

Line wrap □

```

1 <!doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8" />
5 <meta name="viewport" content="width=device-width,initial-scale=1,maximum-scale=1" />
6 <title>Akshat Prasad - Sitemap & URL Governance</title>
7 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.5/font/bootstrap-icons.css">
8 <meta name="google-site-verification" content="Kp1gscirNuecIS_9R4LDKE5ql4w4zt0NmneiPktuMu4" />
9   <!-- Responsive -->
10  <meta name="viewport" content="width=device-width, initial-scale=1" />
11  <!-- ===== AUTHOR & NETWORK AUTHORITY ===== -->
12 <meta name="author" content="Akshat Prasad">
13 <meta name="creator" content="Akshat Prasad">
14 <meta name="publisher" content="Akshat Prasad">
15
16 <link rel="canonical" href="https://akshat-881236.github.io/SitemapGeneratorXml/">
17 <link rel="index" href="https://akshat-881236.github.io/sitemapjs/">
18
19 <link rel="me" href="https://github.com/Akshat-881236">
20 <link rel="me" href="https://akshat-881236.github.io/sitemapjs/">
21
22 <meta name="robots" content="index, follow">
23 <!-- ===== AUTHOR & NETWORK AUTHORITY ===== -->
24 <meta name="author" content="Akshat Prasad">
25 <meta name="creator" content="Akshat Prasad">
26 <meta name="publisher" content="Akshat Prasad">
27
28 <link rel="canonical" href="https://akshat-881236.github.io/SitemapGeneratorXml/">
29 <link rel="index" href="https://akshat-881236.github.io/sitemapjs/">
30
31 <link rel="me" href="https://github.com/Akshat-881236">
32 <link rel="me" href="https://akshat-881236.github.io/sitemapjs/">
33 <link rel="me" href="mailto:akshatpsd2005@gmail.com" />
34 <!-- Sitemap.xml Link-->
35 <link rel="sitemap" type="application/xml" title="Sitemap" href="https://akshat-881236.github.io/sitemapjs/sitemap.xml">
36 <meta name="robots" content="index, follow">
37 <script src="https://akshat-881236.github.io/sitemapjs/sitemap.min.js"></script>
38   <!-- Theme -->
39   <meta name="theme-color" content="#2563eb" />
40   <script src="/SitemapGeneratorXml/seoLoader.js" defer></script>
41 <noscript>
42   <meta name="description"
43     content="Frontend Sitemap Generator by Akshat Prasad">
44 </noscript>
45   <!-- PWA -->
46 <link rel="manifest" href="manifest.json">
47
48   <!-- Icons -->
49 <link rel="stylesheet"
50   href="https://cdn.jsdelivr.net/npm/bootstrap-icons/font/bootstrap-icons.css">
51 <style>
52 /* =====
53 Global Styles & Theme Base
54 Shared across platforms. Platform-specific HTML variations
55 are controlled via body.platform-* classes and media queries.
56 ===== */
57 :root{
58   --bg: #f6f7fb;
59   --panel: #ffffff;
60   --muted: #6b7280;
61   --text: #0f172a;
62   --accent: #2563eb;
63   --success: #16a34a;
64   --danger: #dc2626;
65   --glass: rgba(15,23,42,0.04);
66   --shadow: 0 6px 18px rgba(7,12,34,0.06);
67   --radius: 10px;
68   --max-width: 1200px;
69   --header-height: 72px;
70   --sidebar-width: 260px;
71 }
72 [data-theme="dark"]{
73   --bg: #071024;
74   --panel: #071226;
75   --muted: #9aa6b2;
76   --text: #e6eef6;
77   --accent: #4f8cff;
78   --success: #34d399;
79   --danger: #ff7b7b;
80   --glass: rgba(255,255,255,0.03);
81   --shadow: 0 6px 18px rgba(0,0,0,0.6);
82 }
83 [data-theme="sepia"]{
84   --bg: #fbf6f0;
85   --panel: #ffff9f3;
86   --muted: #7a6246;
87   --text: #2b1f15;
88   --accent: #a3571a;
89   --success: #2e7d32;
90   --danger: #b00020;
91   --glass: rgba(43,31,21,0.03);
92   --shadow: 0 6px 18px rgba(43,31,21,0.04);
93 }
94 *{box-sizing:border-box}
95 html,body{height:100%;margin:0;font-family:Inter,ui-sans-serif,system-ui,-apple-system,"Segoe UI",Roboto,"Helvetica Neue",Arial; background:var(--bg)};
96 .app{

```

```

97  max-width:var(--max-width);
98  margin:12px auto;
99  padding:12px;
100 display:grid;
101 grid-template-columns:var(--sidebar-width) 1fr 360px;
102 gap:12px;
103 align-items:start;
104 }
105 /* Header shared */
106 .header{
107  grid-column:1/-1;
108  display:flex;
109  align-items:center;
110  justify-content:space-between;
111  gap:12px;
112  height:var(--header-height);
113 }
114 .brand{display:flex;align-items:center;gap:12px}
115 .logo{
116  width:48px;height:48px;border-radius:10px;background:linear-gradient(135deg,var(--accent),#00c2ff);display:flex;align-items:center;justify-content:center
117 }
118 .h1{font-size:18px;font-weight:700}
119 .controls{display:flex;align-items:center;gap:8px}
120 .btn{
121  background:var(--panel);
122  border:1px solid rgba(0,0,0,0.04);
123  padding:8px 12px;border-radius:8px;cursor:pointer;display:inline-flex;align-items:center;gap:8px;box-shadow:var(--shadow);
124  font-size:13px;
125 }
126 .btn.primary{background:linear-gradient(90deg,var(--accent),#00b4ff);color:white;border:0}
127 .btn.ghost{background:transparent;border:1px dashed rgba(0,0,0,0.06)}
128 .icon{font-size:16px;color:var(--muted)}
129 .small{font-size:13px;color:var(--muted)}
130 .sidebar{
131  background:var(--panel);
132  border-radius:var(--radius);
133  padding:12px;
134  box-shadow:var(--shadow);
135  height:calc(100vh - 120px);
136  overflow:auto;
137  width:var(--sidebar-width);
138 }
139 .sidebar.hidden{display:none !important}
140 .tabs{display:flex;flex-direction:column;gap:6px}
141 .tab{
142  display:flex;align-items:center;gap:10px;padding:10px;border-radius:8px;cursor:pointer;color:var(--muted);user-select:none
143 }
144 .tab.active{background:linear-gradient(180deg,rgba(0,0,0,0.02),transparent);color:var(--text);font-weight:600}
145 .main{
146  background:var(--panel);
147  border-radius:var(--radius);
148  padding:16px;
149  box-shadow:var(--shadow);
150  min-height:60vh;
151  overflow:auto;
152 }
153 .right{
154  background:var(--panel);
155  border-radius:var(--radius);
156  padding:12px;
157  box-shadow:var(--shadow);
158  height:calc(100vh - 120px);
159  overflow:auto;
160 }
161 .section-title{display:flex;align-items:center;justify-content:space-between;margin-bottom:12px}
162 .grid{display:grid;gap:12px}
163 .columns-2{grid-template-columns:1fr 1fr}
164 .row{display:flex;gap:8px;align-items:center}
165 .input,textarea,select{
166  width:100%;padding:10px;border-radius:8px;border:1px solid rgba(0,0,0,0.06);background:transparent;color:var(--text)
167 }
168 textarea{min-height:90px;resize:vertical}
169 .list{background:var(--glass);padding:8px;border-radius:8px}
170 .folder-item{display:flex;align-items:center;justify-content:space-between;padding:8px;border-radius:8px}
171 .folder-item .meta{font-size:12px;color:var(--muted)}
172 .url-row{display:flex;align-items:center;gap:8px;padding:8px;border-radius:6px;border:1px solid rgba(0,0,0,0.02);margin-bottom:8px}
173 .url-row .left{flex:1;display:flex;gap:10px;align-items:center}
174 .logo-thumb{width:44px;height:44px;border-radius:6px;object-fit:cover;background:var(--glass);display:inline-block}
175 .badge{background:rgba(0,0,0,0.06);padding:6px 8px;border-radius:6px;font-size:12px}
176 .tag{background:rgba(37,99,235,0.08);color:var(--accent);padding:4px 8px;border-radius:6px;font-size:12px}
177 .toolbar{display:flex;gap:8px;align-items:center}
178 .footer{grid-column:1/-1;margin-top:12px;color:var(--muted);font-size:13px;text-align:center}
179
180 /* Toasts */
181 .toast-wrap{position:fixed;right:20px;bottom:20px;display:flex;flex-direction:column;gap:8px;z-index:9999}
182 .toast{padding:10px 12px;border-radius:8px;color:white;min-width:220px;box-shadow:var(--shadow);display:flex;gap:8px;align-items:center}
183 .toast.success{background:var(--success)}.toast.error{background:var(--danger)}.toast.warn{background:#f59e0b}
184
185 /* XML preview */
186 .xml-preview{background:#0b1220;color:#dff1ff;padding:12px;border-radius:8px;font-family:monospace;font-size:13px;overflow:auto;max-height:360px;border:1px solid #fff}
187 .table{width:100%;border-collapse:collapse}
188 .table th,.table td{padding:8px;border-bottom:1px solid rgba(0,0,0,0.04);text-align:left;font-size:13px}
189 .input:focus,textarea:focus,select:focus,.btn:focus{outline:2px solid rgba(37,99,235,0.12)}
190
191 /* =====
192  Swipe handle & gesture hint
193  ===== */
194 #swipeHandle{

```

```

195 position:fixed;left:0;top:50%;transform:translateY(-50%);z-index:9998;
196 width:18px;height:80px;border-radius:0 8px 8px 0;background:linear-gradient(180deg,rgba(0,0,0,0.06),transparent);display:none;align-items:center;justify-content:center
197 }
198 #swipeHandle.visible{display:flex}
199
200 /* =====
201 Platform-specific adjustments
202 We will toggle body.platform-mobile or platform-desktop
203 to adjust DOM or hide/show elements.
204 ===== */
205
206 /* Mobile compressed layout */
207 body.platform-mobile .app{grid-template-columns:1fr; padding:8px}
208 body.platform-mobile .sidebar{position:fixed;left:0;top:0; height:100vh;z-index:9999;transform:translateX(0);transition:transform .25s ease;display:flex;flex-direction:column;align-items:center}
209 body.platform-mobile .sidebar.hidden{transform:translateX(-110%);display:none}
210 body.platform-mobile .right{display:none}
211 body.platform-mobile .header{height:56px}
212 body.platform-mobile .logo{width:40px;height:40px}
213
214 /* Desktop tweaks */
215 @media (min-width:1100px){
216   body.platform-desktop .sidebar{display:flex}
217   #swipeHandle{display:none !important}
218 }
219
220 /* iOS-specific subtle changes */
221 body.platform-ios .header{background-color: #fff; background-image: none; background-clip: border-box; background-size: 100% 100%; background-position: center; border-radius: 10px; box-sizing: border-box; color: #000; font-size: 1em; height: 56px; margin: 0; padding: 0; position: fixed; top: 0; width: 100%; z-index: 9999; transform: translateX(0); transition: transform 0.25s ease; transform-origin: left top; border: 1px solid #000; box-shadow: 0 0 0 1px #000, 0 2px 5px 0 #000, 0 5px 10px 0 #000, 0 10px 20px 0 #000, 0 20px 40px 0 #000, 0 40px 80px 0 #000, 0 80px 160px 0 #000, 0 160px 320px 0 #000, 0 320px 640px 0 #000, 0 640px 1280px 0 #000, 0 1280px 2560px 0 #000, 0 2560px 5120px 0 #000, 0 5120px 10240px 0 #000, 0 10240px 20480px 0 #000, 0 20480px 40960px 0 #000, 0 40960px 81920px 0 #000, 0 81920px 163840px 0 #000, 0 163840px 327680px 0 #000, 0 327680px 655360px 0 #000, 0 655360px 1310720px 0 #000, 0 1310720px 2621440px 0 #000, 0 2621440px 5242880px 0 #000, 0 5242880px 10485760px 0 #000, 0 10485760px 20971520px 0 #000, 0 20971520px 41943040px 0 #000, 0 41943040px 83886080px 0 #000, 0 83886080px 167772160px 0 #000, 0 167772160px 335544320px 0 #000, 0 335544320px 671088640px 0 #000, 0 671088640px 1342177280px 0 #000, 0 1342177280px 2684354560px 0 #000, 0 2684354560px 5368709120px 0 #000, 0 5368709120px 10737418240px 0 #000, 0 10737418240px 21474836480px 0 #000, 0 21474836480px 42949672960px 0 #000, 0 42949672960px 85899345920px 0 #000, 0 85899345920px 171798691840px 0 #000, 0 171798691840px 343597383680px 0 #000, 0 343597383680px 687194767360px 0 #000, 0 687194767360px 1374389534720px 0 #000, 0 1374389534720px 2748779069440px 0 #000, 0 2748779069440px 5497558138880px 0 #000, 0 5497558138880px 10995116277760px 0 #000, 0 10995116277760px 21990232555520px 0 #000, 0 21990232555520px 43980465111040px 0 #000, 0 43980465111040px 87960930222080px 0 #000, 0 87960930222080px 175921860444160px 0 #000, 0 175921860444160px 351843720888320px 0 #000, 0 351843720888320px 703687441776640px 0 #000, 0 703687441776640px 1407374883553280px 0 #000, 0 1407374883553280px 2814749767106560px 0 #000, 0 2814749767106560px 5629499534213120px 0 #000, 0 5629499534213120px 11258999068426240px 0 #000, 0 11258999068426240px 22517998136852480px 0 #000, 0 22517998136852480px 45035996273704960px 0 #000, 0 45035996273704960px 90071992547409920px 0 #000, 0 90071992547409920px 180143985094819840px 0 #000, 0 180143985094819840px 360287970189639680px 0 #000, 0 360287970189639680px 720575940379279360px 0 #000, 0 720575940379279360px 1441151880758558720px 0 #000, 0 1441151880758558720px 2882303761517117440px 0 #000, 0 2882303761517117440px 5764607523034234880px 0 #000, 0 5764607523034234880px 11529215046068469760px 0 #000, 0 11529215046068469760px 23058430092136939520px 0 #000, 0 23058430092136939520px 46116860184273879040px 0 #000, 0 46116860184273879040px 92233720368547758080px 0 #000, 0 92233720368547758080px 184467440737095516160px 0 #000, 0 184467440737095516160px 368934881474191032320px 0 #000, 0 368934881474191032320px 737869762948382064640px 0 #000, 0 737869762948382064640px 1475739525896764129280px 0 #000, 0 1475739525896764129280px 2951479051793528258560px 0 #000, 0 2951479051793528258560px 5902958103587056517120px 0 #000, 0 5902958103587056517120px 11805916207174113034240px 0 #000, 0 11805916207174113034240px 23611832414348226068480px 0 #000, 0 23611832414348226068480px 47223664828696452136960px 0 #000, 0 47223664828696452136960px 94447329657392904273920px 0 #000, 0 94447329657392904273920px 188894659314785808547840px 0 #000, 0 188894659314785808547840px 377789318629571617095680px 0 #000, 0 377789318629571617095680px 755578637259143234191360px 0 #000, 0 755578637259143234191360px 1511157274518286468382720px 0 #000, 0 1511157274518286468382720px 3022314549036572936765440px 0 #000, 0 3022314549036572936765440px 6044629098073145873530880px 0 #000, 0 6044629098073145873530880px 12089258196146291747061760px 0 #000, 0 12089258196146291747061760px 24178516392292583494123520px 0 #000, 0 24178516392292583494123520px 48357032784585166988247040px 0 #000, 0 48357032784585166988247040px 96714065569170333976494080px 0 #000, 0 96714065569170333976494080px 193428131138340667952988160px 0 #000, 0 193428131138340667952988160px 386856262276681335905976320px 0 #000, 0 386856262276681335905976320px 773712524553362671811952640px 0 #000, 0 773712524553362671811952640px 1547425049106725343623905280px 0 #000, 0 1547425049106725343623905280px 3094850098213450687247810560px 0 #000, 0 3094850098213450687247810560px 6189700196426901374495621120px 0 #000, 0 6189700196426901374495621120px 12379400392853802748991242240px 0 #000, 0 12379400392853802748991242240px 24758800785707605497982484480px 0 #000, 0 24758800785707605497982484480px 49517601571415210995964968960px 0 #000, 0 49517601571415210995964968960px 99035203142830421991929937920px 0 #000, 0 99035203142830421991929937920px 198070406285660843983859875840px 0 #000, 0 198070406285660843983859875840px 396140812571321687967719751680px 0 #000, 0 396140812571321687967719751680px 792281625142643375935439503360px 0 #000, 0 792281625142643375935439503360px 1584563250285286751870799006720px 0 #000, 0 1584563250285286751870799006720px 3169126500570573503741598013440px 0 #000, 0 3169126500570573503741598013440px 6338253001141147007483196026880px 0 #000, 0 6338253001141147007483196026880px 12676506002282294014966392053760px 0 #000, 0 12676506002282294014966392053760px 25353012004564588029932784107520px 0 #000, 0 25353012004564588029932784107520px 50706024009129176059865568215040px 0 #000, 0 50706024009129176059865568215040px 10141204801825835219771133630080px 0 #000, 0 10141204801825835219771133630080px 20282409603651670439542267260160px 0 #000, 0 20282409603651670439542267260160px 40564819207303340879084534520320px 0 #000, 0 40564819207303340879084534520320px 81129638414606681758169069040640px 0 #000, 0 81129638414606681758169069040640px 162259276829213363516338138081280px 0 #000, 0 162259276829213363516338138081280px 324518553658426727032676276162560px 0 #000, 0 324518553658426727032676276162560px 649037107316853454065352552325120px 0 #000, 0 649037107316853454065352552325120px 1298074214633706908130705104650240px 0 #000, 0 1298074214633706908130705104650240px 2596148429267413816261410209300480px 0 #000, 0 2596148429267413816261410209300480px 5192296858534827632522820418600960px 0 #000, 0 5192296858534827632522820418600960px 10384593717069655265045640837201920px 0 #000, 0 10384593717069655265045640837201920px 20769187434139310530091281674403840px 0 #000, 0 20769187434139310530091281674403840px 41538374868278621060182563348807680px 0 #000, 0 41538374868278621060182563348807680px 83076749736557242120365126697615360px 0 #000, 0 83076749736557242120365126697615360px 16615349947311448424073025339523120px 0 #000, 0 16615349947311448424073025339523120px 33230699894622896848146050678546240px 0 #000, 0 33230699894622896848146050678546240px 66461399789245793696292101357092480px 0 #000, 0 66461399789245793696292101357092480px 132922799578491587392584202714184960px 0 #000, 0 132922799578491587392584202714184960px 265845599156983174785168405428369920px 0 #000, 0 265845599156983174785168405428369920px 531691198313966349570336810856739840px 0 #000, 0 531691198313966349570336810856739840px 1063382396627932691540673621713479680px 0 #000, 0 1063382396627932691540673621713479680px 2126764793255865383081347243426959360px 0 #000, 0 2126764793255865383081347243426959360px 4253529586511730766162694486853918720px 0 #000, 0 4253529586511730766162694486853918720px 8507059173023461532325388973707837440px 0 #000, 0 8507059173023461532325388973707837440px 17014118346046923064650777947415674880px 0 #000, 0 17014118346046923064650777947415674880px 3402823669209384612930155589483134960px 0 #000, 0 3402823669209384612930155589483134960px 6805647338418769225860311178966269920px 0 #000, 0 6805647338418769225860311178966269920px 1361129467683753845172062235793253840px 0 #000, 0 1361129467683753845172062235793253840px 2722258935367507690344124471586507680px 0 #000, 0 2722258935367507690344124471586507680px 5444517870735015380688248943173015360px 0 #000, 0 5444517870735015380688248943173015360px 10889035741470030761376497886346030720px 0 #000, 0 10889035741470030761376497886346030720px 21778071482940061522752995772692061440px 0 #000, 0 21778071482940061522752995772692061440px 43556142965880123045505991545384122880px 0 #000, 0 43556142965880123045505991545384122880px 87112285931760246091011983090768245760px 0 #000, 0 87112285931760246091011983090768245760px 174224571863520482182023966181536491520px 0 #000, 0 174224571863520482182023966181536491520px 348449143726740964364047932363072983040px 0 #000, 0 348449143726740964364047932363072983040px 696898287453481928728095864726145966080px 0 #000, 0 696898287453481928728095864726145966080px 139379657490696385755619172945229192160px 0 #000, 0 139379657490696385755619172945229192160px 278759314981392771511238345890458384320px 0 #000, 0 278759314981392771511238345890458384320px 557518629962785543022476691780916768640px 0 #000, 0 557518629962785543022476691780916768640px 1115037259925571086044953383561833537280px 0 #000, 0 1115037259925571086044953383561833537280px 2230074519851142172089856767123667074560px 0 #000, 0 2230074519851142172089856767123667074560px 4460149039702284344179713534247334149120px 0 #000, 0 4460149039702284344179713534247334149120px 8920298079404568688359427068494668298240px 0 #000, 0 8920298079404568688359427068494668298240px 1784059615880913737671885413698933658480px 0 #000, 0 1784059615880913737671885413698933658480px 3568119231761827475343770827397867316960px 0 #000, 0 3568119231761827475343770827397867316960px 7136238463523654950687541654795734633920px 0 #000, 0 7136238463523654950687541654795734633920px 1427247692704730980137582330959146927840px 0 #000, 0 1427247692704730980137582330959146927840px 2854495385409461960275164661918293855680px 0 #000, 0 2854495385409461960275164661918293855680px 5708986770818923920550329323836587711360px 0 #000, 0 5708986770818923920550329323836587711360px 11417973541637847841100648667673175422720px 0 #000, 0 11417973541637847841100648667673175422720px 22835947083275695682201297335346350845440px 0 #000, 0 22835947083275695682201297335346350845440px 4567189416655139136440259467069270168880px 0 #000, 0 4567189416655139136440259467069270168880px 9134378833307278272880518934138540337760px 0 #000, 0 9134378833307278272880518934138540337760px 1826875766661455654576103786827708075520px 0 #000, 0 1826875766661455654576103786827708075520px 3653751533322911309152207573655416151040px 0 #000, 0 3653751533322911309152207573655416151040px 7307503066645822618304415147310832302080px 0 #000, 0 7307503066645822618304415147310832302080px 14615006133296445236608830294621664604160px 0 #000, 0 14615006133296445236608830294621664604160px 29230012266592890473217660589243232308320px 0 #000, 0 29230012266592890473217660589243232308320px 58460024533185780946435321178486464616640px 0 #000, 0 58460024533185780946435321178486464616640px 116920049066371561892876622356972932333280px 0 #000, 0 116920049066371561892876622356972932333280px 233840098132743123785753244713945866666560px 0 #000, 0 233840098132743123785753244713945866666560px 46768019626548624757150648942789173333120px 0 #000, 0 46768019626548624757150648942789173333120px 93536039253097249514301297885578346666240px 0 #000, 0 93536039253097249514301297885578346666240px 187072078506194498028602595771156693332480px 0 #000, 0 187072078506194498028602595771156693332480px 374144157012388996057205191542313386664960px 0 #000, 0 374144157012388996057205191542313386664960px 748288314024777992114410383084626673
```

```

293     <div class="tab" data-route="folders" role="tab"><i class="bi bi-folder2 icon"></i> Folder / Project Manager</div>
294     <div class="tab" data-route="urls" role="tab"><i class="bi bi-link-45deg icon"></i> URL Manager</div>
295     <div class="tab" data-route="sitemap" role="tab"><i class="bi bi-file-earmark-code icon"></i> Sitemap Generator</div>
296     <div class="tab" data-route="meta" role="tab"><i class="bi bi-meta icon"></i> SEO & Meta Manager</div>
297     <div class="tab" data-route="auth" role="tab"><i class="bi bi-key-fill icon"></i> Authorization Panel</div>
298     <div class="tab" data-route="dev" role="tab"><i class="bi bi-tools icon"></i> Developer Tools</div>
299   </div>
300   <div class="rule"></div>
301   <div style="display:flex;gap:8px;align-items:center">
302     <button class="btn" id="importBtn" title="Import DB">Import</button>
303     <button class="btn" id="exportBtn" title="Export DB">Export</button>
304   </div>
305   <div class="rule"></div>
306   <div style="font-size:13px;color:var(--muted);">
307     Tips:
308     <ul style="padding-left:18px;margin:6px 0 0 0">
309       <li>Swipe from left edge to open/close navigation (mobile)</li>
310       <li>Ctrl+1..7 to switch tabs</li>
311       <li>Ctrl+Z / Ctrl+Y Undo/Redo</li>
312       <li>Admin-only destructive actions</li>
313     </ul>
314   </div>
315 </div>
316 </nav>
317
318 <!-- Main -->
319 <main class="main" id="mainPanel" tabindex="0">
320   <!-- Dynamically injected content -->
321   <div id="view"></div>
322 </main>
323
324 <!-- Right panel -->
325 <aside class="right" id="rightPanel">
326   <div class="section-title"><strong>Quick Actions</strong></div><div class="small">Keyboard & Quick Links</div></div>
327   <div class="grid">
328     <div>
329       <label class="small">Global Search</label>
330       <div style="display:flex;gap:8px;margin-top:6px">
331         <input id="globalSearch" class="input" placeholder="Search URLs, titles, tags..." aria-label="Global search">
332         <button class="btn" id="searchBtn"><i class="bi bi-search"></i></button>
333       </div>
334     </div>
335     <div>
336       <label class="small">Filters</label>
337       <div style="display:flex;gap:8px;margin-top:6px">
338         <select id="filterFolder" class="input" aria-label="Filter folder"><option value="">All Folders</option></select>
339         <select id="filterType" class="input" aria-label="Filter network"><option value="">All</option><option value="internal">Internal</option><option value="external">External</option></select>
340       </div>
341     </div>
342     <div>
343       <label class="small">Theme</label>
344       <div style="display:flex;gap:8px;margin-top:6px">
345         <select id="themeSelect" class="input" aria-label="Theme select"><option value="auto">Auto</option><option value="light">Light</option><option value="dark">Dark</option><option value="sepia">Sepia</option></select>
346       </div>
347     </div>
348     <div>
349       <label class="small">System Health</label>
350       <div class="list" id="healthPanel">
351         <div id="healthContent" class="small">Loading...</div>
352       </div>
353     </div>
354     <div>
355       <label class="small">Undo / Redo</label>
356       <div style="display:flex;gap:8px;margin-top:6px">
357         <button class="btn" id="undoBtn" title="Undo (Ctrl+Z)"><i class="bi bi-arrow-counterclockwise"></i></button>
358         <button class="btn" id="redoBtn" title="Redo (Ctrl+Y)"><i class="bi bi-arrow-clockwise"></i></button>
359       </div>
360     </div>
361   </div>
362   <div class="rule"></div>
363
364   <div>
365     <label class="small">Selected URL Preview</label>
366     <div id="previewPanel" class="list" style="margin-top:8px">
367       <div class="small">No URL selected</div>
368     </div>
369   </div>
370 </aside>
371
372 <div class="footer">Sitemap Governance • LocalStorage | <span id="countsLabel">0 folders • 0 URLs</span></div>
373 </div>
374
375 <!-- Toast container -->
376 <div class="toast-wrap" id="toasts" aria-live="polite" aria-atomic="true"></div>
377
378 <!-- Hidden file input for import -->
379 <input type="file" id="importFile" accept="application/json" style="display:none" />
380
381 <script>
382 /*
383 Enhanced Sitemap Governance System (single-file SPA)
384 Updates:
385 - Sidebar navigation toggleable via swipe gestures and click
386 - Platform detection (mobile/desktop/iOS/Android) changes certain UI HTML pieces
387 - Themes include light/dark/sepia and 'auto' detection; theme toggles persist
388 - All advanced features from prior version persist (DB, sitemap, meta, auth, undo/redo)
389 Single global namespace: window.SITEMAP_APP
390 */

```

```

391
392 (function(window, document){
393   'use strict';
394
395   const SITEMAP_APP = (function(){
396
397     /* =====
398      Configuration & Keys
399      ===== */
400   const SCHEMA_VERSION = 1;
401   const STORAGE_KEY = 'sg_db_v' + SCHEMA_VERSION;
402   const AUTH_KEY = 'sg_auth';
403   const THEME_KEY = 'sg_theme_pref';
404   const MANUAL_SITEMAP_KEY = 'sg_manual_sitemap';
405   const UI_STATE_KEY = 'sg_ui_state_v1';
406
407   /* =====
408      Data Schema
409      ===== */
410   const DEFAULT_DB = {
411     meta: {
412       title: 'My Sites Network',
413       description: 'A curated network of GitHub Pages projects.',
414       keywords: 'github, pages, sitemap, seo',
415       author: '',
416       robots: 'index, follow'
417     },
418     folders: [],
419     schemaVersion: SCHEMA_VERSION,
420     settings: {
421       splitSitemapsPerFolder: false,
422       autoUpdateLastmod: true
423     },
424     _lastGenerated: null
425   };
426
427   const uid = (prefix='id') => prefix + '_' + Math.random().toString(36).slice(2,10);
428
429   function safeParseJSON(str, fallback=null){
430     try{ return JSON.parse(str); } catch(e) { return fallback; }
431   }
432
433   function loadDB(){
434     let raw = localStorage.getItem(STORAGE_KEY);
435     if(!raw){
436       // attempt migration fallback keys
437       const fallbackKeys = Object.keys(localStorage).filter(k=>k.startsWith('sg_db_'));
438       if(fallbackKeys.length) raw = localStorage.getItem(fallbackKeys[0]);
439     }
440     const parsed = safeParseJSON(raw, null);
441     if(!parsed || typeof parsed !== 'object' || parsed.schemaVersion !== SCHEMA_VERSION){
442       const db = JSON.parse(JSON.stringify(DEFAULT_DB));
443       saveDB(db);
444       return db;
445     }
446     // defensive: ensure structure exists
447     parsed.folders = Array.isArray(parsed.folders) ? parsed.folders : [];
448     parsed.meta = parsed.meta || DEFAULT_DB.meta;
449     parsed.settings = parsed.settings || DEFAULT_DB.settings;
450     return parsed;
451   }
452
453   function saveDB(db){
454     db.schemaVersion = SCHEMA_VERSION;
455     try{ localStorage.setItem(STORAGE_KEY, JSON.stringify(db)); } catch(e){ console.error('saveDB failed', e); showToast('Failed to save DB', 'error') }
456   }
457
458   /* =====
459      Authorization Flow
460      - Hashing via SubtleCrypto
461      - Session stored in sessionStorage
462      ===== */
463   const AUTH = {
464     async hashPass(pass, saltHex){
465       const enc = new TextEncoder();
466       const salt = saltHex ? hexToBytes(saltHex) : crypto.getRandomValues(new Uint8Array(16));
467       const data = new Uint8Array(salt.length + enc.encode(pass).length);
468       data.set(salt, 0);
469       data.set(enc.encode(pass), salt.length);
470       const digest = await crypto.subtle.digest('SHA-256', data);
471       return {hash: bytesToHex(new Uint8Array(digest)), salt: bytesToHex(salt)};
472     },
473     async setPass(pass){
474       const {hash, salt} = await AUTH.hashPass(pass);
475       localStorage.setItem(AUTH_KEY, JSON.stringify({hash, salt}));
476       showToast('Admin passcode set', 'success');
477     },
478     isPassSet(){ return !!localStorage.getItem(AUTH_KEY); },
479     async verify(pass){
480       const raw = localStorage.getItem(AUTH_KEY);
481       if(!raw) return false;
482       const {hash, salt} = safeParseJSON(raw, {});
483       const res = await AUTH.hashPass(pass, salt);
484       return res.hash === hash;
485     },
486     loginSession(){ sessionStorage.setItem('sg_session','1'); },
487     logoutSession(){ sessionStorage.removeItem('sg_session'); },
488     isAdmin(){ return sessionStorage.getItem('sg_session') === '1'; },
489   }

```



```

587     if(m && m.description){
588         if(m.description.length < META.limits.descriptionMin) errors.push({folder:folder.name,issue:'Meta description too short'});
589         if(m.description.length > META.limits.descriptionMax) errors.push({folder:folder.name,issue:'Meta description too long'});
590         if(descs.has(m.description)) errors.push({folder:folder.name,issue:'Duplicate meta description with ' + descs.get(m.description)} );
591         descs.set(m.description, folder.name);
592     }
593 }
594 if(db.meta){
595     if(!db.meta.description || db.meta.description.trim()=='') errors.push({global:true,issue:'Global meta description empty'});
596     else{
597         if(db.meta.description.length < META.limits.descriptionMin) errors.push({global:true,issue:'Global meta description too short'});
598         if(db.meta.description.length > META.limits.descriptionMax) errors.push({global:true,issue:'Global meta description too long'});
599     }
600 }
601 return errors;
602 }

/* =====
603  App Core: state, undo/redo, DB ops
604 ===== */
605 let currentDB = loadDB();
606 let undoStack = [];
607 let redoStack = [];
608 const UNDO_LIMIT = 200;
609
610 function pushUndo(action){
611     undoStack.push({ts:Date.now(),action,db:deepCopy(currentDB)});
612     if(undoStack.length>UNDO_LIMIT) undoStack.shift();
613     redoStack = [];
614     persistUIState();
615     renderHealth();
616 }
617
618 function canUndo(){ return undoStack.length>0; }
619 function canRedo(){ return redoStack.length>0; }
620
621 function performUndo(){
622     if(!canUndo()) return false;
623     const last = undoStack.pop();
624     redoStack.push({ts:Date.now(), db: deepCopy(currentDB)});
625     currentDB = deepCopy(last.db);
626     saveDB(currentDB);
627     renderAll();
628     showToast('Undo: '+last.action,'success');
629     return true;
630 }
631
632 function performRedo(){
633     if(!canRedo()) return false;
634     const next = redoStack.pop();
635     undoStack.push({ts:Date.now(), db: deepCopy(currentDB)});
636     currentDB = deepCopy(next.db);
637     saveDB(currentDB);
638     renderAll();
639     showToast('Redo','success');
640     return true;
641 }

642 function autosave(action='autosave'){
643     saveDB(currentDB);
644     persistUIState();
645     renderCounts();
646     renderHealth();
647 }
648
649 function deepCopy(o){ return safeParseJSON(JSON.stringify(o), JSON.parse(JSON.stringify(DEFAULT_DB))); }
650
651 /* Folders CRUD */
652 function createFolder(name){
653     if(!name || !name.trim()){ showToast('Folder name required','error'); return null; }
654     pushUndo('Create Folder: ' + name);
655     const f = { id:uid('f'), name: name.trim(), metaOverride: null, urls: []};
656     currentDB.folders.push(f);
657     autosave('createFolder');
658     renderAll();
659     showToast('Folder created','success');
660     return f;
661 }
662
663 function renameFolder(id, newName){
664     const f = findFolder(id);
665     if(!f){ showToast('Folder not found','error'); return false; }
666     pushUndo('Rename Folder: ' + f.name + ' -> ' + newName);
667     f.name = newName;
668     autosave('renameFolder');
669     renderAll();
670     return true;
671 }
672
673 function deleteFolder(id){
674     const f = findFolder(id);
675     if(!f){ showToast('Folder not found','error'); return false; }
676     if(!AUTH.isAdmin()){ showToast('Admin required to delete folder','error'); return false; }
677     pushUndo('Delete Folder: ' + f.name);
678     currentDB.folders = currentDB.folders.filter(x=>x.id!==id);
679     autosave('deleteFolder');
680     renderAll();
681     showToast('Folder deleted','success');
682     return true;
683 }
684
685 function findFolder(id){ return currentDB.folders.find(f=>f.id==id); }

/* URL CRUD */

```

```

685     function validateUrlRecord(u){
686       if(!u.loc || !String(u.loc).trim()) return {ok:false,err:'loc required'};
687       if(!SITEMAP.validLoc(u.loc)) return {ok:false,err:'Invalid loc: must be absolute URL or start with /'};
688       return {ok:true};
689   }
690   function addUrlToFolder(folderId, urlObj){
691     const f = findFolder(folderId);
692     if(!f){ showToast('Folder not found','error'); return null; }
693     const tmp = Object.assign({}, urlTemplate(), urlObj);
694     const v = validateUrlRecord(tmp);
695     if(!v.ok){ showToast(v.err,'error'); return null; }
696     pushUndo('Add URL to ' + f.name);
697     tmp.id = uid('u');
698     f.urls.push(tmp);
699     if(currentDB.settings.autoUpdateLastmod) tmp.lastmod = (new Date()).toISOString();
700     autosave('addUrl');
701     renderAll();
702     return tmp;
703   }
704   function editUrl(folderId, urlId, updates){
705     const f = findFolder(folderId);
706     if(!f) return false;
707     const idx = f.urls.findIndex(u=>u.id==urlId);
708     if(idx<0) return false;
709     const copy = Object.assign({}, f.urls[idx], updates);
710     const v = validateUrlRecord(copy);
711     if(!v.ok){ showToast(v.err,'error'); return false; }
712     pushUndo('Edit URL: ' + f.urls[idx].loc);
713     f.urls[idx] = copy;
714     if(currentDB.settings.autoUpdateLastmod) f.urls[idx].lastmod = (new Date()).toISOString();
715     autosave('editUrl');
716     renderAll();
717     return true;
718   }
719   function deleteUrl(folderId, urlId){
720     if(!AUTH.isAdmin()){ showToast('Admin required to delete URL','error'); return false; }
721     const f = findFolder(folderId);
722     if(!f) return false;
723     const idx = f.urls.findIndex(u=>u.id==urlId);
724     if(idx<0) return false;
725     pushUndo('Delete URL: ' + f.urls[idx].loc);
726     f.urls.splice(idx,1);
727     autosave('deleteUrl');
728     renderAll();
729     return true;
730   }
731   function cloneUrl(folderId, urlId, toFolderId=null){
732     const from = findFolder(folderId);
733     if(!from) return false;
734     const u = from.urls.find(x=>x.id==urlId);
735     if(!u) return false;
736     const target = toFolderId ? findFolder(toFolderId) : from;
737     if(!target) return false;
738     pushUndo('Clone URL: ' + u.loc);
739     const copy = deepcopy(u);
740     copy.id = uid('u');
741     target.urls.push(copy);
742     autosave('cloneUrl');
743     renderAll();
744     return true;
745   }
746   function moveUrl(fromFolderId, urlId, toFolderId){
747     const from = findFolder(fromFolderId);
748     const to = findFolder(toFolderId);
749     if(!from || !to) return false;
750     const idx = from.urls.findIndex(x=>x.id==urlId);
751     if(idx<0) return false;
752     pushUndo(`Move URL ${from.urls[idx].loc} -> ${to.name}`);
753     const [u] = from.urls.splice(idx,1);
754     to.urls.push(u);
755     autosave('moveUrl');
756     renderAll();
757     return true;
758   }
759
760   function urlTemplate(){ return { loc: '', title: '', description: '', tags: [], network: 'internal', logo: '', lastmod: '', changefreq: '', priorit
761
762   /* =====
763    Import / Export / Reset
764    ===== */
765   function exportDB(){
766     const dataStr = JSON.stringify(currentDB, null, 2);
767     const blob = new Blob([dataStr], {type:'application/json'});
768     const url = URL.createObjectURL(blob);
769     const a = document.createElement('a'); a.href = url; a.download = `sg_db_export_${(new Date().toISOString()).slice(0,10)}.json`;
770     document.body.appendChild(a); a.click(); a.remove(); URL.revokeObjectURL(url); showToast('Export started','success');
771   }
772   function importDBFromFile(file){
773     const reader = new FileReader();
774     reader.onload = function(e){
775       try{
776         const parsed = JSON.parse(String(e.target.result));
777         if(!parsed || typeof parsed != 'object' || !Array.isArray(parsed.folders)){ showToast('Invalid import file','error'); return; }
778         parsed.schemaVersion = SCHEMA_VERSION;
779         pushUndo('Import DB');
780         currentDB = parsed;
781         saveDB(currentDB);
782         undoStack = []; redoStack = [];
783       }
784     }
785   }

```

```

783     renderAll();
784     showToast('Import successful','success');
785   }catch(err){
786     showToast('Import failed: ' + err.message,'error'); console.error(err);
787   }
788 }
789 reader.readAsText(file);
790 }
791 function resetSystem(){
792   if(!AUTH.isAdmin()){ showToast('Admin required to reset DB','error'); return; }
793   if(!confirm('Reset system and delete all data? This action is irreversible.')) return;
794   pushUndo('Reset DB');
795   currentDB = JSON.parse(JSON.stringify(DEFAULT_DB));
796   saveDB(currentDB);
797   undoStack = []; redoStack = [];
798   showToast('System reset','success');
799   renderAll();
800 }
801
802 /* =====
803    UI Rendering & Router
804    ===== */
805 const dom = {
806   app: document.getElementById('app'),
807   navTabs: document.getElementById('navTabs'),
808   view: document.getElementById('view'),
809   routeHint: document.getElementById('routeHint'),
810   dbVersionLabel: document.getElementById('dbVersionLabel'),
811   countsLabel: document.getElementById('countsLabel'),
812   rightPanel: document.getElementById('rightPanel'),
813   previewPanel: document.getElementById('previewPanel'),
814   healthContent: document.getElementById('healthContent'),
815   filterFolder: document.getElementById('filterFolder'),
816   filterType: document.getElementById('filterType'),
817   globalSearch: document.getElementById('globalSearch'),
818   themeSelect: document.getElementById('themeSelect'),
819   themeToggle: document.getElementById('themeToggle'),
820   loginBtn: document.getElementById('loginBtn'),
821   loginLabel: document.getElementById('loginLabel'),
822   importBtn: document.getElementById('importBtn'),
823   exportBtn: document.getElementById('exportBtn'),
824   importFile: document.getElementById('importFile'),
825   undoBtn: document.getElementById('undoBtn'),
826   redoBtn: document.getElementById('redoBtn'),
827   searchBtn: document.getElementById('searchBtn'),
828   helpBtn: document.getElementById('helpBtn'),
829   sidebar: document.getElementById('sidebar'),
830   sidebarToggle: document.getElementById('sidebarToggle'),
831   swipeHandle: document.getElementById('swipeHandle'),
832   platformBadge: document.getElementById('platformBadge')
833 };
834
835 const ROUTES = ['dashboard', 'folders', 'urls', 'sitemap', 'meta', 'auth', 'dev'];
836 let currentRoute = 'dashboard';
837
838 function navigate(route, push=true){
839   if(!ROUTES.includes(route)) route='dashboard';
840   currentRoute = route;
841   updateTabs();
842   renderView(route);
843   dom.routeHint.textContent = route[0].toUpperCase() + route.slice(1);
844   if(push) history.pushState({route}, '', '#'+route);
845 }
846
847 window.addEventListener('popstate', (e)=> {
848   const st = (e.state && e.state.route) ? e.state.route : (location.hash ? location.hash.slice(1) : 'dashboard');
849   navigate(st, false);
850 });
851
852 function updateTabs(){
853   Array.from(dom.navTabs.querySelectorAll('.tab')).forEach(t=>{
854     if(t.dataset.route === currentRoute) t.classList.add('active'); else t.classList.remove('active');
855   });
856 }
857
858 /* =====
859    Platform Detection & UI Variants
860    - Adds classes to body: platform-mobile, platform-desktop, platform-ios, platform-android
861    - Adjusts small bits of HTML (compact header) for mobile
862    ===== */
863 const PLATFORM = detectPlatform();
864 function detectPlatform(){
865   const ua = navigator.userAgent || '';
866   const isMobile = /Mobi|Android|iPhone|iPad|iPod|Opera Mini/i.test(ua) || window.innerWidth < 900;
867   const isIOS = /iPad|iPhone|iPod/.test(ua) && !window.MSStream;
868   const isAndroid = /Android/.test(ua);
869   return { isMobile, isDesktop: !isMobile, isIOS, isAndroid, ua };
870 }
871
872 function applyPlatformClasses(){
873   const body = document.body;
874   if(PLATFORM.isMobile) body.classList.add('platform-mobile'); else body.classList.add('platform-desktop');
875   if(PLATFORM.isIOS) body.classList.add('platform-ios');
876   if(PLATFORM.isAndroid) body.classList.add('platform-android');
877   // show platform badge
878   const badge = PLATFORM.isMobile ? (PLATFORM.isIOS ? 'iOS' : (PLATFORM.isAndroid ? 'Android' : 'Mobile')) : 'Desktop';
879   dom.platformBadge.textContent = badge;
880   // mobile: smaller title

```

```

881     if(PLATFORM.isMobile){
882         const titleBlock = document.getElementById('titleBlock');
883         if(titleBlock) titleBlock.innerHTML = '<div style="font-weight:700">Sitemap Governance</div>';
884     }
885 }
886
887 /* =====
888  Swipe Gesture Handling
889  - Controlled for both touch and mouse (pointer)
890  - If sidebar hidden, swiping right from left edge opens it
891  - If sidebar visible, swiping left over sidebar closes it
892  - Also provides a visible handle when hidden on mobile
893  ===== */
894 let touchState = {startX:0, startY:0, active:false, isPointer:false};
895 const SWIPE_THRESHOLD = 60; // px
896 const EDGE_REGION = 36; // px from left edge to start open
897
898 function onTouchStart(e){
899     const p = (e.touches && e.touches[0]) || e;
900     touchState.startX = p.clientX; touchState.startY = p.clientY; touchState.active = true; touchState.isPointer = !!e.pointerType;
901 }
902 function onTouchMove(e){
903     if(!touchState.active) return;
904     const p = (e.touches && e.touches[0]) || e;
905     const dx = p.clientX - touchState.startX;
906     const dy = p.clientY - touchState.startY;
907     if(Math.abs(dx) < Math.abs(dy)) return; // vertical scroll
908     // if starting near edge and swipe right
909     if(!isSidebarVisible() && touchState.startX <= EDGE_REGION && dx > SWIPE_THRESHOLD){
910         showSidebar(true);
911         touchState.active = false;
912     }
913     // if sidebar visible and swipe left
914     if(isSidebarVisible() && touchState.startX <= (getSidebarWidth()) && dx < -SWIPE_THRESHOLD){
915         showSidebar(false);
916         touchState.active = false;
917     }
918 }
919 function onTouchEnd(e){ touchState.active = false; }
920
921 function addGestureListeners(){
922     // touch
923     window.addEventListener('touchstart', onTouchStart, {passive:true});
924     window.addEventListener('touchmove', onTouchMove, {passive:true});
925     window.addEventListener('touchend', onTouchEnd);
926     // pointer for mouse drag
927     let pointerDown = false;
928     window.addEventListener('pointerdown', (e)=>{ if(e.isPrimary){ onTouchStart(e); pointerDown=true; } });
929     window.addEventListener('pointermove', (e)=>{ if(pointerDown) onTouchMove(e); });
930     window.addEventListener('pointerup', (e)=>{ pointerDown=false; onTouchEnd(e); });
931     // swipe handle click
932     dom.swipeHandle.addEventListener('click', ()=> showSidebar(true));
933     // sidebar toggle button
934     dom.sidebarToggle.addEventListener('click', ()=> toggleSidebar());
935     // adaptive visibility of handle
936     window.addEventListener('resize', adjustSwipeHandle);
937     adjustSwipeHandle();
938 }
939
940 function getSidebarWidth(){ return parseInt(getComputedStyle(document.documentElement).getPropertyValue('--sidebar-width')) || 260; }
941 function isSidebarVisible(){ return !dom.sidebar.classList.contains('hidden'); }
942 function showSidebar(show=true){
943     if(show){
944         dom.sidebar.classList.remove('hidden');
945         dom.swipeHandle.classList.remove('visible');
946         // on mobile ensure main layout shifts (CSS handles)
947     }else{
948         dom.sidebar.classList.add('hidden');
949         // when hidden, show handle if mobile
950         if(PLATFORM.isMobile) dom.swipeHandle.classList.add('visible');
951     }
952     persistUIState();
953 }
954 function toggleSidebar(){ showSidebar(!isSidebarVisible()); }
955 function adjustSwipeHandle(){
956     if(PLATFORM.isMobile && !isSidebarVisible()) dom.swipeHandle.classList.add('visible'); else dom.swipeHandle.classList.remove('visible');
957     // position handle mid-left
958     const appRect = document.getElementById('app').getBoundingClientRect();
959     dom.swipeHandle.style.top = (appRect.top + appRect.height/2) + 'px';
960 }
961
962 /* =====
963  Search, Filters, Quick Utilities
964  ===== */
965 function performSearch(q){
966     q = String(q || '').trim().toLowerCase();
967     if(!q){ showToast('Empty search','warn'); return; }
968     const results = [];
969     currentDB.folders.forEach(f=>{
970         f.urls.forEach(u=>{
971             const hay = [u.loc,u.title,u.description,(u.tags||[]).join(' '),f.name].join(' ').toLowerCase();
972             if(hay.includes(q)) results.push({folder:f, url:u});
973         });
974     });
975     const html = ['<div class="section-title"><strong>Search Results</strong><div class="small">' + results.length + ' matches</div></div>'];
976     if(!results.length) html.push('<div class="small">No results</div>');
977     else{
978         html.push('<div>');

```

```

979     results.slice(0,200).forEach(r=>{
980       html.push(`<div class="url-row"><div class="left"></div><div style="font-weight:600">${escapeHtml(r.url.title||r.url.loc)}</div><div class="small">${escapeHtml(r.url.loc)}</div></div>`);
981       `)>;
982     html.push('</div>');
983   }
984   dom.view.innerHTML = html.join('');
985   if(PLATFORM.isMobile) showSidebar(false);
986 }
987
988 /* =====
989  Renderers for each route
990  (kept modular; reused logic from previous version)
991  ===== */
992 function renderAll(){ renderCounts(); renderHealth(); renderFilters(); renderView(currentRoute); renderHeader(); META.setGlobal(currentDB); persist
993 function renderHeader(){ document.getElementById('dbVersionLabel').textContent = 'Schema v' + currentDB.schemaVersion; document.getElementById('log
994 function renderCounts(){ const folderCount = currentDB.folders.length; const urlCount = currentDB.folders.reduce((s,f)=>s+f.urls.length,0); documen
995 function renderHealth(){ const urlCount = currentDB.folders.reduce((s,f)=>s+f.urls.length,0); const dupUrls = findDuplicateUrls(); const metaIssues
996
997 function renderFilters(){
998   const sel = dom.filterFolder; if(!sel) return;
999   sel.innerHTML = '<option value="">All Folders</option>';
1000  currentDB.folders.forEach(f= { const o = document.createElement('option'); o.value = f.id; o.textContent = f.name; sel.appendChild(o); });
1001 }
1002
1003 function renderView(route){
1004   switch(route){
1005     case 'dashboard': renderDashboard(); break;
1006     case 'folders': renderFolders(); break;
1007     case 'urls': renderUrlManager(); break;
1008     case 'sitemap': renderSitemap(); break;
1009     case 'meta': renderMetaManager(); break;
1010     case 'auth': renderAuthPanel(); break;
1011     case 'dev': renderDevTools(); break;
1012     default: renderDashboard();
1013   }
1014   // On mobile, auto-hide sidebar after navigating
1015   if(PLATFORM.isMobile) showSidebar(false);
1016 }
1017
1018 /* -----
1019  Dashboard
1020  ----- */
1021 function renderDashboard(){
1022   const counts = currentDB.folders.reduce((acc,f)=>{ acc.urls += f.urls.length; return acc; }, {urls:0});
1023   const dup = findDuplicateUrls();
1024   const metaIssues = validateMeta(currentDB);
1025   const html = [];
1026   html.push(`<div class="section-title"><strong>Dashboard</strong><div class="small">Overview & quick actions</div></div>`);
1027   html.push(`<div class="grid columns-2" style="gap:12px">
1028     <div>
1029       <div class="list">
1030         <div style="display:flex;justify-content:space-between;align-items:center">
1031           <div><strong>${currentDB.folders.length}</strong><div class="small">Folders / Projects</div></div>
1032           <div><strong>${counts.urls}</strong><div class="small">Total URLs</div></div>
1033         </div>
1034         <div class="rule"></div>
1035         <div class="small">Use the Folder Manager to create and organize projects. Sitemap generator will only include sitemap fields.</div>
1036         <div style="margin-top:8px;display:flex;gap:8px">
1037           <button class="btn primary" id="newFolderBtn"><i class="bi bi-folder-plus"></i> New Folder</button>
1038           <button class="btn" id="goToFoldersBtn">Open Folder Manager</button>
1039         </div>
1040       </div>
1041     </div>
1042   </div>
1043
1044   <div style="margin-top:12px" class="list">
1045     <div style="display:flex;justify-content:space-between;align-items:center"><strong>Meta Health</strong><div class="small">${metaIssues.length}</div>
1046     <div class="small" style="margin-top:8px">${metaIssues.length? metaIssues.map(m=> `${m.issue} ${m.folder?'('+m.folder+')':''}`).join('<br>')}
1047     </div>
1048   </div>
1049
1050   <div>
1051     <div class="list">
1052       <strong>Quick Sitemap</strong>
1053       <div class="small">Generate sitemap for selected folders or all data.</div>
1054       <div class="rule"></div>
1055       <div style="display:flex;gap:8px">
1056         <button class="btn" id="quickSitemapBtn"><i class="bi bi-file-earmark-code"></i> Build sitemap.xml</button>
1057         <button class="btn" id="quickIndexBtn"><i class="bi bi-journal"></i> Build sitemap-index.xml</button>
1058       </div>
1059       <div class="rule"></div>
1060       <div class="small">Duplicate URLs: <strong>${dup.length}</strong></div>
1061     </div>
1062
1063     <div style="margin-top:12px" class="list">
1064       <strong>Recent Activity</strong>
1065       <div class="small" id="recentActivity">No recent activity</div>
1066       </div>
1067     </div>
1068   </div>
1069
1070   dom.view.innerHTML = html.join('');
1071
1072   document.getElementById('newFolderBtn').addEventListener('click', ()=>{
1073     const name = prompt('Folder name:');
1074     if(name) createFolder(name);
1075   });
1076   document.getElementById('goToFoldersBtn').addEventListener('click', ()=>navigate('folders'));
1077   document.getElementById('quickSitemapBtn').addEventListener('click', ()=>{ navigate('sitemap'); setTimeout(()=>document.getElementById('buildAllF

```

```

1077
1078    document.getElementById('quickIndexBtn').addEventListener('click', ()=>{ navigate('sitemap'); setTimeout(()=>document.getElementById('buildIndexEx
1079    })
1080
1081    /* -----
1082     Folder Manager
1083     ----- */
1084    function renderFolders(){
1085        const html = ['<div class="section-title"><strong>Folder / Project Manager</strong><div class="small">Create, rename, reorder, and delete project
1086        html.push('<div style="display:flex;gap:12px"><div style="flex:1"><div class="list">');
1087        html.push('<div style="display:flex;justify-content:space-between;align-items:center;margin-bottom:8px"><div><strong>Folders</strong></div><div>
1088        if(!currentDB.folders.length) html.push('<div class="small">No folders. Add one to get started.</div>');
1089        else{
1090            currentDB.folders.forEach(f=>{
1091                html.push('<div class="folder-item" data-id="${f.id}"><div><div style="font-weight:600">${escapeHtml(f.name)}</div><div class="meta small">${
1092                    <div style="display:flex;gap:6px">
1093                        <button class="btn" data-act="open" title="Open"><i class="bi bi-box-arrow-up-right"></i></button>
1094                        <button class="btn" data-act="rename" title="Rename"><i class="bi bi-pencil"></i></button>
1095                        <button class="btn" data-act="meta" title="Meta override"><i class="bi bi-meta"></i></button>
1096                        <button class="btn" data-act="delete" title="Delete"><i class="bi bi-trash"></i></button>
1097                    </div></div>');
1098            });
1099        }
1100        html.push('</div></div><div style="width:320px"><div class="list"><strong>Folder Details</strong><div class="small" id="folderDetails">Select a f
1101        dom.view.innerHTML = html.join('');
1102
1103        document.getElementById('addFolderBtn').addEventListener('click', ()=>{
1104            const name = prompt('Folder name:');
1105            if(name) createFolder(name);
1106        });
1107
1108        Array.from(dom.view.querySelectorAll('.folder-item')).forEach(el=>{
1109            const id = el.dataset.id;
1110            el.querySelector('[data-act="open"]').addEventListener('click', ()=>{ navigate('urls'); setTimeout(()=>renderFolderInUrlPanel(id), 150); });
1111            el.querySelector('[data-act="rename"]').addEventListener('click', ()=>{
1112                const newName = prompt('New name:', findFolder(id).name);
1113                if(newName) renameFolder(id, newName);
1114            });
1115            el.querySelector('[data-act="meta"]').addEventListener('click', ()=>{
1116                const f = findFolder(id);
1117                const desc = f.metaOverride && f.metaOverride.description ? f.metaOverride.description : '';
1118                const title = f.metaOverride && f.metaOverride.title ? f.metaOverride.title : '';
1119                const keywords = f.metaOverride && f.metaOverride.keywords ? f.metaOverride.keywords : '';
1120                const author = f.metaOverride && f.metaOverride.author ? f.metaOverride.author : '';
1121                const robots = f.metaOverride && f.metaOverride.robots ? f.metaOverride.robots : '';
1122                const titleN = prompt('Per-folder title override (leave empty to clear)', title);
1123                const descN = prompt('Per-folder description override (leave empty to clear)', desc);
1124                const keywordsN = prompt('Keywords (comma separated)', keywords);
1125                const authorN = prompt('Author:', author);
1126                const robotsN = prompt('Robots:', robots);
1127                pushUndo('Folder meta override: ' + f.name);
1128                f.metaOverride = (titleN||descN||keywordsN||authorN||robotsN) ? {title: titleN, description: descN, keywords: keywordsN, author: authorN, ro
1129                autosave('folderMeta');
1130                renderAll();
1131            });
1132            el.querySelector('[data-act="delete"]').addEventListener('click', ()=>{ if(confirm('Delete folder and all URLs?')) deleteFolder(id); });
1133        });
1134    }
1135
1136    function renderFolderInUrlPanel(folderId){ navigate('urls'); setTimeout(()=> { const select = dom.view.querySelector('.select-folder-filter'); if(s
1137
1138    /* -----
1139     URL Manager
1140     ----- */
1141    function renderUrlManager(preselectFolderId){
1142        const html = [];
1143        html.push('<div class="section-title"><strong>URL Manager</strong><div class="small">Add, edit, clone, move, and manage URLs. Undo/Redo supported
1144        html.push('<div style="display:flex;gap:12px"><div style="flex:1"><div class="list">');
1145        html.push('<div style="display:flex;justify-content:space-between;align-items:center;margin-bottom:8px"><div><strong>URLs</strong></div><div clas
1146        html.push('<div id="urlListContainer"></div>');
1147        html.push('</div></div><div style="width:320px"><div class="list"><strong>URL Editor</strong><div id="urlEditorPanel" class="small">Select a URL
1148        dom.view.innerHTML = html.join('');
1149
1150        const sel = dom.view.querySelector('.select-folder-filter');
1151        currentDB.folders.forEach(f=> { const o = document.createElement('option'); o.value = f.id; o.textContent = f.name; sel.appendChild(o); });
1152        if(preselectFolderId) sel.value = preselectFolderId;
1153
1154        dom.view.querySelector('#newUrlBtn').addEventListener('click', ()=> openUrlEditor({folderId: sel.value || (currentDB.folders[0] && currentDB.folc
1155        sel.addEventListener('change', ()=> renderUrlList(sel.value));
1156
1157        renderUrlList(sel.value);
1158    }
1159
1160    function renderUrlList(folderId){
1161        const container = dom.view.querySelector('#urlListContainer');
1162        container.innerHTML = '';
1163        const foldersToShow = folderId ? currentDB.folders.filter(f=>f.id==folderId) : currentDB.folders;
1164        if(!foldersToShow.length) container.innerHTML = '<div class="small">No folders available</div>'; return;
1165        foldersToShow.forEach(f=>{
1166            const h = document.createElement('div');
1167            h.innerHTML = `<div style="margin-bottom:6px;display:flex;justify-content:space-between;align-items:center"><div style="font-weight:600">${esc
1168            const list = document.createElement('div');
1169            list.className = 'list';
1170            if(!f.urls.length) list.innerHTML = '<div class="small">No URLs</div>';
1171            else{
1172                f.urls.forEach(u=>{
1173                    const row = document.createElement('div');
1174                    row.className = 'url-row';
1175                    row.innerHTML = `<div class="left"><input type="checkbox" class="select-url" data-folder="${f.id}" data-url="${u.id}" aria-label="Select UF
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2
```

```

1175 
1176 <div><div style="font-weight:600">${escapeHtml(u.title || u.loc)}</div><div class="small">${escapeHtml(u.loc)}</div></div></div>
1177 <div style="display:flex;flex-direction:column;align-items:flex-end;gap:6px">
1178   <div style="display:flex;gap:6px">
1179     <button class="btn" data-act="edit" data-folder="${f.id}" data-url="${u.id}" title="Edit"><i class="bi bi-pencil"></i></button>
1180     <button class="btn" data-act="clone" data-folder="${f.id}" data-url="${u.id}" title="Clone"><i class="bi bi-files"></i></button>
1181     <button class="btn" data-act="move" data-folder="${f.id}" data-url="${u.id}" title="Move"><i class="bi bi-arrow-right-square"></i><
1182       <button class="btn" data-act="delete" data-folder="${f.id}" data-url="${u.id}" title="Delete"><i class="bi bi-trash"></i></button>
1183     </div>
1184     <div class="small">${u.tags && u.tags.length ? u.tags.map(t=>'<span class="tag">'+escapeHtml(t)+'</span>').join(' ') : ''}</div>
1185   </div>;
1186   list.appendChild(row);
1187 );
1188 }
1189 h.appendChild(list);
1190 container.appendChild(h);
1191 );
1192 );
1193 Array.from(container.querySelectorAll('button[data-act]')).forEach(btn=>{
1194   const act = btn.dataset.act;
1195   if(act==='edit') btn.addEventListener('click', ()=> openUrlEditor({folderId:btn.dataset.folder, urlId:btn.dataset.url}));
1196   if(act==='clone') btn.addEventListener('click', ()=> {
1197     const toFolder = prompt('Target folder id (leave empty to same folder):');
1198     cloneUrl(btn.dataset.folder, btn.dataset.url, toFolder || null);
1199   });
1200   if(act==='move') btn.addEventListener('click', ()=> {
1201     const target = prompt('Target folder id:');
1202     if(target) moveUrl(btn.dataset.folder, btn.dataset.url, target);
1203   });
1204   if(act==='delete') btn.addEventListener('click', ()=> {
1205     if(confirm('Delete URL?')) deleteUrl(btn.dataset.folder, btn.dataset.url);
1206   });
1207 });
1208 );
1209 Array.from(container.querySelectorAll('.select-url')).forEach(cb=>
1210   cb.addEventListener('change', (e)=> { const fId = cb.dataset.folder; const uId = cb.dataset.url; if(cb.checked) showPreview(fId, uId); });
1211 );
1212 );
1213 );
1214 function openUrlEditor({folderId, urlId} = {}){
1215   const panel = dom.view.querySelector('#urlEditorPanel');
1216   const folder = folderId ? findFolder(folderId) : (currentDB.folders[0]||null);
1217   if(!folder){ panel.innerHTML = '<div class="small">No folder available. Create a folder first.</div>'; return; }
1218   const record = urlId ? (folder.urls.find(u=>u.id==urlId) || null) : null;
1219   const r = record ? Object.assign({}, record) : urlTemplate();
1220   const html = [];
1221   html.push(<div style="display:grid;gap:8px">');
1222   html.push('<label class="small">Folder</label><select id="editorFolder" class="input">${currentDB.folders.map(f=> <option value="${f.id}" ${f.id==
1223     urlId ? 'selected' : ''}>${f.name}</option>)}</select>');
1224   html.push('<label class="small">URI (loc)</label><input id="editorLoc" class="input" value="${escapeAttr(r.loc)}" placeholder="https://example.cc
1225     />');
1226   html.push('<label class="small">Title</label><input id="editorTitle" class="input" value="${escapeAttr(r.title)}"/>');
1227   html.push('<label class="small">Description</label><textarea id="editorDesc" class="input">${escapeAttr(r.description)}</textarea>');
1228   html.push('<label class="small">Tags (comma separated)</label><input id="editorTags" class="input" value="${escapeAttr((r.tags||[]).join(','))}">');
1229   html.push('<label class="small">Network Type</label><select id="editorNetwork" class="input"><option value="internal">Internal</option><option va
1230     lue="external">External</option></select>');
1231   html.push('<label class="small">Logo URL (optional)</label><input id="editorLogo" class="input" value="${escapeAttr(r.logo)}"/>');
1232   html.push('<div class="rule"></div><div class="small">Sitemap fields (only these will be in XML)</div>');
1233   html.push('<label class="small">Last Modified (ISO)</label><input id="editorLastmod" class="input" value="${escapeAttr(r.lastmod)}"/>');
1234   html.push('<label class="small">Change Frequency</label><select id="editorChangefreq" class="input"><option value="">(leave empty)</option><option
1235     value="never">Never</option><option value="hourly">Hourly</option><option value="daily">Daily</option><option value="weekly">Weekly</option>
1236     <option value="monthly">Monthly</option><option value="yearly">Yearly</option></select>');
1237   html.push('<label class="small">Priority (0.0 - 1.0)</label><input id="editorPriority" class="input" value="${escapeAttr(r.priority)}"/>');
1238   html.push('<div style="display:flex;gap:8px;margin-top:6px"><button class="btn primary" id="saveUrlBtn"><i class="bi bi-save"></i> Save</button>
1239     </div>');
1240   panel.innerHTML = html.join('');
1241 );
1242 panel.querySelector('#editorNetwork').value = r.network || 'internal';
1243 panel.querySelector('#editorChangefreq').value = r.changefreq || '';
1244 );
1245 panel.querySelector('#saveUrlBtn').addEventListener('click', ()=>{
1246   const updated = {
1247     loc: panel.querySelector('#editorLoc').value.trim(),
1248     title: panel.querySelector('#editorTitle').value.trim(),
1249     description: panel.querySelector('#editorDesc').value.trim(),
1250     tags: panel.querySelector('#editorTags').value.split(',').map(s=>s.trim()).filter(Boolean),
1251     network: panel.querySelector('#editorNetwork').value,
1252     logo: panel.querySelector('#editorLogo').value.trim(),
1253     lastmod: panel.querySelector('#editorLastmod').value.trim(),
1254     changefreq: panel.querySelector('#editorChangefreq').value,
1255     priority: panel.querySelector('#editorPriority').value.trim()
1256   };
1257   const targetFolderId = panel.querySelector('#editorFolder').value;
1258   if(urlId){
1259     const t = editUrl(folder.id, urlId, updated);
1260     if(t) showToast('URL updated', 'success');
1261   }else{
1262     const a = addUrlToFolder(targetFolderId, updated);
1263     if(a) showToast('URL added', 'success');
1264   }
1265 });
1266 panel.querySelector('#cancelUrlBtn').addEventListener('click', ()=>{ panel.innerHTML = '<div class="small">Cancelled</div>'; });
1267 );
1268 );
1269 function showPreview(folderId, urlId){
1270   const f = findFolder(folderId); if(!f) return;
1271   const u = f.urls.find(x=>x.id==urlId); if(!u) return;
1272   const html = [];
1273   html.push(<div style="display:flex;gap:8px"><div>
1274     <div class="rule"></div></div>');
1275   html.push('<div class="small">Network: '+escapeHtml(u.network)+'
```

```

1273 }
1274
1275 /* -----
1276   Sitemap Generator View
1277 ----- */
1278 function renderSitemap(){
1279   const html = [];
1280   html.push('<div class="section-title"><strong>Sitemap Generator</strong></div>');
1281   html.push('<div style="display:flex;gap:12px"><div style="flex:1"><div class="list"></div>');
1282   html.push('<div style="display:flex;gap:8px;align-items:center;margin-bottom:8px"><button class="btn" id="selectAllBtn">Select All</button><button');
1283   html.push('</div></div><div style="width:360px"><div class="list"><strong>Preview & Actions</strong><div class="small">XML preview and manual edi');
1284   html.push('<div style="display:flex;gap:8px"><button class="btn" id="manualEditBtn"><i class="bi bi-pencil-square"></i> Manual Edit</button><but');
1285   html.push('<div style="margin-top:8px" class="small">Manual edit is restricted to Admin mode.</div></div></div>');
1286   dom.view.innerHTML = html.join('');
1287
1288 const list = dom.view.querySelector('#sitemapList');
1289 currentDB.folders.forEach(f=>{
1290   const s = document.createElement('div');
1291   s.innerHTML = <div style="font-weight:600;margin-bottom:6px">${escapeHtml(f.name)} <span class="small">(${f.urls.length})</span></div>;
1292   const ul = document.createElement('div'); ul.style.paddingLeft='8px';
1293   f.urls.forEach(u=>{
1294     const row = document.createElement('div'); row.style.display='flex'; row.style.alignItems='center'; row.style.justifyContent='space-between';
1295     const left = document.createElement('div'); left.style.display='flex'; left.style.alignItems='center'; left.style.gap='8px';
1296     const cb = document.createElement('input'); cb.type='checkbox'; cb.dataset.folder=f.id; cb.dataset.url=u.id; cb.className='sitemap-select';
1297     left.appendChild(cb);
1298     const ttl = document.createElement('div'); ttl.innerHTML = `<div style="font-weight:600">${escapeHtml(u.title||u.loc)}</div><div class="small">${u.lastmod}</div>`;
1299     left.appendChild(ttl);
1300     const right = document.createElement('div'); right.innerHTML = `<div class="small">lastmod:${escapeHtml(u.lastmod)} • ${escapeHtml(u.changeTime)}</div>`;
1301     row.appendChild(left); row.appendChild(right); ul.appendChild(row);
1302   });
1303   s.appendChild(ul); list.appendChild(s);
1304 });
1305
1306
1307 dom.view.querySelector('#selectAllBtn').addEventListener('click', ()=> Array.from(dom.view.querySelectorAll('.sitemap-select')).forEach(cb=>cb.click());
1308 dom.view.querySelector('#clearAllBtn').addEventListener('click', ()=> Array.from(dom.view.querySelectorAll('.sitemap-select')).forEach(cb=>cb.checked=false);
1309 dom.view.querySelector('#buildAllBtn').addEventListener('click', ()=> {
1310   const selected = gatherSelectedUrls();
1311   if(!selected.length){ showToast('No URLs selected', 'warn'); return; }
1312   if(currentDB.settings.autoUpdateLastmod){
1313     const now = (new Date()).toISOString();
1314     selected.forEach(u=>{ u.lastmod = now; });
1315   }
1316   const xml = SITEMAP.buildSitemap(selected);
1317   dom.view.querySelector('#xmlPreview').textContent = xml;
1318   showToast('Sitemap generated', 'success');
1319   currentDB._lastGenerated = {type:'sitemap', ts:Date.now(), xml};
1320 });
1321 dom.view.querySelector('#buildIndexBtn').addEventListener('click', ()=>{
1322   const sitemaps = [];
1323   currentDB.folders.forEach(f=>{
1324     if(f.urls.length==0) return;
1325     const xml = SITEMAP.buildSitemap(f.urls);
1326     const blob = new Blob([xml], {type:'application/xml'});
1327     const url = URL.createObjectURL(blob);
1328     sitemaps.push({loc:url, lastmod:(new Date()).toISOString()});
1329   });
1330   const indexXml = SITEMAP.buildSitemapIndex(sitemaps);
1331   dom.view.querySelector('#xmlPreview').textContent = indexXml;
1332   showToast('Sitemap index generated', 'success');
1333   currentDB._lastGenerated = {type:'index', ts:Date.now(), xml:indexXml, sitemaps};
1334 });
1335 dom.view.querySelector('#downloadSitemapBtn').addEventListener('click', ()=>{
1336   if(!currentDB._lastGenerated || !currentDB._lastGenerated.xml){ showToast('No sitemap to download', 'warn'); return; }
1337   const blob = new Blob([currentDB._lastGenerated.xml], {type:'application/xml'});
1338   const a = document.createElement('a'); a.href = URL.createObjectURL(blob); a.download = currentDB._lastGenerated.type==='index' ? 'sitemap-index.xml' : 'sitemap.xml';
1339   document.body.appendChild(a); a.click(); a.remove();
1340   showToast('Download started', 'success');
1341 });
1342 dom.view.querySelector('#copySitemapBtn').addEventListener('click', async ()=>{
1343   if(!currentDB._lastGenerated || !currentDB._lastGenerated.xml){ showToast('No sitemap', 'warn'); return; }
1344   try{ await navigator.clipboard.writeText(currentDB._lastGenerated.xml); showToast('Sitemap copied', 'success'); }catch(e){ showToast('Copy failed', 'error'); }
1345 });
1346 dom.view.querySelector('#manualEditBtn').addEventListener('click', async ()=>{
1347   if(!AUTH.isAdmin()){ const ok = await AUTH.ensureAdmin(); if(!ok) return; }
1348   const current = localStorage.getItem(MANUAL_SITEMAP_KEY) || (currentDB._lastGenerated ? currentDB._lastGenerated.xml : '');
1349   const newXml = prompt('Edit sitemap XML (replace entire content):', current || '');
1350   if(newXml==null) return;
1351   const valid = SITEMAP.validateXMLString(newXml);
1352   if(!valid.ok && !confirm(`XML seems invalid. Save anyway? ${valid.error}`)) return;
1353   localStorage.setItem(MANUAL_SITEMAP_KEY, newXml);
1354   dom.view.querySelector('#xmlPreview').textContent = newXml;
1355   showToast('Manual sitemap saved', 'success');
1356 });
1357 dom.view.querySelector('#shareBtn').addEventListener('click', async ()=>{
1358   if(!currentDB._lastGenerated || !currentDB._lastGenerated.xml){ showToast('No sitemap to share', 'warn'); return; }
1359   if(navigator.share){
1360     try{
1361       const blob = new Blob([currentDB._lastGenerated.xml], {type:'application/xml'});
1362       const filesArray = [new File([blob]), currentDB._lastGenerated.type==='index'? 'sitemap-index.xml': 'sitemap.xml', {type:'application/xml'}];
1363       await navigator.share({title:'Sitemap', files:filesArray, text:'Sitemap generated by SG'});
1364       showToast('Share invoked', 'success');
1365     }catch(e){ showToast('Share failed', 'error'); }
1366     }else{ showToast('Web Share API not supported', 'warn'); }
1367   });
1368
1369 const manual = localStorage.getItem(MANUAL_SITEMAP_KEY);
1370 if(manual) dom.view.querySelector('#xmlPreview').textContent = manual;

```

```

1371 }
1372
1373 function gatherSelectedUrls(){
1374   const selected = [];
1375   Array.from(dom.view.querySelectorAll('.sitemap-select')).forEach(cb=>{
1376     if(cb.checked){
1377       const folder = findFolder(cb.dataset.folder);
1378       const url = folder && folder.urls.find(u=>u.id === cb.dataset.url);
1379       if(url) selected.push(url);
1380     }
1381   });
1382   return selected;
1383 }
1384
1385 /* -----
1386  Meta Manager
1387  ----- */
1388 function renderMetaManager(){
1389   const html = [];
1390   html.push('<div class="section-title"><strong>SEO & Meta Manager</strong><div class="small">Manage global meta tags and per-folder overrides. Adm');
1391   html.push('<div style="display:grid;grid-template-columns:1fr 30px;gap:12px"><div class="list">');
1392   html.push('<div style="display:flex;justify-content:space-between;align-items:center"><div><strong>Global Meta</strong><div class="small">These a');
1393   html.push('<div style="margin-top:8px"><label class="small">Title</label><input id="metaTitle" class="input" value="'+escapeAttr(currentDB.meta.t');
1394   html.push('<label class="small">Description</label><textare a id="metaDescription" class="input" value="'+escapeAttr(currentDB.meta.description)+'</texta');
1395   html.push('<label class="small">Keywords</label><input id="metaKeywords" class="input" value="'+escapeAttr(currentDB.meta.keywords)+'>');
1396   html.push('<label class="small">Author</label><input id="metaAuthor" class="input" value="'+escapeAttr(currentDB.meta.author)+'>');
1397   html.push('<label class="small">Robots</label><input id="metaRobots" class="input" value="'+escapeAttr(currentDB.meta.robots)+'></div>');
1398   html.push('</div><div class="list"><strong>Validation</strong><div class="small" id="metaValidation">Checking...</div></div>');
1399   dom.view.innerHTML = html.join('');
1400   document.getElementById('saveMetaBtn').addEventListener('click', ()=>{
1401     const nmeta = {
1402       title: document.getElementById('metaTitle').value.trim(),
1403       description: document.getElementById('metaDescription').value.trim(),
1404       keywords: document.getElementById('metaKeywords').value.trim(),
1405       author: document.getElementById('metaAuthor').value.trim(),
1406       robots: document.getElementById('metaRobots').value.trim()
1407     };
1408     if(!nmeta.description){ showToast('Meta description cannot be empty','error'); return; }
1409     if(nmeta.description.length < META.limits.descriptionMin || nmeta.description.length > META.limits.descriptionMax){ showToast('Meta descriptor');
1410     pushUndo('Global meta update');
1411     currentDB.meta = nmeta;
1412     autosave('metaSave');
1413     META.setGlobal(currentDB);
1414     showToast('Meta saved','success');
1415   });
1416   const issues = validateMeta(currentDB);
1417   document.getElementById('metaValidation').innerHTML = issues.length ? issues.map(i=>'• ${i.issue} ${i.folder?('('+i.folder+')'):''}).join('<br>');
1418 }
1419
1420 /* -----
1421  Auth Panel
1422  ----- */
1423 function renderAuthPanel(){
1424   const html = [];
1425   html.push('<div class="section-title"><strong>Authorization Panel</strong><div class="small">Manage admin passcode, session, and security options');
1426   html.push('<div class="list"><div class="small">Admin session: <strong>' + (AUTH.isAdmin() ? 'Active' : 'Inactive') + '</strong></div>');
1427   html.push('<div style="display:flex;gap:8px;margin-top:8px"><button class="btn" id="setPassBtn">Set / Reset Passcode</button><button class="btn" id="runChecksBt');
1428   html.push('<div class="rule"></div><div class="small">Passcode is stored hashed with SHA-256 and a random salt. Session is stored in sessionStorage');
1429   dom.view.innerHTML = html.join('');
1430   document.getElementById('setPassBtn').addEventListener('click', async ()=>{
1431     const ok = await AUTH.ensureAdmin('Enter current admin passcode to change (leave empty if none set):');
1432     let proceed = true;
1433     if(!AUTH.isPassSet()) proceed = confirm('No existing passcode detected. Create one now?');
1434     if(!proceed) return;
1435     const newPass = prompt('Enter new admin passcode:');
1436     if(!newPass) return;
1437     await AUTH.setPass(newPass);
1438     showToast('Passcode set','success');
1439     renderAll();
1440   });
1441   document.getElementById('loginActionBtn').addEventListener('click', async ()=>{
1442     if(AUTH.isAdmin()){ AUTH.logoutSession(); showToast('Logged out','success'); renderAll(); } else { const ok = await AUTH.ensureAdmin(); if(ok)
1443   }};
1444 }
1445
1446 /* -----
1447  Developer Tools
1448  ----- */
1449 function renderDevTools(){
1450   const dbJson = JSON.stringify(currentDB, null, 2);
1451   const html = [];
1452   html.push('<div class="section-title"><strong>Developer Tools</strong><div class="small">Diagnostics, integrity checks, import/export, and manual');
1453   html.push('<div style="display:grid;grid-template-columns:1fr 30px;gap:12px"><div class="list"><strong>System Integrity</strong><div class="smal');
1454   html.push('<div style="display:flex;gap:8px"><button class="btn" id="runChecksBtn">Run Checks</button><button class="btn" id="resetBtn">Reset</button>');
1455   html.push('<div class="rule"></div><strong>Raw DB</strong><pre style="white-space:pre-wrap;font-family:monospace;font-size:12px;padding:8px;borde');
1456   html.push('<div class="list"><strong>Quick Utilities</strong><div class="small" style="margin-top:8px"><button class="btn" id="exportBtn2">Export');
1457   dom.view.innerHTML = html.join('');
1458   document.getElementById('runChecksBtn').addEventListener('click', ()=> {
1459     const issues = runIntegrityChecks();
1460     document.getElementById('integrityPanel').innerHTML = issues.length ? issues.map(i=>'• ${i}`).join('<br>') : 'No integrity issues';
1461     showToast('Checks complete','success');
1462   });
1463   document.getElementById('resetBtn').addEventListener('click', resetSystem);
1464   document.getElementById('exportBtn2').addEventListener('click', exportDB);
1465   document.getElementById('importBtn2').addEventListener('click', ()=> dom.importFile.click());
1466   document.getElementById('validateXmlBtn').addEventListener('click', ()=>{
1467     const xml = (currentDB._lastGenerated && currentDB._lastGenerated.xml) || localStorage.getItem(MANUAL_SITEMAP_KEY) || '';
1468     if(!xml){ showToast('No XML to validate','warn'); return; }

```

```

1469     const res = SITEMAP.validateXMLString(xml);
1470     const out = document.getElementById('devOutput');
1471     out.innerHTML = '<div class="small">' + (res.ok?('XML valid: ${escapeHtml(res.error)}')) + '</div>';
1472   }
1473 }
1474
1475 /* -----
1476  * Integrity & Helpers
1477  * -----
1478 */
1479 function findDuplicateUrls(){
1480   const map = new Map(); const dups = [];
1481   currentDB.folders.forEach(f=>{
1482     f.urls.forEach(u=>{
1483       const key = String(u.loc).trim().toLowerCase();
1484       if(map.has(key)) dups.push({loc:u.loc, folders:[map.get(key), f.name]});
1485       else map.set(key, f.name);
1486     });
1487   });
1488   return dups;
1489 }
1490 function runIntegrityChecks(){
1491   const issues=[];
1492   const ids = new Set();
1493   currentDB.folders.forEach(f=>{
1494     if(ids.has(f.id)) issues.push('Duplicate folder id: '+f.id); else ids.add(f.id);
1495     f.urls.forEach(u=>{
1496       if(ids.has(u.id)) issues.push('Duplicate url id: '+u.id); else ids.add(u.id);
1497       if(!u.loc) issues.push('Empty loc in url id: '+u.id);
1498       if(!SITEMAP.validLoc(u.loc)) issues.push('Invalid loc: '+u.loc);
1499       if(!SITEMAP.validChangefreq(u.changefreq) && u.changefreq) issues.push('Invalid changefreq: '+u.changefreq+ ' for '+u.loc);
1500       const pr = SITEMAP.normalizePriority(u.priority);
1501       if(pr!=='' && (isNaN(Number(pr)) || Number(pr)<0 || Number(pr)>1)) issues.push('Invalid priority for '+u.loc);
1502     });
1503   });
1504   const metaIssues = validateMeta(currentDB);
1505   metaIssues.forEach(m=>issues.push('Meta: '+m.issue+(m.folder?(' ('+m.folder+')'): '')));
1506   return issues;
1507 }
1508
1509 /* -----
1510  * Utilities & Helpers
1511  * -----
1512 */
1513 function showToast(msg, type='success', ttl=3500){
1514   const toastContainer = document.getElementById('toasts');
1515   const t = document.createElement('div');
1516   t.className = 'toast ' + (type==='error' ? 'error' : (type==='warn' ? 'warn' : 'success'));
1517   t.innerHTML = `<div style="flex:1">${escapeHtml(msg)}</div><div style="opacity:.9;min-width:30px;text-align:right;color:rgba(255,255,255,0.9)"><t
1518   t.querySelector('button').addEventListener('click', ()=> t.remove());
1519   toastContainer.appendChild(t);
1520   setTimeout(()=>{ t.remove(); }, ttl);
1521 }
1522 function escapeHtml(s){ if(s==null) return ''; return String(s).replace(/(&gt;|&lt;|&gt;|&lt;)/g, function(m){ return {'&gt;': '>', '&lt;': '<', '&gt;': '>', '&lt;': '<'});
1523 function escapeAttr(s){ return escapeHtml(s).replace(/"/g,'quot;');
1524 function sanitizeUrl(u){ if(!u) return ''; try{ const s = String(u).trim(); if(s.startsWith('data:')) return s; const url = new URL(s, location.origin);
1525 function bytesToHex(bytes){ return Array.from(bytes).map(b=>b.toString(16).padStart(2,'0')).join('');
1526 function hexToBytes(hex){ const len = hex.length/2; const out = new Uint8Array(len); for(let i=0;i<len;i++) out[i]=parseInt(hex.substr(i*2,2),16);
1527
1528 /* -----
1529  * Shortcuts & helpers wiring
1530  * -----
1531 */
1532 function globalShortcuts(e){
1533   if(e.ctrlKey && !e.shiftKey && !e.altKey){
1534     if(e.key === '1'){ e.preventDefault(); navigate('dashboard'); }
1535     if(e.key === '2'){ e.preventDefault(); navigate('folders'); }
1536     if(e.key === '3'){ e.preventDefault(); navigate('urls'); }
1537     if(e.key === '4'){ e.preventDefault(); navigate('sitemap'); }
1538     if(e.key === '5'){ e.preventDefault(); navigate('meta'); }
1539     if(e.key === '6'){ e.preventDefault(); navigate('auth'); }
1540     if(e.key === '7'){ e.preventDefault(); navigate('dev'); }
1541     if(e.key === 'z'){ if(!e.shiftKey){ e.preventDefault(); performUndo(); } }
1542     if(e.key === 'y'){ e.preventDefault(); performRedo(); }
1543     if(e.key === 'f'){ e.preventDefault(); document.getElementById('globalSearch').focus(); }
1544   }
1545 }
1546 function showHelp(){ alert('Keyboard Shortcuts:\nCtrl+1..7 switch tabs\nCtrl+Z Undo, Ctrl+Y Redo\nCtrl+F Focus search\nSwipe from left edge to
1547
1548 /* -----
1549  * Theme Management
1550  * - Supports auto/light/dark/sephia
1551  * - Also changes small HTML variant for certain platforms if needed
1552  * -----
1553 */
1554 function applyTheme(pref){
1555   if(pref === 'auto'){
1556     const dark = window.matchMedia('(prefers-color-scheme: dark)').matches;
1557     document.documentElement.setAttribute('data-theme', dark ? 'dark' : 'light');
1558   }else{
1559     document.documentElement.setAttribute('data-theme', pref === 'dark' ? 'dark' : (pref==='sephia' ? 'sephia' : 'light'));
1560   }
1561   localStorage.setItem(THEME_KEY, pref);
1562   // Example of theme-based small DOM modification: change header badge contrast slightly
1563   if(pref === 'sephia') document.body.classList.add('theme-sephia'); else document.body.classList.remove('theme-sephia');
1564 }
1565 function cycleTheme(){
1566   const cur = localStorage.getItem(THEME_KEY) || 'auto';
1567   const order = ['auto','light','dark','sephia'];
1568   const next = order[(order.indexOf(cur)+1)%order.length];
1569   localStorage.setItem(THEME_KEY, next);

```

```

1566     document.getElementById('themeSelect').value = next;
1567     applyTheme(next);
1568     showToast('Theme: ' + next,'success');
1569   }
1570 
1571 /* =====
1572 Persistence for UI state (sidebar hidden, undo size etc)
1573 ===== */
1574 function persistUIState(){
1575   const state = { sidebarHidden: dom.sidebar.classList.contains('hidden'), ts:Date.now(), undo:undoStack.length, redo:redoStack.length };
1576   localStorage.setItem(UI_STATE_KEY, JSON.stringify(state));
1577 }
1578 function restoreUIState(){
1579   const raw = localStorage.getItem(UI_STATE_KEY);
1580   const s = safeParseJSON(raw, {});
1581   if(s && s.sidebarHidden) dom.sidebar.classList.add('hidden'); else dom.sidebar.classList.remove('hidden');
1582   // ensure swipe handle visibility
1583   if(PLATFORM.isMobile && dom.sidebar.classList.contains('hidden')) dom.swipeHandle.classList.add('visible'); else dom.swipeHandle.classList.remove(
1584 )
1585 
1586 /* =====
1587 Attach events & init
1588 ===== */
1589 function init(){
1590   applyPlatformClasses();
1591   addGestureListeners();
1592   // event binding for nav
1593   dom.navTabs.addEventListener('click', (e)=>{ const t = e.target.closest('.tab'); if(t && t.dataset.route) navigate(t.dataset.route); });
1594   dom.loginBtn.addEventListener('click', onLoginClick);
1595   dom.themeSelect.addEventListener('change', (e)=>{ localStorage.setItem(THEME_KEY, e.target.value); applyTheme(e.target.value); });
1596   dom.themeToggle.addEventListener('click', ()=> cycleTheme());
1597   dom.importBtn.addEventListener('click', ()=> dom.importFile.click());
1598   dom.exportBtn.addEventListener('click', exportDB);
1599   dom.importFile.addEventListener('change', (e)=>{ if(e.target.files && e.target.files[0]) importDBFromFile(e.target.files[0]); e.target.value=''; });
1600   dom.undoBtn.addEventListener('click', performUndo);
1601   dom.redoBtn.addEventListener('click', performRedo);
1602   dom.searchBtn.addEventListener('click', ()=>performSearch(dom.globalSearch.value));
1603   dom.helpBtn.addEventListener('click', showHelp);
1604   document.addEventListener('keydown', globalShortcuts);
1605 
1606   // platform-specific initial theme
1607   const savedTheme = localStorage.getItem(THEME_KEY) || 'auto';
1608   document.getElementById('themeSelect').value = savedTheme;
1609   applyTheme(savedTheme);
1610 
1611   // restore UI state
1612   restoreUIState();
1613 
1614   // initial nav based on hash
1615   const initial = location.hash ? location.hash.slice(1) : 'dashboard';
1616   navigate(initial, false);
1617   renderAll();
1618 }
1619 
1620 /* -----
1621   Login click handler for auth panel
1622   -----
1623 */
1624 async function onLoginClick(){
1625   if(AUTH.isAdmin()){
1626     AUTH.logoutSession();
1627     showToast('Logged out','success');
1628     renderAll();
1629   }else{
1630     const ok = await AUTH.ensureAdmin();
1631     if(ok) renderAll();
1632   }
1633 }
1634 
1635 /* =====
1636 Small helper utilities & expose API
1637 ===== */
1638 function escapeHtmlForAttr(s){ return escapeHtml(s).replace(/"/g,'"'); }
1639 
1640 const api = {
1641   getDB: ()=>deepCopy(currentDB),
1642   saveDB: ()=>{ saveDB(currentDB); showToast('DB saved','success'); },
1643   loadDB: ()=>{ currentDB = loadDB(); renderAll(); showToast('DB loaded','success'); },
1644   importFromJSON: (json)=>{ currentDB = json; saveDB(currentDB); renderAll(); },
1645   AUTH, SITEMAP, META
1646 };
1647 
1648 // wire import file input outside of control (global)
1649 document.getElementById('importFile').addEventListener('change', (e)=>{ if(e.target.files && e.target.files[0]) importDBFromFile(e.target.files[0]) });
1650 
1651 // expose global
1652 window.SITEMAP_APP = api;
1653 
1654 // init after microtask to allow DOM ready
1655 setTimeout(init, 60);
1656 
1657 return api;
1658 }()
1659 
1660 /* =====
1661 Utility functions shared by both modules
1662 ===== */
1663 // Basic escape helpers used in inline event contexts

```

```
1664 function escapeHtml(s){ if(s==null) return ''; return String(s).replace(/[\&<>"']/g, function(m){ return {'&':'&amp;','<':'&lt;','>':'&gt;','"':'&quot;'}[m]; })
1665 function escapeAttr(s){ return escapeHtml(s).replace(/\"/g,'&quot;'); }
1666 function sanitizeUrl(u){ if(!u) return ''; try{ const s = String(u).trim(); if(s.startsWith('data:')) return s; const url = new URL(s, location.origin); return url.href; }
1667 function bytesToHex(bytes){ return Array.from(bytes).map(b=>b.toString(16).padStart(2,'0')).join(''); }
1668 function hexToBytes(hex){ const len = hex.length/2; const out = new Uint8Array(len); for(let i=0;i<len;i++) out[i]=parseInt(hex.substr(i*2,2),16); return out;
1669 }
1670 })(window, document);
1671 </script>
1672 </body>
1673 </html>
```