className="h3">Quick Links</div>
196    <div className="muted" style={{ marginTop: 10 }}>
197      If you don't want to add a Navbar link yet, you can still op
en the page directly:
198      </div>
199      <div className="row" style={{ marginTop: 10 }}>
200        <span className="badge">/admin/maintenance</span>
201      </div>
202    </div>
203  </div>
204  </div>
205  );
206 }

```