

=====

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";
43
44 import AdminRoute from "../components/AdminRoute";
45 import AdminLogin from "../pages/admin/AdminLogin";
46
47 export default function App() {
48   const location = useLocation();
49   const isStayLit = location.pathname.startsWith("/staylit");
50   useThemeMode(isStayLit);
51
52   return (
53     <div
54       style={{
55         minHeight: "100vh",
```

```

55
56 // Dark neon-friendly overlay so text stays readable over ANY ba
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 ,
75   backgroundImageSize: "cover",
76   backgroundPosition: "center",
77   backgroundRepeat: "no-repeat",
78   backgroundAttachment: "fixed",
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
ute><MaintenanceDashboardPage /></AdminRoute></AdminGuard>} />
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
49 @keyframes pageEnter {
50     from {
51         opacity: 0;
52         transform: translateY(10px);
53         filter: blur(2px);
54     }
55     to {
56         opacity: 1;
57         transform: translateY(0);
58         filter: blur(0);
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, BlinkMacSystemFo
nt,
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     // getActiveTheme reads from your theme store (html attribute/localS
76     // torage/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 >
92 <div
93     style={{
94         maxWidth: 1200,
95         margin: "0 auto",
96         padding: "14px 18px",
97         display: "flex",
98         alignItems: "center",
99         justifyContent: "space-between",
100        gap: 16,
101        flexWrap: "wrap",
102    }}
103 >
104     {/* Brand */}
105     <button
106         type="button"
107         onClick={() => navigate("/") }
108         style={{
109             background: "transparent",
110             border: "none",
111             padding: 0,
112             cursor: "pointer",
113             display: "flex",
114             alignItems: "center",
115             gap: 10,
116         }}
117         aria-label="Go to Home"
118     >
119         <img
120             src={rvLogo}
121             alt="Roberts Ventures LLC"
122             style={{
123                 width: 32,
124                 height: 32,
125                 objectFit: "contain",
126                 filter: "drop-shadow(0 0 14px rgba(56,189,248,.22))",
127             }}
128         />
129         <div style={{ display: "grid", lineHeight: 1.15 }}>
130             <div
131                 style={{
132                     fontSize: 18,
133                     fontWeight: 950,
134                     letterSpacing: ".06em",
135                     textTransform: "uppercase",
136                     background: "linear-gradient(90deg,#38bdf8,#a78bfa,#f472
b6)"
137                     ,
138                     WebkitBackgroundClip: "text",
139                     WebkitTextFillColor: "transparent",
140                     filter: "drop-shadow(0 0 16px rgba(56,189,248,.25))",
141                 }}
142             > Roberts Ventures LLC
143             </div>
144             <div
145                 style={{
146                     fontSize: 11,
147                     color: "rgba(248,250,252,.62)",

```

```

148         fontWeight: 800,
149     }}
150     >
151     Services • Stay Lit Candle Co.
152 </div>
153 </div>
154 </button>
155
156 {/* Nav */}
157 <nav
158     style={{
159         display: "flex",
160         gap: 8,
161         flexWrap: "wrap",
162         justifyContent: "center",
163         alignItems: "center",
164     }}
165     aria-label="Primary"
166 >
167     <NavLink
168         to="/"
169         end
170         style={({ isActive }) => ({
171             ...linkStyleBase,
172             ...(isActive ? linkStyleActive : linkStyleInactive),
173             color: "rgba(248,250,252,0.92)",
174         })}
175     >
176     Home
177 </NavLink>
178
179     <NavLink
180         to="/services"
181         style={({ isActive }) => ({
182             ...linkStyleBase,
183             ...(isActive ? linkStyleActive : linkStyleInactive),
184             color: "rgba(248,250,252,0.92)",
185         })}
186     >
187     Services
188 </NavLink>
189
190     <NavLink
191         to="/schedule"
192         style={({ isActive }) => ({
193             ...linkStyleBase,
194             ...(isActive ? linkStyleActive : linkStyleInactive),
195             color: "rgba(248,250,252,0.92)",
196         })}
197     >
198     Schedule
199 </NavLink>
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 >
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 { isAdminAuthenticated } from "../adminAuth";
4
5  type Props = { children: React.ReactElement };
6
7  export default function AdminGuard({ children }: Props) {
8    const loc = useLocation();
9    if (isAdminAuthenticated()) 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
5 // --- Lockout (brute-force protection) ---
6 const LOCK_KEY = "rv_admin_lock_v1";
7 const MAX_FAILS = 5;
8 const LOCK_MINUTES = 15;
9
10 type LockState = {
11   fails: number;
12   lockedUntil: number; // epoch ms; 0 means not locked
13 };
14
15 function now(): number {
16   return Date.now();
17 }
18
19 function safeParse<T>(raw: string | null): T | null {
20   if (!raw) return null;
21   try {
22     return JSON.parse(raw) as T;
23   } catch {
24     return null;
25   }
26 }
27
28 // =====
29 // Session auth (existing)
30 // =====
31
32 export function isAdminAuthenticated(): 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 setAdminAuthenticated(): 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 clearAdminAuthenticated(): void {
53   localStorage.removeItem(KEY);
54 }
55
56 // =====
57 // Lockout helpers (NEW)
58 // =====
59
60 export function getAdminLock(): LockState {
61   const state = safeParse<LockState>(localStorage.getItem(LOCK_KEY));
62   if (!state) return { fails: 0, lockedUntil: 0 };
63
64   // If lock expired, normalize back to unlocked

```

```

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     // Ensure fields exist
71     return {
72         fails: typeof state.fails === "number" ? state.fails : 0,
73         lockedUntil: typeof state.lockedUntil === "number" ? state.lockedUntil : 0,
74     };
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 protection.
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     }
29     return;
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: "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         <form onSubmit={submit} className="stack">
48           <input
49             className="field"
50             type="password"
51             value={code}
52             onChange={(e) => setCode(e.target.value)}
53             placeholder="Passcode"
54             autoComplete="current-password"
55           />
56           {err ? <div style={{ color: "crimson", fontWeight: 800 }}>{err}
57         </div> : null}
58         <button className="btn btn-primary" type="submit">Unlock Admin
59       </form>
60     </div>
61   );
62 }

```

```

=====
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   setAdminAuthenticated,
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
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", borde
r: "1px solid rgba(34,197,94,0.20)" };
76   if (status === "in_progress")
77     return { background: "rgba(14,165,233,0.12)", color: "#0c4a6e", bord
er: "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
progress" | "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: "", creat
edAt: now, updatedAt: now };
202       return {
203         ...r,
204         quote: {
205           ...qd,
206           updatedAt: now,
207           lines: [...qd.lines, { id: uid(), label: "Custom line item", q
ty: 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.qty, 0)
* clampNum(ln.unitPrice, 0), 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.qty, 0) * clampNum(ln.unitPrice, 0);
236       return `-${ln.label} - ${ln.qty} x ${money(ln.unitPrice)} = ${mon
ey(lineTotal)}`;
237     });
238
239     const appt = r.appointmentStart ? fmt(r.appointmentStart) : "-";
240
241     return [
242       `Roberts Ventures LLC - Quote`,
243       ``,
244       `Customer: ${r.customer?.name || "-"}`,
245       `Phone: ${r.customer?.phone || "-"}`,
246       r.customer?.address ? `Address: ${r.customer.address}` : null,
247       `Requested Appointment: ${appt}`,
248       ``,
249       `Line Items:`,
250       ...(lines.length ? lines : [ ` - (No line items yet)` ]),

```

```

251     ``
252     `Subtotal: ${money(subtotal)}`,
253     taxRate > 0 ? `Tax (${taxRate}%): ${money(tax)}` : null,
254     `Total: ${money(total)}`,
255
256     qd?.notes?.trim() ? `Notes: ${qd.notes.trim()}` : null,
257
258     `Reply to confirm and we'll get you scheduled.`
259   ]
260   .filter(Boolean)
261   .join("\n");
262 }
263
264 async function copyQuote(index: number) {
265   const r = reqs[index];
266   if (!r?.quote) return toast("Create a quote first.", "warning", "No
Quote", 1800);
267   const text = buildQuoteText(r);
268   try {
269     await navigator.clipboard.writeText(text);
270     toast("Quote copied to clipboard.", "success", "Copied", 1800);
271   } catch {
272     toast("Clipboard blocked. Select the preview text and copy.", "war
ning", "Clipboard", 2400);
273   }
274 }
275
276 function openEmail(index: number) {
277   const r = reqs[index];
278   if (!r?.quote) return toast("Create a quote first.", "warning", "No
Quote", 1800);
279   const subject = encodeURIComponent(`Roberts Ventures LLC Quote - ${r
.customer?.name || "Customer"}`);
280   const body = encodeURIComponent(buildQuoteText(r));
281   window.location.href = `mailto:?subject=${subject}&body=${body}`;
282 }
283
284 function markQuoteSent(index: number) {
285   const now = new Date().toISOString();
286   updateRequestByIndex(index, (r) => {
287     if (!r.quote) return r;
288     const st = safeStatus(r.status);
289     return {
290       ...r,
291       status: st === "new" ? "in_progress" : st,
292       quote: { ...r.quote, sentAt: now, updatedAt: now },
293     };
294   });
295   toast("Marked quote as sent.", "success", "Saved", 1600);
296 }
297
298 // -----
299 // Render
300 // -----
301 return (
302   <div className="stack page">
303     <section className="panel card card-center" style={{ maxWidth: 110
0, margin: "0 auto" }}>
304       <div className="row" style={{ justifyContent: "space-between", a
lignItems: "center", width: "100%", flexWrap: "wrap", gap: 10 }}>
305         <div style={{ display: "grid", gap: 6 }}>
306           <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={{
boxShadow: "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, ..
.badgeStyle(st) }}>
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                     </div>
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.address
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                                     {(r.items ?? []).map((it, j) => (
409                                     <div key={j} className="panel" style={{ padding: 1
0, borderRadius: 12 }}>
410                                     <div style={{ fontWeight: 950 }}>
411                                         {it.serviceId} <span className="muted">· qty {
it.qty}</span>
412                                     </div>
413                                         {it.note ? (
414                                             <div className="muted" style={{ fontWeight: 80
415                                                 {it.note}
416                                             </div>
417                                         ) : null}
418                                     </div>
419                                 )))}
420                             </div>
421                         </div>
422
423                             {quoteOpen ? (
424                                 <div style={{ marginTop: 16 }} className="panel card">
425                                     <div className="row" style={{ justifyContent: "space
-between", alignItems: "flex-start", gap: 10, flexWrap: "wrap" }}>
426                                         <div style={{ display: "grid", gap: 6 }}>
427                                             <div className="h3" style={{ margin: 0 }}>Quote
Builder</div>
428                                             <div className="muted" style={{ fontWeight: 900
}}>
429                                                 Edit unit prices, add custom lines, then copy
or email the quote.
430                                             </div>
431                                         </div>
432
433                                             <div className="row" style={{ gap: 8, flexWrap: "w
rap" }}>
434                                                 <span className="badge rate-bright" style={{ jus
tifyContent: "center" }}>Subtotal: {money(totals.subtotal)}</span>
435                                                 <span className="badge rate-bright" style={{ jus
tifyContent: "center" }}>Total: {money(totals.total)}</span>
436                                             </div>
437                                         </div>
438
439                                     <div style={{ marginTop: 12, display: "grid", gap: 1
0 }}>
440                                         {(r.quote?.lines ?? []).map((ln) => (
441                                             <div key={ln.id} className="panel" style={{ padd
ing: 12, borderRadius: 12, display: "grid", gap: 10 }}>
442                                                 <div style={{ display: "grid", gridTemplateCol
umns: "1fr 110px 140px 90px", gap: 10, alignItems: "center" }}>
443                                                     <input className="field" value={ln.label} on
Change={e => setQuoteLine(idx, ln.id, { label: e.target.value })} />
444                                                     <input className="field" inputMode="numeric"
value={ln.qty} onChange={e => setQuoteLine(idx, ln.id, { qty: clampNum(e.targ
et.value, 1) })} />
445                                                     <input className="field" inputMode="decimal"
value={ln.unitPrice} onChange={e => setQuoteLine(idx, ln.id, { unitPrice: cla
mpNum(e.target.value, 0) })} />
446                                                     <button className="btn btn-ghost" onClick={e
=> removeQuoteLine(idx, ln.id)}>Remove</button>
447                                                 </div>
448
449                                             <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

```



```

note.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 </div>
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:
10, flexWrap: "wrap" }}>
74         <input className="field" style={{ flex: 1, minWidth: 240 }} va
lue={q} onChange={(e) => setQ(e.target.value)} placeholder="Search name, phone,
email, status, type..." />
75         <span className="badge" style={{ justifyContent: "center" }}>T
otal: {filtered.length}</span>
76     </div>
77 </section>
78
79     {filtered.length === 0 ? (
80         <section className="panel card card-center" style={{ maxWidth: 1
100, margin: "0 auto" }}>
81             <h3 className="h3">No build submissions found</h3>
82             <p className="body" style={{ maxWidth: 780 }}>
83                 Builds are stored in <code>localStorage</code> on the same d
evice/browser. Once Firebase is connected, these will sync.
84             </p>
85             <Link className="btn btn-ghost" to="/builds/new">Create a test
build</Link>
86         </section>
87     ) : (
88         <section className="stack" style={{ maxWidth: 1100, margin: "0 a
uto" }}>
89             {filtered
90                 .slice()
91                 .sort((a, b) => safe(b.createdAt).localeCompare(safe(a.creat
edAt)))
92                 .map((b) => (
93                     <article key={b.id} className="panel card" style={{ paddin
g: 16 }}>
94                         <div className="row" style={{ justifyContent: "space-bet
ween", alignItems: "flex-start", gap: 10, flexWrap: "wrap" }}>
95                             <div style={{ display: "grid", gap: 6 }}>
96                                 <div style={{ fontWeight: 950, color: "#0f172a" }}>
97                                     Build <span className="badge" style={{ marginLeft:
8, fontSize: 12 }}>{String(b.id).slice(-8).toUpperCase()}</span>
98                                     <span className="badge" style={{ marginLeft: 8 }}>
{String(b.status || "draft").toUpperCase()}</span>
99                                 </div>
100
101                                 <div className="muted" style={{ fontWeight: 850 }}>
102                                     Created: {fmt(b.createdAt)} • Updated: {fmt(b.upda
tedAt)}
103                                 </div>
104
105                                 <div className="muted" style={{ fontWeight: 850 }}>
106                                     Customer: <strong>{b.customer?.name || "-"}</stron
g> • {b.customer?.phone || "-"} • {b.customer?.email || "-"}
107                                 </div>
108
109                                 <div className="muted" style={{ fontWeight: 850 }}>
110                                     Type: {b.project?.type || "-"} • Dims: {b.project?
.dims?.lengthIn ?? "-"}x{b.project?.dims?.widthIn ?? "-"}x{b.project?.dims?.heig
htIn ?? "-"} in
111                                 </div>
112                             </div>
113

```

```

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, type
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: "
USD" }).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().
toISOString() };
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(), versions: [nextV, ...b.versions.slice(1)] };
64     upsertBuild(next);
65     setBuild(next);
66     toast("Estimate recalculated.", "success", "Saved", 1600);
67 }
68
69 function removeLastCustomerNoteAdmin() {
70     if (!confirm("Remove the most recent customer note and regenerate pr
reviews?")) return;
71     const next = removeLastCustomerNote(b.id);
72     if (!next) {
73         toast("Could not remove note.", "error", "Error", 1800);
74         return;
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?.
.dims?.lengthIn ?? "-"}x{build.project?.dims?.widthIn ?? "-"}x{build.project?.di
ms?.heightIn ?? "-"} in
97         </div>
98     </div>
99
100     <div className="row" style={{ gap: 10, flexWrap: "wrap" }}>
101         <Link className="btn btn-ghost" to="/admin/builds">Back</Lin
k>
102         <Link className="btn btn-ghost" to={` /builds/${build.id}`}>O
pen customer preview</Link>
103     </div>
104 </div>
105
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.esti
matePublic.total)}</span>
140                 <span className="badge">Range: {money(v.estimatePublic.r
angelow || v.estimatePublic.total)} - {money(v.estimatePublic.rangeHigh || v.est
imatePublic.total)}</span>
141             </div>
142             <div className="muted" style={{ fontWeight: 850 }}>
143                 Materials {money(v.estimatePublic.materials)} • Labor {m
oney(v.estimatePublic.labor)} • Finish {money(v.estimatePublic.finish)} • Overhe
ad {money(v.estimatePublic.overhead)}
144             </div>
145         </div>
146     )}
147
148     <div className="row" style={{ gap: 10, flexWrap: "wrap", justi
fyContent: "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, m
arginTop: 12, width: "100%", maxWidth: 1000 }}>
154         <div style={{ fontWeight: 950, color: "#0f172a" }}>Renders (la
test version)</div>
155         <div style={{ marginTop: 10, display: "grid", gridTemplateColu
mns: "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).toUpper
Case()} <span className="badge" style={{ marginLeft: 8 }}>{String(r.status).toUp
perCase()}</span></div>
159                     <div style={{ marginTop: 8, width: "100%", height: 140,
borderRadius: 12, overflow: "hidden", border: "1px solid rgba(15,23,42,0.16)" }}>
>
160                         {r.imageUrl ? (
161                             <img src={r.imageUrl} alt="render" style={{ widt
h: "100%", height: "100%", objectFit: "cover" }} />
162                         ) : (
163                             <div className="card-center" style={{ width: "100%",
height: "100%" }}>
164                                 <div className="muted" style={{ fontWeight: 850 }}>
>No image yet</div>
165                             </div>
166                         )}
167                     </div>
168                     <div className="muted" style={{ fontWeight: 850, marginT
op: 8 }}>
169                         {r.estimatePublic ? `Render est: ${money(r.estimatePub
lic.total)}` : "Render est: -"}
170                     </div>
171                 </div>
172             )})
173         </div>
174     </div>
175 </section>
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: "applic
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>({});
50   const [lsBytes, setLsBytes] = React.useState<number>(NaN);
51   const [online, setOnline] = React.useState<boolean>(navigator.onLine);
52   const [now, setNow] = React.useState<Date>(new Date());
53
54   React.useEffect(() => {
55     setLsBytes(getLocalStorageBytes());
56

```



```

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: lsBytes,
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 }

```