

■ Lost & Found Frontend - React Code Export

■ Frontend React/JS Files:

```
[JSON] package.json
[CSS] src\App.css
[JS] src\App.js
[JS] src\components\AdminTable.js
[JS] src\components\ClaimsTable.js
[JS] src\components\FeedbackForm.js
[JS] src\components\FlashMessage.js
[JS] src\components\Header.js
[JS] src\components\ItemTable.js
[JS] src\components\MessageList.js
[JS] src\config\api.js
[CSS] src\index.css
[JS] src\index.js
[JS] src\pages\AdminPage.js
[JS] src\pages\HomePage.js
[JS] src\pages\ItemRegisterPage.js
[JS] src\pages\ItemsPage.js
[JS] src\pages\LoginPage.js
[JS] src\pages\RegisterPage.js
[JS] tailwind.config.js
```

■ File: package.json

```
=====
1: {
2:   "name": "lost-and-found-frontend",
3:   "version": "0.1.0",
4:   "private": true,
5:   "dependencies": {
6:     "@testing-library/dom": "^10.4.0",
7:     "@testing-library/jest-dom": "^6.6.3",
8:     "@testing-library/react": "^16.3.0",
9:     "@testing-library/user-event": "^13.5.0",
10:    "dayjs": "^1.11.13",
11:    "lucide-react": "^0.525.0",
12:    "react": "^19.1.0",
13:    "react-dom": "^19.1.0",
14:    "react-hot-toast": "^2.5.2",
15:    "react-router-dom": "^7.6.3",
16:    "react-scripts": "5.0.1",
17:    "sweetalert2": "^11.22.2",
18:    "sweetalert2-react-content": "^5.1.0",
19:    "web-vitals": "^2.1.4"
20:  },
21:  "scripts": {
22:    "start": "react-scripts start",
23:    "build": "react-scripts build",
24:    "test": "react-scripts test",
25:    "eject": "react-scripts eject"
26:  },
27:  "eslintConfig": {
28:    "extends": [
29:      "react-app",
30:      "react-app/jest"
31:    ]
32:  },
33:  "browserslist": {
34:    "production": [
35:      ">0.2%",
36:      "not dead",
37:      "not op_mini all"
38:    ],
39:    "development": [
40:      "last 1 chrome version",
41:      "last 1 firefox version",
42:      "last 1 safari version"
43:    ]
44:  },
45:  "devDependencies": {
46:    "autoprefixer": "^10.4.21",
47:    "postcss": "^8.5.6",
48:    "tailwindcss": "^3.4.17"
49:  }
50: }
```

■ File: src\App.css

■ File: src\App.js

```
=====
1: import {
2:   BrowserRouter as Router,
3:   Routes,
4:   Route,
5:   useLocation,
6: } from "react-router-dom";
7: import { useEffect } from "react";
8: import { Toaster } from "react-hot-toast";
9:
10: import LoginPage from "../pages/LoginPage";
11: import RegisterPage from "../pages/RegisterPage";
12: import HomePage from "../pages/HomePage";
13: import AdminPage from "../pages/AdminPage";
14: import ItemRegisterPage from "../pages/ItemRegisterPage";
15: import ItemsPage from "../pages/ItemsPage";
16: import Header from "../components/Header";
17:
18: // Wrapper component to conditionally show header
19: function LayoutWrapper() {
20:   const location = useLocation();
21:
22:   // Define paths where header should be hidden
23:   const hideHeaderPaths = ["/", "/register"];
```

```

24:   const shouldHideHeader = hideHeaderPaths.includes(location.pathname);
25:
26:   // Optional: Scroll to top on route change
27:   useEffect(() => {
28:     window.scrollTo(0, 0);
29:   }, [location.pathname]);
30:
31:   return (
32:     <>
33:       <Toaster position="top-center" reverseOrder={false} />
34:       {!shouldHideHeader && <Header />}
35:       <Routes>
36:         <Route path="/" element={<LoginPage />} />
37:         <Route path="/register" element={<RegisterPage />} />
38:         <Route path="/home" element={<HomePage />} />
39:         <Route path="/admin" element={<AdminPage />} />
40:         <Route path="/register-item" element={<ItemRegisterPage />} />
41:         <Route path="/lost-found-items" element={<ItemsPage />} />
42:       </Routes>
43:     </>
44:   );
45: }
46:
47: export default function App() {
48:   return (
49:     <Router>
50:       <LayoutWrapper />
51:     </Router>
52:   );
53: }

```

■ File: src\components\AdminTable.js

```

1: export default function AdminTable({
2:   title,
3:   type,
4:   data,
5:   headers,
6:   rowRenderer,
7: }) {
8:   return (
9:     <section className="mb-8">
10:      <h2 className="text-xl font-bold text-gray-800 mb-3">{title}</h2>
11:
12:      {data.length === 0 ? (
13:        <p className="text-gray-500 text-sm italic">No {type}s found.</p>
14:      ) : (
15:        <div className="overflow-x-auto shadow border border-gray-200 rounded-lg">
16:          <table className="min-w-full text-sm text-left text-gray-700">
17:            <thead className="bg-gray-100 text-xs uppercase text-gray-600 sticky top-0 z-1
18:            | 0">
19:              <tr>
20:                {headers.map((h, idx) => (
21:                  <th key={idx} className="px-5 py-3 border-b font-bold">
22:                    {h}
23:                  </th>
24:                ))}
25:              </tr>
26:            </thead>
27:            <tbody>
28:              {data.map((item, i) => (
29:                <tr>
30:                  key={i}
31:                  className={`border-b ${
32:                    i % 2 === 0 ? "bg-white" : "bg-gray-50"
33:                  } hover:bg-blue-50 transition duration-150`}
34:                  >
35:                    {rowRenderer(item).map((cell, j) => (
36:                      <td key={j} className="px-5 py-3 whitespace-nowrap">
37:                        {cell}
38:                      </td>
39:                    ))}
40:                  </tr>
41:                ))}
42:              </tbody>
43:            </table>
44:          </div>
45:        )}
46:      </section>
47:    );
48: }

```

■ File: src\components\ClaimsTable.js

```
=====
1: // src/components/ClaimsTable.js
2: import React from "react";
3:
4: export default function ClaimsTable({ claims }) {
5:   const formatDate = (dateString) => {
6:     if (!dateString) return "N/A";
7:     const date = new Date(dateString);
8:     return date.toLocaleDateString("en-US", {
9:       year: "numeric",
10:      month: "short",
11:      day: "numeric",
12:      hour: "2-digit",
13:      minute: "2-digit",
14:    });
15:  };
16:
17:  return (
18:    <div className="overflow-x-auto rounded-xl border border-gray-200">
19:      <table className="min-w-full table-auto">
20:        <thead>
21:          <tr className="bg-gray-100 text-left text-sm font-semibold text-gray-700">
22:            <th className="px-4 py-3">ITEM</th>
23:            <th className="px-4 py-3">LOCATION</th>
24:            <th className="px-4 py-3">CLAIMED AT</th>
25:          </tr>
26:        </thead>
27:        <tbody>
28:          {claims.length === 0 ? (
29:            <tr>
30:              <td colSpan={3} className="text-center py-8 text-gray-500">
31:                No claims yet.
32:              </td>
33:            </tr>
34:          ) : (
35:            claims.map((claim, idx) => (
36:              <tr>
37:                <td key={idx}
38:                  className="border-t text-sm text-gray-800 hover:bg-gray-50 transition"
39:                >
40:                  <td className="px-4 py-3 font-medium">{claim.itemName}</td>
41:                  <td className="px-4 py-3">{claim.location}</td>
42:                  <td className="px-4 py-3 text-sm text-gray-600">
43:                    {formatDate(claim.claimedAt)}
44:                  </td>
45:                </tr>
46:              </td>
47:            </td>
48:          </tbody>
49:        </table>
50:      </div>
51:    );
52:  }
```

■ File: src\components\FeedbackForm.js

```
=====
1: import { useState } from "react";
2: import toast from "react-hot-toast";
3:
4: export default function FeedbackForm() {
5:   const [text, setText] = useState("");
6:   const [loading, setLoading] = useState(false);
7:
8:   const handleSubmit = async (e) => {
9:     e.preventDefault();
10:    if (!text.trim()) return;
11:
12:    setLoading(true);
13:
14:    try {
15:      const res = await fetch("/submit_feedback", {
16:        method: "POST",
17:        headers: { "Content-Type": "application/json" },
18:        credentials: "include",
19:        body: JSON.stringify({ feedback: text.trim() }),
20:      });
21:
22:      if (res.ok) {
23:        setText("");
24:        toast.success("Thank you for your feedback!");
25:      } else {
26:        toast.error("Error submitting feedback.");

```

```

27:     }
28:   } catch (err) {
29:     toast.error("Network error. Please try again.");
30:   } finally {
31:     setLoading(false);
32:   }
33: };
34:
35: return (
36:   <form onSubmit={handleSubmit} className="space-y-4">
37:     <textarea
38:       className="w-full px-4 py-2 border border-gray-300 rounded-xl focus:outline-none focus:ring-2 focus:ring-green-500 text-sm resize-none"
39:       rows="4"
40:       required
41:       value={text}
42:       onChange={(e) => setText(e.target.value)}
43:       placeholder="Write your feedback..."
44:     />
45:     <div className="flex justify-end">
46:       <button
47:         type="submit"
48:         disabled={loading}
49:         className="bg-green-600 hover:bg-green-700 text-white px-6 py-2 rounded-xl transition disabled:opacity-50"
50:       >
51:         {loading ? "Submitting..." : "Submit"}
52:       </button>
53:     </div>
54:   </form>
55: );
56: }

```

■ File: src\components\FlashMessage.js

■ File: src\components\Header.js

```

1: // src/components/Header.js
2: import { Link, useNavigate } from "react-router-dom";
3: import { useEffect, useState } from "react";
4: import { API_BASE_URL, API_ENDPOINTS } from "../config/api";
5: import toast from "react-hot-toast";
6:
7: export default function Header() {
8:   const [user, setUser] = useState(null);
9:   const [loading, setLoading] = useState(true);
10:  const [sidebarOpen, setSidebarOpen] = useState(false);
11:  const navigate = useNavigate();
12:
13:  useEffect(() => {
14:    fetchUser();
15:  }, []);
16:
17:  const fetchUser = async () => {
18:    try {
19:      const res = await fetch(`${API_BASE_URL}${API_ENDPOINTS.ME}`, {
20:        credentials: "include", // Send cookies with request
21:      });
22:
23:      if (res.ok) {
24:        const response = await res.json();
25:        // Backend returns: { success: true, message: "...", data: { id, name, email, role
26:        | } }
27:        setUser(response.data);
28:      } else {
29:        setUser(null);
30:      }
31:    } catch (err) {
32:      console.error("Error fetching user:", err);
33:      setUser(null);
34:    } finally {
35:      setLoading(false);
36:    }
37:  };
38:
39:  const handleLogout = async () => {
40:    try {
41:      const res = await fetch(`${API_BASE_URL}${API_ENDPOINTS.LOGOUT}`, {
42:        method: "POST",
43:        credentials: "include", // Send cookies

```

```

43:     });
44:
45:     if (res.ok) {
46:         setUser(null);
47:         toast.success("Logged out successfully");
48:         navigate("/");
49:     } else {
50:         toast.error("Logout failed");
51:     }
52: } catch (err) {
53:     console.error("Logout error:", err);
54:     toast.error("Logout failed");
55: }
56: };
57:
58: if (loading) return null;
59:
60: return (
61:     <>
62:         <header className="bg-white text-gray-800 shadow-md sticky top-0 z-30">
63:             <div className="max-w-7xl mx-auto px-4 py-3 flex items-center justify-between">
64:                 { /* Logo */ }
65:                 <Link to={user?.role === "ADMIN" ? "/admin" : "/home"}>
66:                     <h1 className="text-2xl font-bold text-blue-600 hover:text-blue-700 transition
| ">
67:                         Lost & Found
68:                     </h1>
69:                 </Link>
70:
71:                 { /* Desktop nav */ }
72:                 <nav className="hidden md:flex items-center gap-6 text-sm">
73:                     <Link
74:                         to="/home"
75:                         className="hover:text-blue-600 transition font-medium"
76:                     >
77:                         Home
78:                     </Link>
79:
80:                     {user ? (
81:                         <>
82:                             <Link
83:                                 to="/register-item"
84:                                 className="hover:text-blue-600 transition font-medium"
85:                             >
86:                                 Register Item
87:                             </Link>
88:                             <Link
89:                                 to="/lost-found-items"
90:                                 className="hover:text-blue-600 transition font-medium"
91:                             >
92:                                 Browse Items
93:                             </Link>
94:                             {user.role === "ADMIN" && (
95:                                 <Link
96:                                     to="/admin"
97:                                     className="hover:text-blue-600 transition font-medium"
98:                                 >
99:                                     Admin Dashboard
100:                                 </Link>
101:                             )}
102:                             <div className="flex items-center gap-3 ml-2">
103:                                 <span className="text-sm text-gray-600 bg-gray-100 px-3 py-1 rounded-ful
| 1">
104:                                     Hi, {user.name}
105:                                 </span>
106:                                 <button
107:                                     onClick={handleLogout}
108:                                     className="bg-red-600 text-white px-4 py-2 rounded-lg hover:bg-red-700
| transition font-medium"
109:                                 >
110:                                     Logout
111:                                 </button>
112:                             </div>
113:                         </>
114:                     ) : (
115:                         <>
116:                             <Link
117:                                 to="/"
118:                                 className="hover:text-blue-600 transition font-medium"
119:                             >
120:                                 Login
121:                             </Link>
122:                             <Link
123:                                 to="/register"
124:                                 className="bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-blue-700

```

```

| transition font-medium"
125:         >
126:             Register
127:         </Link>
128:     </>
129: }}
130: </nav>
131:
132: { /* Mobile Hamburger */}
133: <button
134:     className="md:hidden text-gray-800 focus:outline-none"
135:     onClick={() => setSidebarOpen(true)}
136:     aria-label="Open menu"
137: >
138:     <svg className="w-6 h-6" fill="none" stroke="currentColor">
139:         <path
140:             strokeLinecap="round"
141:             strokeLinejoin="round"
142:             strokeWidth={2}
143:             d="M4 6h16M4 12h16M4 18h16"
144:         />
145:     </svg>
146: </button>
147: </div>
148: </header>
149:
150: { /* Sidebar overlay */}
151: {sidebarOpen && (
152:     <div
153:         className="fixed inset-0 bg-black bg-opacity-40 z-40"
154:         onClick={() => setSidebarOpen(false)}
155:     ></div>
156: )}
157:
158: { /* Sidebar */}
159: <div
160:     className={`fixed top-0 left-0 h-full w-64 bg-white shadow-lg z-50 transform trans
| ition-transform duration-300 ${
161:         sidebarOpen ? "translate-x-0" : "-translate-x-full"
162:     } md:hidden`}
163: >
164:     <div className="p-4 flex items-center justify-between border-b">
165:         <h2 className="text-lg font-bold text-blue-600">Menu</h2>
166:         <button
167:             className="text-gray-600 text-2xl"
168:             onClick={() => setSidebarOpen(false)}
169:             aria-label="Close menu"
170:         >
171:             x
172:         </button>
173:     </div>
174:     <div className="flex flex-col gap-4 p-4 text-sm">
175:         {user ? (
176:             <>
177:                 <div className="pb-3 border-b">
178:                     <p className="text-xs text-gray-500">Logged in as</p>
179:                     <p className="font-medium text-gray-800">{user.name}</p>
180:                 </div>
181:                 <Link
182:                     to="/home"
183:                     onClick={() => setSidebarOpen(false)}
184:                     className="hover:text-blue-600 py-2"
185:                 >
186:                     Home
187:                 </Link>
188:                 <Link
189:                     to="/register-item"
190:                     onClick={() => setSidebarOpen(false)}
191:                     className="hover:text-blue-600 py-2"
192:                 >
193:                     Register Item
194:                 </Link>
195:                 <Link
196:                     to="/lost-found-items"
197:                     onClick={() => setSidebarOpen(false)}
198:                     className="hover:text-blue-600 py-2"
199:                 >
200:                     Browse Items
201:                 </Link>
202:                 {user.role === "ADMIN" && (
203:                     <Link
204:                         to="/admin"
205:                         onClick={() => setSidebarOpen(false)}
206:                         className="hover:text-blue-600 py-2"
207:                     >

```

```

208:         Admin Dashboard
209:       </Link>
210:     )}
211:     <button
212:       onClick={() => {
213:         handleLogout();
214:         setSidebarOpen(false);
215:       }}
216:       className="text-left text-red-600 hover:underline py-2"
217:     >
218:       Logout
219:     </button>
220:   </>
221: ) : (
222:   <>
223:     <Link
224:       to="/"
225:       onClick={() => setSidebarOpen(false)}
226:       className="hover:text-blue-600 py-2"
227:     >
228:       Login
229:     </Link>
230:     <Link
231:       to="/register"
232:       onClick={() => setSidebarOpen(false)}
233:       className="hover:text-blue-600 py-2"
234:     >
235:       Register
236:     </Link>
237:   </>
238: )}
239: </div>
240: </div>
241: </>
242: );
243: }

```

■ File: src\components\ItemTable.js

```

1: // src/components/ItemTable.js
2: import React from "react";
3:
4: export default function ItemTable({ items }) {
5:   const formatDate = (dateString) => {
6:     if (!dateString) return "N/A";
7:     const date = new Date(dateString);
8:     return date.toLocaleDateString("en-US", {
9:       year: "numeric",
10:      month: "short",
11:      day: "numeric",
12:      hour: "2-digit",
13:      minute: "2-digit",
14:    });
15:   };
16:
17:   return (
18:     <div className="overflow-x-auto rounded-xl border border-gray-200">
19:       <table className="min-w-full table-auto">
20:         <thead>
21:           <tr className="bg-gray-100 text-left text-sm font-semibold text-gray-700">
22:             <th className="px-4 py-3">ITEM</th>
23:             <th className="px-4 py-3">STATUS</th>
24:             <th className="px-4 py-3">LOCATION</th>
25:             <th className="px-4 py-3">CREATED AT</th>
26:           </tr>
27:         </thead>
28:         <tbody>
29:           {items.length === 0 ? (
30:             <tr>
31:               <td colSpan={4} className="text-center py-8 text-gray-500">
32:                 No items registered yet.
33:               </td>
34:             </tr>
35:           ) : (
36:             items.map((item, idx) => (
37:               <tr
38:                 key={idx}
39:                 className="border-t text-sm text-gray-800 hover:bg-gray-50 transition"
40:               >
41:                 <td className="px-4 py-3 font-medium">{item.name}</td>
42:                 <td className="px-4 py-3">
43:                   <span
44:                     className={`inline-block px-2 py-1 rounded-full text-xs font-semibold

```



```

    | ${
45:         item.status === "FOUND"
46:         ? "bg-green-100 text-green-800"
47:         : item.status === "LOST"
48:         ? "bg-yellow-100 text-yellow-800"
49:         : "bg-blue-100 text-blue-800"
50:     }`
51:     >
52:     {item.status}
53:     </span>
54: </td>
55: <td className="px-4 py-3">{item.location}</td>
56: <td className="px-4 py-3 text-sm text-gray-600">
57:     {formatDate(item.createdAt)}
58: </td>
59: </tr>
60: ))
61: )}
62: </tbody>
63: </table>
64: </div>
65: );
66: }

```

■ File: src/components/MessageList.js

```

=====
1: // src/components/MessageList.js
2: import { useState } from "react";
3: import { API_BASE_URL, API_ENDPOINTS } from "../config/api";
4: import toast from "react-hot-toast";
5:
6: export default function MessageList({ messages }) {
7:   const [replies, setReplies] = useState({});
8:   const [sentReplies, setSentReplies] = useState({});
9:   const [sendingReply, setSendingReply] = useState({});
10:
11:   const handleReply = async (msgId, senderId, itemId) => {
12:     const replyText = replies[msgId]?.trim();
13:     if (!replyText) {
14:       toast.error("Please enter a message");
15:       return;
16:     }
17:
18:     if (replyText.length > 1000) {
19:       toast.error("Message cannot exceed 1000 characters");
20:       return;
21:     }
22:
23:     setSendingReply((prev) => ({ ...prev, [msgId]: true }));
24:
25:     try {
26:       const res = await fetch(`${API_BASE_URL}${API_ENDPOINTS.REPLY}`, {
27:         method: "POST",
28:         credentials: "include",
29:         headers: { "Content-Type": "application/json" },
30:         body: JSON.stringify({
31:           receiverId: senderId,
32:           itemId: itemId,
33:           message: replyText,
34:         }),
35:       });
36:
37:       const data = await res.json();
38:
39:       if (res.ok && data.success) {
40:         toast.success(data.message || "Reply sent!");
41:
42:         // Store the sent reply
43:         setSentReplies((prev) => ({
44:           ...prev,
45:           [msgId]: [
46:             ...(prev[msgId] || []),
47:             {
48:               text: replyText,
49:               timestamp: new Date().toLocaleString(),
50:             },
51:           ],
52:         }));
53:
54:         // Clear the input
55:         setReplies((prev) => ({ ...prev, [msgId]: "" }));
56:       } else {
57:         toast.error(data.message || "Failed to send reply.");

```

```

58:     }
59:   } catch (error) {
60:     console.error("Reply error:", error);
61:     toast.error("Server error while sending reply.");
62:   } finally {
63:     setSendingReply((prev) => ({ ...prev, [msgId]: false }));
64:   }
65: };
66:
67: const formatDate = (dateString) => {
68:   if (!dateString) return "N/A";
69:   const date = new Date(dateString);
70:   return date.toLocaleString("en-US", {
71:     year: "numeric",
72:     month: "short",
73:     day: "numeric",
74:     hour: "2-digit",
75:     minute: "2-digit",
76:   });
77: };
78:
79: if (!messages.length)
80:   return (
81:     <p className="text-gray-500 text-sm italic text-center py-8">
82:       No messages yet.
83:     </p>
84:   );
85:
86: return (
87:   <div className="space-y-6">
88:     {messages.map((msg) => (
89:       <div
90:         key={msg.id}
91:         className="bg-white border border-gray-200 rounded-2xl p-5 shadow-sm hover:shado
| w-md transition"
92:       >
93:         <div className="mb-3 flex flex-wrap gap-x-4 gap-y-1">
94:           <p className="text-sm text-gray-600">
95:             <span className="font-medium text-gray-700">From:</span>{" "}
96:             {msg.senderName}
97:           </p>
98:           <p className="text-sm text-gray-600">
99:             <span className="font-medium text-gray-700">Regarding:</span>{" "}
100:             {msg.itemName}
101:           </p>
102:         </div>
103:
104:         <div className="mb-4">
105:           <div className="bg-gray-50 border border-gray-100 rounded-xl p-3 text-gray-800
| text-sm leading-relaxed">
106:             {msg.message}
107:           </div>
108:           <div className="text-xs text-gray-400 mt-1">
109:             {formatDate(msg.sentAt)}
110:           </div>
111:         </div>
112:
113:         {sentReplies[msg.id]?.length > 0 && (
114:           <div className="mb-4 space-y-3">
115:             <div className="text-xs text-gray-500 font-medium uppercase tracking-wide">
116:               Your Replies
117:             </div>
118:             {sentReplies[msg.id].map((reply, index) => (
119:               <div key={index} className="flex justify-end">
120:                 <div className="max-w-xs lg:max-w-md">
121:                   <div className="bg-blue-600 text-white rounded-xl px-4 py-2 text-sm">
122:                     {reply.text}
123:                   </div>
124:                   <div className="text-xs text-gray-400 mt-1 text-right">
125:                     {reply.timestamp}
126:                   </div>
127:                 </div>
128:               </div>
129:             )))
130:           </div>
131:         )}
132:
133:         <textarea
134:           placeholder="Write your reply..."
135:           className="w-full border border-gray-300 rounded-xl px-4 py-2 text-sm focus:ou
| tline-none focus:ring-2 focus:ring-blue-500 transition resize-none"
136:           rows="3"
137:           value={replies[msg.id] || ""}
138:           onChange={(e) =>
139:             setReplies((prev) => ({ ...prev, [msg.id]: e.target.value }))

```

```

140:         }
141:         maxLength={1000}
142:       />
143:       <div className="flex justify-between items-center mt-2">
144:         <span className="text-xs text-gray-500">
145:           {(replies[msg.id] || []).length}/1000 characters
146:         </span>
147:         <button
148:           className="bg-blue-600 hover:bg-blue-700 text-white text-sm px-4 py-2 rounded
149: | d-xl transition disabled:opacity-50 disabled:cursor-not-allowed"
150:           onClick={() => handleReply(msg.id, msg.senderId, msg.itemId)}
151:           disabled={
152:             !replies[msg.id]??.trim() || sendingReply[msg.id]
153:           }
154:           >
155:             {sendingReply[msg.id] ? "Sending..." : "Send Reply"}
156:           </button>
157:         </div>
158:       )}
159:     </div>
160:   );
161: }

```

■ File: src\config\api.js

```

=====
1: // src/config/api.js
2: export const API_BASE_URL = process.env.REACT_APP_API_URL || 'http://localhost:8080';
3:
4: export const API_ENDPOINTS = {
5:   // Auth endpoints
6:   LOGIN: '/api/auth/login',
7:   REGISTER: '/api/auth/register',
8:   LOGOUT: '/api/auth/logout',
9:   REFRESH: '/api/auth/refresh',
10:  VALIDATE: '/api/auth/validate',
11:  ME: '/api/auth/me',
12:
13:  // Items endpoints
14:  ITEMS: '/items',
15:  ITEM_BY_ID: (id) => `/items/${id}`,
16:
17:  // Claims endpoints
18:  CLAIMS: '/claims',
19:  CLAIM_ITEM: (id) => `/claims/item/${id}`,
20:  CLAIM_BY_ID: (id) => `/claims/${id}`,
21:
22:  // Messages endpoints
23:  MESSAGES: '/messages',
24:  MESSAGES_SENT: '/messages/sent',
25:  REPLY: '/messages/reply',
26:  MESSAGE_BY_ID: (id) => `/messages/${id}`,
27:
28:  // Feedback endpoint
29:  FEEDBACK: '/feedback',
30:
31:  // Dashboard endpoint
32:  DASHBOARD: '/dashboard',
33:
34:  // Admin endpoints
35:  ADMIN_DASHBOARD: '/admin/dashboard',
36:  ADMIN_ITEMS: '/admin/items',
37:  ADMIN_CLAIMS: '/admin/claims',
38:  ADMIN_USERS: '/admin/users',
39:  ADMIN_FEEDBACK: '/admin/feedback',
40:  ADMIN_DELETE_ITEM: (id) => `/admin/items/${id}`,
41:  ADMIN_DELETE_CLAIM: (id) => `/admin/claims/${id}`,
42:  ADMIN_DELETE_USER: (id) => `/admin/users/${id}`,
43:  ADMIN_DELETE_FEEDBACK: (id) => `/admin/feedback/${id}`,
44: };
45:
46: // Helper function to make API calls with proper error handling
47: export const apiCall = async (endpoint, options = {}) => {
48:   const defaultOptions = {
49:     credentials: 'include', // Always include cookies
50:     headers: {
51:       'Content-Type': 'application/json',
52:       ...options.headers,
53:     },
54:   };
55:
56:   // Merge options
57:   const finalOptions = { ...defaultOptions, ...options };

```

```

58:
59: // Remove Content-Type for FormData
60: if (options.body instanceof FormData) {
61:   delete finalOptions.headers['Content-Type'];
62: }
63:
64: try {
65:   const response = await fetch(`${API_BASE_URL}${endpoint}`, finalOptions);
66:
67:   // Handle token refresh if needed (401 Unauthorized)
68:   if (response.status === 401 && endpoint !== API_ENDPOINTS.REFRESH) {
69:     // Try to refresh token
70:     const refreshResponse = await fetch(`${API_BASE_URL}${API_ENDPOINTS.REFRESH}`, {
71:       method: 'POST',
72:       credentials: 'include',
73:     });
74:
75:     if (refreshResponse.ok) {
76:       // Retry original request
77:       return fetch(`${API_BASE_URL}${endpoint}`, finalOptions);
78:     } else {
79:       // Refresh failed, redirect to login
80:       window.location.href = '/';
81:       throw new Error('Session expired');
82:     }
83:   }
84:
85:   return response;
86: } catch (error) {
87:   console.error('API call error:', error);
88:   throw error;
89: }
90: };

```

■ File: src\index.css

```

1: @tailwind base;
2: @tailwind components;
3: @tailwind utilities;
4:
5:
6:

```

■ File: src\index.js

```

1: import React from 'react';
2: import ReactDOM from 'react-dom/client';
3: import './index.css';
4: import App from './App';
5: import reportWebVitals from './reportWebVitals';
6:
7: const root = ReactDOM.createRoot(document.getElementById('root'));
8: root.render(
9:   <React.StrictMode>
10:     <App />
11:   </React.StrictMode>
12: );
13:
14: // If you want to start measuring performance in your app, pass a function
15: // to log results (for example: reportWebVitals(console.log))
16: // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
17: reportWebVitals();

```

■ File: src\pages\AdminPage.js

```

1: // src/pages/AdminPage.js
2: import { useEffect, useState } from "react";
3: import { useNavigate } from "react-router-dom";
4: import { API_BASE_URL, API_ENDPOINTS } from "../config/api";
5: import AdminTable from "../components/AdminTable";
6: import toast, { Toaster } from "react-hot-toast";
7: import Swal from "sweetalert2";
8: import withReactContent from "sweetalert2-react-content";
9:
10: const MySwal = withReactContent(Swal);
11:
12: export default function AdminPage() {
13:   const navigate = useNavigate();
14:   const [loading, setLoading] = useState(true);

```

```

15: const [items, setItems] = useState([]);
16: const [claims, setClaims] = useState([]);
17: const [users, setUsers] = useState([]);
18: const [feedback, setFeedback] = useState([]);
19: const [stats, setStats] = useState({});
20:
21: // Validate Admin Access
22: const validateAdminAccess = async () => {
23:   try {
24:     const res = await fetch(`${API_BASE_URL}${API_ENDPOINTS.ME}`, {
25:       credentials: "include",
26:     });
27:
28:     if (!res.ok) {
29:       toast.error("Authentication required");
30:       navigate("/");
31:       return false;
32:     }
33:
34:     const response = await res.json();
35:     const user = response.data;
36:
37:     if (!user || user.role !== "ADMIN") {
38:       toast.error("Access denied. Admin privileges required.");
39:       navigate("/");
40:       return false;
41:     }
42:     return true;
43:   } catch (err) {
44:     toast.error("Error validating admin access.");
45:     navigate("/");
46:     return false;
47:   }
48: };
49:
50: // Fetch Admin Dashboard Data
51: const fetchAdminData = async () => {
52:   try {
53:     const res = await fetch(`${API_BASE_URL}${API_ENDPOINTS.ADMIN_DASHBOARD}`, {
54:       credentials: "include",
55:     });
56:
57:     if (!res.ok) {
58:       if (res.status === 403) {
59:         toast.error("Access denied");
60:         navigate("/");
61:       }
62:       throw new Error("Failed to load dashboard");
63:     }
64:
65:     const data = await res.json();
66:
67:     if (data.success) {
68:       setItems(data.items || []);
69:       setClaims(data.claims || []);
70:       setUsers(data.users || []);
71:       setFeedback(data.feedback || []);
72:       setStats(data.stats || {});
73:     }
74:   } catch (err) {
75:     toast.error("Failed to load admin data.");
76:     console.error(err);
77:   } finally {
78:     setLoading(false);
79:   }
80: };
81:
82: useEffect(() => {
83:   const init = async () => {
84:     const isAdmin = await validateAdminAccess();
85:     if (isAdmin) await fetchAdminData();
86:   };
87:   init();
88: }, []);
89:
90: // Delete Handler with Confirmation
91: const handleDelete = async (type, id) => {
92:   const confirmed = await MySwal.fire({
93:     title: `Delete this ${type}?`,
94:     text: `This action cannot be undone.`,
95:     icon: "warning",
96:     showCancelButton: true,
97:     confirmButtonColor: "#d333",
98:     cancelButtonColor: "#3085d6",
99:     confirmButtonText: "Yes, delete it!",

```

```

100:         cancelButtonText: "Cancel",
101:     });
102:
103:     if (!confirmed.isConfirmed) return;
104:
105:     let endpoint;
106:     switch (type) {
107:         case "item":
108:             endpoint = API_ENDPOINTS.ADMIN_DELETE_ITEM(id);
109:             break;
110:         case "claim":
111:             endpoint = API_ENDPOINTS.ADMIN_DELETE_CLAIM(id);
112:             break;
113:         case "user":
114:             endpoint = API_ENDPOINTS.ADMIN_DELETE_USER(id);
115:             break;
116:         case "feedback":
117:             endpoint = API_ENDPOINTS.ADMIN_DELETE_FEEDBACK(id);
118:             break;
119:         default:
120:             return;
121:     }
122:
123:     try {
124:         const res = await fetch(`${API_BASE_URL}${endpoint}`, {
125:             method: "DELETE",
126:             credentials: "include",
127:         });
128:
129:         const data = await res.json();
130:
131:         if (res.ok && data.success) {
132:             toast.success(data.message || `${type} deleted successfully.`);
133:             await fetchAdminData(); // Refresh data
134:         } else {
135:             toast.error(data.message || `Failed to delete ${type}.`);
136:         }
137:     } catch (err) {
138:         console.error(err);
139:         toast.error(`Error deleting ${type}.`);
140:     }
141: };
142:
143: const formatDate = (dateString) => {
144:     if (!dateString) return "N/A";
145:     const date = new Date(dateString);
146:     return date.toLocaleDateString("en-US", {
147:         year: "numeric",
148:         month: "short",
149:         day: "numeric",
150:     });
151: };
152:
153: if (loading) {
154:     return (
155:         <div className="min-h-screen flex items-center justify-center">
156:             <div className="text-center">
157:                 <div className="animate-spin rounded-full h-12 w-12 border-b-2 border-blue-600 m
| x-auto mb-4"></div>
158:                 <p className="text-gray-600">Verifying admin access...</p>
159:             </div>
160:         </div>
161:     );
162: }
163:
164: return (
165:     <div className="max-w-7xl mx-auto px-6 py-8 space-y-8">
166:         <Toaster position="top-center" />
167:
168:         <div className="text-center">
169:             <h1 className="text-3xl font-bold text-blue-600 mb-2">
Admin Dashboard
170:             </h1>
171:             <p className="text-gray-600">Manage all platform activities</p>
172:         </div>
173:
174:         <div>
175:             <div>
176:                 <div>
177:                     <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-4">
178:                         <div className="bg-blue-50 border border-blue-200 rounded-xl p-4">
179:                             <h3 className="text-sm font-medium text-blue-600">Total Items</h3>
180:                             <p className="text-2xl font-bold text-blue-700 mt-1">
181:                                 {stats.totalItems || 0}
182:                             </p>
183:                         </div>

```

```

184:         <div className="bg-green-50 border border-green-200 rounded-xl p-4">
185:             <h3 className="text-sm font-medium text-green-600">Total Claims</h3>
186:             <p className="text-2xl font-bold text-green-700 mt-1">
187:                 {stats.totalClaims || 0}
188:             </p>
189:         </div>
190:         <div className="bg-purple-50 border border-purple-200 rounded-xl p-4">
191:             <h3 className="text-sm font-medium text-purple-600">Total Users</h3>
192:             <p className="text-2xl font-bold text-purple-700 mt-1">
193:                 {stats.totalUsers || 0}
194:             </p>
195:         </div>
196:         <div className="bg-orange-50 border border-orange-200 rounded-xl p-4">
197:             <h3 className="text-sm font-medium text-orange-600">Total Feedback</h3>
198:             <p className="text-2xl font-bold text-orange-700 mt-1">
199:                 {stats.totalFeedback || 0}
200:             </p>
201:         </div>
202:     </div>
203: )}
204:
205: { /* Items Table */}
206: <AdminTable
207:     title="Registered Items"
208:     type="item"
209:     data={items}
210:     headers={[ "ID", "Name", "Location", "Status", "Created By", "Actions" ]}
211:     rowRenderer={(item) => [
212:         item.id,
213:         item.name,
214:         item.location,
215:         <span
216:             className={`inline-block px-2 py-1 rounded-full text-xs font-medium ${
217:                 item.status === "FOUND"
218:                     ? "bg-green-100 text-green-700"
219:                     : item.status === "LOST"
220:                     ? "bg-yellow-100 text-yellow-700"
221:                     : "bg-blue-100 text-blue-700"
222:             }`}
223:         >
224:             {item.status}
225:         </span>,
226:         item.creatorName,
227:         <button
228:             onClick={() => handleDelete("item", item.id)}
229:             className="text-red-600 hover:text-red-800 text-sm font-medium"
230:         >
231:             Delete
232:         </button>,
233:     ]}
234: />
235:
236: { /* Claims Table */}
237: <AdminTable
238:     title="Claims"
239:     type="claim"
240:     data={claims}
241:     headers={[ "ID", "Item", "Claimant", "Email", "Claimed At", "Actions" ]}
242:     rowRenderer={(claim) => [
243:         claim.id,
244:         claim.itemName,
245:         claim.claimantName,
246:         claim.claimantEmail,
247:         formatDate(claim.claimedAt),
248:         <button
249:             onClick={() => handleDelete("claim", claim.id)}
250:             className="text-red-600 hover:text-red-800 text-sm font-medium"
251:         >
252:             Delete
253:         </button>,
254:     ]}
255: />
256:
257: { /* Users Table */}
258: <AdminTable
259:     title="Users"
260:     type="user"
261:     data={users}
262:     headers={[ "ID", "Name", "Email", "Role", "Registered", "Actions" ]}
263:     rowRenderer={(user) => [
264:         user.id,
265:         user.name,
266:         user.email,
267:         <span
268:             className={`inline-block px-2 py-1 rounded-full text-xs font-medium ${

```

```

269:         user.role === "ADMIN"
270:         ? "bg-red-100 text-red-700"
271:         : "bg-gray-100 text-gray-700"
272:     }` }
273: >
274:     {user.role}
275: </span>,
276:     formatDate(user.createdAt),
277:     user.role === "ADMIN" ? (
278:         <span className="text-gray-400 text-sm">Protected</span>
279:     ) : (
280:         <button
281:             onClick={() => handleDelete("user", user.id)}
282:             className="text-red-600 hover:text-red-800 text-sm font-medium"
283:         >
284:             Delete
285:         </button>
286:     ),
287: ]}
288: />
289:
290: { /* Feedback Table */ }
291: <AdminTable
292:     title="Feedback"
293:     type="feedback"
294:     data={feedback}
295:     headers={["ID", "User", "Feedback", "Submitted", "Actions"]}
296:     rowRenderer={(fb) => [
297:         fb.id,
298:         fb.userName || "Anonymous",
299:         <div className="max-w-md truncate">{fb.feedbackText}</div>,
300:         formatDate(fb.submittedAt),
301:         <button
302:             onClick={() => handleDelete("feedback", fb.id)}
303:             className="text-red-600 hover:text-red-800 text-sm font-medium"
304:         >
305:             Delete
306:         </button>,
307:     ]}
308: />
309: </div>
310: );
311: }

```

■ File: src\pages\HomePage.js

```

=====
1: // src/pages/HomePage.js
2: import { useEffect, useState } from "react";
3: import { useNavigate } from "react-router-dom";
4: import { API_BASE_URL, API_ENDPOINTS } from "../config/api";
5: import ItemTable from "../components/ItemTable";
6: import MessageList from "../components/MessageList";
7: import ClaimsTable from "../components/ClaimsTable";
8: import toast, { Toaster } from "react-hot-toast";
9:
10: export default function HomePage() {
11:     const navigate = useNavigate();
12:     const [name, setName] = useState("");
13:     const [role, setRole] = useState("");
14:     const [items, setItems] = useState([]);
15:     const [claims, setClaims] = useState([]);
16:     const [messages, setMessages] = useState([]);
17:     const [loading, setLoading] = useState(true);
18:     const [feedbackText, setFeedbackText] = useState("");
19:     const [submittingFeedback, setSubmittingFeedback] = useState(false);
20:
21:     useEffect(() => {
22:         fetchHomeData();
23:     }, [navigate]);
24:
25:     const fetchHomeData = async () => {
26:         try {
27:             const res = await fetch(`${API_BASE_URL}${API_ENDPOINTS.DASHBOARD}`, {
28:                 method: "GET",
29:                 credentials: "include", // Send cookies
30:             });
31:
32:             if (!res.ok) {
33:                 navigate("/");
34:                 return;
35:             }
36:
37:             const data = await res.json();

```



```

38:
39:     setName(data.user.name);
40:     setRole(data.user.role);
41:     setItems(data.items || []);
42:     setClaims(data.claims || []);
43:     setMessages(data.messages || []);
44:   } catch (err) {
45:     console.error("Failed to fetch dashboard:", err);
46:     navigate("/");
47:   } finally {
48:     setLoading(false);
49:   }
50: };
51:
52: const handleFeedbackSubmit = async (e) => {
53:   e.preventDefault();
54:
55:   if (!feedbackText.trim()) {
56:     toast.error("Please enter your feedback");
57:     return;
58:   }
59:
60:   if (feedbackText.trim().length < 10) {
61:     toast.error("Feedback must be at least 10 characters long");
62:     return;
63:   }
64:
65:   if (feedbackText.trim().length > 2000) {
66:     toast.error("Feedback cannot exceed 2000 characters");
67:     return;
68:   }
69:
70:   setSubmittingFeedback(true);
71:
72:   try {
73:     const res = await fetch(`${API_BASE_URL}${API_ENDPOINTS.FEEDBACK}`, {
74:       method: "POST",
75:       credentials: "include", // Send cookies
76:       headers: {
77:         "Content-Type": "application/json",
78:       },
79:       body: JSON.stringify({ feedback: feedbackText.trim() }),
80:     });
81:
82:     const data = await res.json();
83:
84:     if (res.ok && data.success) {
85:       toast.success(data.message || "Thank you for your feedback!");
86:       setFeedbackText("");
87:     } else {
88:       toast.error(data.message || "Failed to submit feedback.");
89:     }
90:   } catch (err) {
91:     console.error(err);
92:     toast.error("Something went wrong while submitting feedback.");
93:   } finally {
94:     setSubmittingFeedback(false);
95:   }
96: };
97:
98: if (loading) {
99:   return (
100:     <div className="min-h-screen flex items-center justify-center text-gray-600">
101:       <div className="text-center">
102:         <div className="animate-spin rounded-full h-12 w-12 border-b-2 border-blue-600 m
| x-auto mb-4"></div>
103:         <p>Loading your dashboard...</p>
104:       </div>
105:     </div>
106:   );
107: }
108:
109: return (
110:   <div className="p-6 max-w-6xl mx-auto space-y-10 pb-20">
111:     <Toaster position="top-center" />
112:
113:     <div className="text-center">
114:       <h1 className="text-3xl font-bold text-gray-800">Welcome, {name}</h1>
115:       <p className="text-sm text-gray-500 mt-1">
116:         Here's an overview of your account and activity.
117:       </p>
118:     </div>
119:
120:     { /* My Registered Items */ }
121:     <section className="bg-white p-6 rounded-2xl shadow-sm border border-gray-200">

```

```

122:         <h2 className="text-xl font-semibold text-gray-700 mb-4 border-b pb-2">
123:             My Registered Items
124:         </h2>
125:         <ItemTable items={items} />
126:     </section>
127:
128:     { /* My Claims */ }
129:     <section className="bg-white p-6 rounded-2xl shadow-sm border border-gray-200">
130:         <h2 className="text-xl font-semibold text-gray-700 mb-4 border-b pb-2">
131:             My Claims
132:         </h2>
133:         <ClaimsTable claims={claims} />
134:     </section>
135:
136:     { /* Messages */ }
137:     <section className="bg-white p-6 rounded-2xl shadow-sm border border-gray-200">
138:         <h2 className="text-xl font-semibold text-gray-700 mb-4 border-b pb-2">
139:             Messages
140:         </h2>
141:         <MessageList messages={messages} />
142:     </section>
143:
144:     { /* Platform Feedback - Only for non-admin users */ }
145:     {role !== "ADMIN" && (
146:         <section className="bg-white p-6 rounded-2xl shadow-sm border border-gray-200">
147:             <h2 className="text-xl font-semibold text-gray-700 mb-4 border-b pb-2">
148:                 Platform Feedback
149:             </h2>
150:             <form onSubmit={handleFeedbackSubmit} className="space-y-4">
151:                 <textarea
152:                     rows={4}
153:                     className="w-full px-4 py-3 border border-gray-300 rounded-xl focus:outline-
| none focus:ring-2 focus:ring-blue-500 resize-none"
154:                     placeholder="Share your experience with us... (minimum 10 characters)"
155:                     value={feedbackText}
156:                     onChange={(e) => setFeedbackText(e.target.value)}
157:                     required
158:                     maxLength={2000}
159:                 />
160:                 <div className="flex justify-between items-center">
161:                     <span className="text-xs text-gray-500">
162:                         {feedbackText.length}/2000 characters
163:                     </span>
164:                     <button
165:                         type="submit"
166:                         disabled={submittingFeedback || feedbackText.trim().length < 10}
167:                         className="bg-blue-600 text-white px-6 py-2 rounded-xl hover:bg-blue-700 t
| ransition disabled:opacity-50 disabled:cursor-not-allowed"
168:                     >
169:                         {submittingFeedback ? "Submitting..." : "Submit Feedback"}
170:                     </button>
171:                 </div>
172:             </form>
173:         </section>
174:     )}
175: </div>
176: );
177: }

```

■ File: src\pages\ItemRegisterPage.js

```

=====
1: // src/pages/ItemRegisterPage.js
2: import { useState } from "react";
3: import { useNavigate } from "react-router-dom";
4: import { CameraIcon, Loader2, X } from "lucide-react";
5: import { API_BASE_URL, API_ENDPOINTS } from "../config/api";
6: import toast, { Toaster } from "react-hot-toast";
7:
8: export default function RegisterItemPage() {
9:     const navigate = useNavigate();
10:    const [formData, setFormData] = useState({
11:        name: "",
12:        description: "",
13:        location: "",
14:        status: "LOST", // Backend expects uppercase: LOST or FOUND
15:    });
16:    const [image, setImage] = useState(null);
17:    const [previewUrl, setPreviewUrl] = useState(null);
18:    const [loading, setLoading] = useState(false);
19:
20:    const handleChange = (e) => {
21:        setFormData((prev) => ({ ...prev, [e.target.name]: e.target.value }));
22:    };

```

```

23:
24: const handleFileChange = (e) => {
25:   const file = e.target.files[0];
26:   if (file) {
27:     // Validate file size (5MB max)
28:     if (file.size > 5 * 1024 * 1024) {
29:       toast.error("Image size must not exceed 5MB");
30:       return;
31:     }
32:
33:     // Validate file type
34:     if (!file.type.startsWith("image/")) {
35:       toast.error("Only image files are allowed");
36:       return;
37:     }
38:
39:     setImage(file);
40:     setPreviewUrl(URL.createObjectURL(file));
41:   }
42: };
43:
44: const clearImage = () => {
45:   setImage(null);
46:   setPreviewUrl(null);
47: };
48:
49: const handleSubmit = async (e) => {
50:   e.preventDefault();
51:
52:   if (!image) {
53:     toast.error("Please select an image file.");
54:     return;
55:   }
56:
57:   setLoading(true);
58:
59:   const data = new FormData();
60:   data.append("name", formData.name.trim());
61:   data.append("description", formData.description.trim());
62:   data.append("location", formData.location.trim());
63:   data.append("status", formData.status);
64:   data.append("image", image);
65:
66:   try {
67:     const res = await fetch(`${API_BASE_URL}${API_ENDPOINTS.ITEMS}`, {
68:       method: "POST",
69:       body: data,
70:       credentials: "include", // Send cookies
71:       // Don't set Content-Type header for FormData - browser will set it with boundary
72:     });
73:
74:     const result = await res.json();
75:
76:     if (res.ok && result.success) {
77:       toast.success(result.message || "Item registered successfully!");
78:       setTimeout(() => navigate("/lost-found-items"), 2000);
79:     } else {
80:       // Handle validation errors
81:       if (result.errors) {
82:         Object.values(result.errors).forEach((error) => toast.error(error));
83:       } else {
84:         toast.error(result.message || "Failed to register item.");
85:       }
86:     }
87:   } catch (err) {
88:     console.error(err);
89:     toast.error("Something went wrong. Please try again.");
90:   } finally {
91:     setLoading(false);
92:   }
93: };
94:
95: return (
96:   <div className="max-w-3xl mx-auto px-4 py-10">
97:     <Toaster position="top-center" />
98:
99:     <div className="bg-white shadow-xl rounded-2xl p-8 border border-gray-200">
100:       <div className="text-center mb-6">
101:         <h2 className="text-3xl font-bold text-blue-600 flex justify-center items-center gap-2">
102:           <CameraIcon className="w-7 h-7" />
103:           Register Lost / Found Item
104:         </h2>
105:         <p className="text-sm text-gray-500 mt-2">
106:           Submit the details to help match items with their rightful owners.

```

```

107:         </p>
108:     </div>
109:
110:     <form onSubmit={handleSubmit} className="space-y-5">
111:         { /* Item Name */ }
112:         <div>
113:             <label className="block text-sm font-medium text-gray-700 mb-1">
114:                 Item Name <span className="text-red-500">*</span>
115:             </label>
116:             <input
117:                 type="text"
118:                 name="name"
119:                 value={formData.name}
120:                 onChange={handleChange}
121:                 required
122:                 className="w-full border border-gray-300 px-4 py-2 rounded-lg focus:ring-2 f
|ocus:ring-blue-500 focus:outline-none focus:border-transparent"
123:                 placeholder="e.g., Black Wallet, iPhone 13"
124:             />
125:         </div>
126:
127:         { /* Description */ }
128:         <div>
129:             <label className="block text-sm font-medium text-gray-700 mb-1">
130:                 Description <span className="text-red-500">*</span>
131:             </label>
132:             <textarea
133:                 name="description"
134:                 value={formData.description}
135:                 onChange={handleChange}
136:                 required
137:                 rows={4}
138:                 className="w-full border border-gray-300 px-4 py-2 rounded-lg focus:ring-2 f
|ocus:ring-blue-500 focus:outline-none focus:border-transparent resize-none"
139:                 placeholder="Provide details like color, brand, distinctive features, conten
|ts..."
140:             />
141:         </div>
142:
143:         { /* Location */ }
144:         <div>
145:             <label className="block text-sm font-medium text-gray-700 mb-1">
146:                 Location <span className="text-red-500">*</span>
147:             </label>
148:             <input
149:                 type="text"
150:                 name="location"
151:                 value={formData.location}
152:                 onChange={handleChange}
153:                 required
154:                 className="w-full border border-gray-300 px-4 py-2 rounded-lg focus:ring-2 f
|ocus:ring-blue-500 focus:outline-none focus:border-transparent"
155:                 placeholder="Where was it found/lost? e.g., Library 2nd Floor, Main Campus"
156:             />
157:         </div>
158:
159:         { /* Status */ }
160:         <div>
161:             <label className="block text-sm font-medium text-gray-700 mb-1">
162:                 Status <span className="text-red-500">*</span>
163:             </label>
164:             <select
165:                 name="status"
166:                 value={formData.status}
167:                 onChange={handleChange}
168:                 className="w-full border border-gray-300 px-4 py-2 rounded-lg focus:ring-2 f
|ocus:ring-blue-500 focus:outline-none focus:border-transparent"
169:             >
170:                 <option value="LOST">Lost - I lost this item</option>
171:                 <option value="FOUND">Found - I found this item</option>
172:             </select>
173:         </div>
174:
175:         { /* Image Upload */ }
176:         <div>
177:             <label className="block text-sm font-medium text-gray-700 mb-1">
178:                 Upload Image <span className="text-red-500">*</span>
179:                 <span className="text-xs text-gray-500 ml-2">(Max 5MB)</span>
180:             </label>
181:             <input
182:                 type="file"
183:                 accept="image/*"
184:                 onChange={handleFileChange}
185:                 required={!image}
186:                 className="w-full text-sm file:mr-4 file:py-2 file:px-4 file:rounded-lg file

```

```

| :border-0 file:text-sm file:font-medium file:bg-blue-50 file:text-blue-700 hover:file:bg-b
| lue-100"
187:     />
188:   </div>
189:
190:   {/* Preview */}
191:   {previewUrl && (
192:     <div className="relative">
193:       <label className="block text-sm font-medium text-gray-700 mb-2">
194:         Preview
195:       </label>
196:       <div className="relative">
197:         <img
198:           src={previewUrl}
199:           alt="Preview"
200:           className="w-full max-h-[400px] object-contain border border-gray-300 ro
| unded-lg shadow"
201:         />
202:         <button
203:           type="button"
204:           onClick={clearImage}
205:           className="absolute top-2 right-2 bg-red-500 text-white p-1 rounded-full
| hover:bg-red-600 transition"
206:           aria-label="Remove image"
207:         >
208:           <X className="w-4 h-4" />
209:         </button>
210:       </div>
211:     </div>
212:   )}
213:
214:   {/* Submit Button */}
215:   <div className="pt-4">
216:     <button
217:       type="submit"
218:       disabled={loading}
219:       className={`w-full flex justify-center items-center gap-2 bg-blue-600 hover:
| bg-blue-700 text-white font-semibold py-3 rounded-lg transition duration-200 ${
220:         loading ? "opacity-70 cursor-not-allowed" : ""
221:       }`}
222:     >
223:       {loading ? (
224:         <>
225:           <Loader2 className="animate-spin w-5 h-5" />
226:           Submitting...
227:         </>
228:       ) : (
229:         <>
230:           <CameraIcon className="w-5 h-5" />
231:           Register Item
232:         </>
233:       )}
234:     </button>
235:   </div>
236: </form>
237:
238: <div className="mt-6 text-center">
239:   <button
240:     onClick={() => navigate("/lost-found-items")}
241:     className="text-sm text-blue-600 hover:underline"
242:   >
243:     ? Back to Browse Items
244:   </button>
245: </div>
246: </div>
247: </div>
248: );
249: }

```

■ File: src\pages\ItemsPage.js

```

=====
1: // src/pages/ItemsPage.js
2: import { useEffect, useState } from "react";
3: import { API_BASE_URL, API_ENDPOINTS } from "../config/api";
4: import toast, { Toaster } from "react-hot-toast";
5:
6: export default function ItemsPage() {
7:   const [items, setItems] = useState([]);
8:   const [search, setSearch] = useState("");
9:   const [filter, setFilter] = useState(""); // Backend expects "" for all, "LOST", "FOUND"
| , or "CLAIMED"
10:   const [currentUser, setCurrentUser] = useState(null);
11:   const [loading, setLoading] = useState(true);

```

```

12:
13: const [showMessageBox, setShowMessageBox] = useState(false);
14: const [showClaimModal, setShowClaimModal] = useState(false);
15: const [currentItem, setCurrentItem] = useState(null);
16: const [messageText, setMessageText] = useState("");
17: const [sendingMessage, setSendingMessage] = useState(false);
18: const [claimingItem, setClaimingItem] = useState(false);
19:
20: useEffect(() => {
21:   fetchUser();
22:   fetchItems();
23: }, []);
24:
25: useEffect(() => {
26:   const interval = setInterval(() => {
27:     fetchItems();
28:   }, 60000); // Refresh every minute
29:   return () => clearInterval(interval);
30: }, []);
31:
32: const fetchUser = async () => {
33:   try {
34:     const res = await fetch(`${API_BASE_URL}${API_ENDPOINTS.ME}`, {
35:       credentials: "include",
36:     });
37:     if (res.ok) {
38:       const data = await res.json();
39:       setCurrentUser(data.data); // Backend returns data in 'data' field
40:     }
41:   } catch (err) {
42:     console.error("Failed to fetch user");
43:   }
44: };
45:
46: const fetchItems = async () => {
47:   try {
48:     const params = new URLSearchParams();
49:     if (search) params.append("search", search);
50:     if (filter) params.append("status", filter);
51:
52:     const res = await fetch(
53:       `${API_BASE_URL}${API_ENDPOINTS.ITEMS}?${params.toString()}`,
54:       {
55:         credentials: "include",
56:       }
57:     );
58:
59:     if (!res.ok) throw new Error("Failed to fetch items");
60:
61:     const data = await res.json();
62:     setItems(data.items || []);
63:   } catch (err) {
64:     console.error(err);
65:     toast.error("Failed to load items");
66:   } finally {
67:     setLoading(false);
68:   }
69: };
70:
71: // Trigger search when search or filter changes
72: useEffect(() => {
73:   const debounce = setTimeout(() => {
74:     fetchItems();
75:   }, 500);
76:   return () => clearTimeout(debounce);
77: }, [search, filter]);
78:
79: const handleClaim = async (itemId) => {
80:   setClaimingItem(true);
81:   try {
82:     const res = await fetch(
83:       `${API_BASE_URL}${API_ENDPOINTS.CLAIM_ITEM(itemId)}`,
84:       {
85:         method: "POST",
86:         credentials: "include",
87:       }
88:     );
89:
90:     const data = await res.json();
91:
92:     if (res.ok && data.success) {
93:       toast.success(data.message || "Item claimed successfully!");
94:       setShowClaimModal(false);
95:       fetchItems(); // Refresh items list
96:     } else {

```

```

97:         toast.error(data.message || "Failed to claim item.");
98:     }
99:     } catch (err) {
100:         console.error("Claim failed", err);
101:         toast.error("Something went wrong.");
102:     } finally {
103:         setClaimingItem(false);
104:     }
105: };
106:
107: const handleSendMessage = async () => {
108:     if (!currentItem || !messageText.trim()) {
109:         toast.error("Please enter a message");
110:         return;
111:     }
112:
113:     if (messageText.trim().length > 1000) {
114:         toast.error("Message cannot exceed 1000 characters");
115:         return;
116:     }
117:
118:     setSendingMessage(true);
119:
120:     try {
121:         const res = await fetch(`${API_BASE_URL}${API_ENDPOINTS.MESSAGES}`, {
122:             method: "POST",
123:             headers: { "Content-Type": "application/json" },
124:             credentials: "include",
125:             body: JSON.stringify({
126:                 receiverId: currentItem.createdBy,
127:                 itemId: currentItem.id,
128:                 message: messageText.trim(),
129:             }),
130:         });
131:
132:         const data = await res.json();
133:
134:         if (res.ok && data.success) {
135:             toast.success(data.message || "Message sent!");
136:             setShowMessageBox(false);
137:             setMessageText("");
138:             setCurrentItem(null);
139:         } else {
140:             toast.error(data.message || "Failed to send message.");
141:         }
142:     } catch (err) {
143:         console.error("Message failed", err);
144:         toast.error("Something went wrong.");
145:     } finally {
146:         setSendingMessage(false);
147:     }
148: };
149:
150: const formatDate = (dateString) => {
151:     if (!dateString) return "N/A";
152:     const date = new Date(dateString);
153:     return date.toLocaleDateString("en-US", {
154:         year: "numeric",
155:         month: "short",
156:         day: "numeric",
157:         hour: "2-digit",
158:         minute: "2-digit",
159:     });
160: };
161:
162: if (loading) {
163:     return (
164:         <div className="min-h-screen flex items-center justify-center">
165:             <div className="text-center">
166:                 <div className="animate-spin rounded-full h-12 w-12 border-b-2 border-blue-600 m
| x-auto mb-4"></div>
167:                 <p className="text-gray-600">Loading items...</p>
168:             </div>
169:         </div>
170:     );
171: }
172:
173: return (
174:     <div className="max-w-6xl mx-auto p-6 space-y-6">
175:         <Toaster position="top-center" reverseOrder={false} />
176:
177:         <h1 className="text-3xl font-bold text-blue-600 text-center">
178:             Lost and Found Items
179:         </h1>
180:

```

```

181:      {/* Search and Filter */}
182:      <div className="flex flex-col sm:flex-row gap-4 justify-between items-center">
183:        <input
184:          type="text"
185:          placeholder="Search by name, location, or description..."
186:          className="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-none
| focus:ring-2 focus:ring-blue-500"
187:          value={search}
188:          onChange={(e) => setSearch(e.target.value)}
189:        />
190:        <select
191:          value={filter}
192:          onChange={(e) => setFilter(e.target.value)}
193:          className="px-4 py-2 border border-gray-300 rounded-lg focus:outline-none focus:
| ring-2 focus:ring-blue-500 min-w-[150px]"
194:        >
195:          <option value="">All Items</option>
196:          <option value="LOST">Lost</option>
197:          <option value="FOUND">Found</option>
198:          <option value="CLAIMED">Claimed</option>
199:        </select>
200:      </div>
201:
202:      {/* Items Grid */}
203:      <div className="grid gap-6 grid-cols-1 sm:grid-cols-2 lg:grid-cols-3">
204:        {items.map((item) => (
205:          <div
206:            key={item.id}
207:            className="border border-gray-200 rounded-xl overflow-hidden shadow-sm hover:s
| hadow-md transition bg-white flex flex-col"
208:          >
209:            {item.image && (
210:              <img
211:                src={`${API_BASE_URL}/${item.image}`}
212:                alt={item.name}
213:                className="w-full h-48 object-cover"
214:              />
215:            )}
216:            <div className="p-4 space-y-2 flex-1 flex flex-col justify-between">
217:              <div>
218:                <h2 className="text-lg font-semibold text-gray-800">{item.name}</h2>
219:                <p className="text-sm text-gray-500 flex items-center gap-1">
220:                  <span>?</span> {item.location}
221:                </p>
222:                <p className="text-sm text-gray-600 mt-2 line-clamp-2">
223:                  {item.description}
224:                </p>
225:              </div>
226:
227:              <div className="text-xs text-gray-500 mt-2 space-y-1">
228:                <p>
229:                  Posted by: <strong>{item.creatorName}</strong>
230:                </p>
231:                <p>On: {formatDate(item.createdAt)}</p>
232:              </div>
233:
234:              <div className="mt-2">
235:                <span
236:                  className={`inline-block text-xs px-3 py-1 rounded-full font-medium ${
237:                    item.status === "FOUND"
238:                      ? "bg-green-100 text-green-700"
239:                      : item.status === "LOST"
240:                      ? "bg-yellow-100 text-yellow-700"
241:                      : "bg-blue-100 text-blue-700"
242:                  }`}
243:                >
244:                  {item.status}
245:                </span>
246:              </div>
247:
248:              {/* Action Buttons */}
249:              <div className="mt-3 flex gap-2">
250:                {item.status === "CLAIMED" ? (
251:                  <p className="text-sm text-red-600 font-medium">
252:                    This item has been claimed
253:                  </p>
254:                ) : currentUser && item.createdBy === currentUser.id ? (
255:                  <p className="text-sm text-gray-600 italic">
256:                    You posted this item
257:                  </p>
258:                ) : item.status === "FOUND" ? (
259:                  <
260:                    <button
261:                      onClick={() => {
262:                        setCurrentItem(item);

```



```

263:                 setShowClaimModal(true);
264:             }}
265:             className="flex-1 text-sm bg-blue-600 text-white py-2 rounded-lg hov
| er:bg-blue-700 transition font-medium"
266:         >
267:             Claim Item
268:         </button>
269:         <button
270:             onClick={() => {
271:                 setCurrentItem(item);
272:                 setShowMessageBox(true);
273:             }}
274:             className="flex-1 text-sm border border-blue-600 text-blue-600 py-2
| rounded-lg hover:bg-blue-50 transition font-medium"
275:         >
276:             Message
277:         </button>
278:     </>
279: ) : item.status === "LOST" ? (
280:     <button
281:         onClick={() => {
282:             setCurrentItem(item);
283:             setShowMessageBox(true);
284:         }}
285:         className="w-full text-sm border border-blue-600 text-blue-600 py-2 ro
| unded-lg hover:bg-blue-50 transition font-medium"
286:     >
287:         Message Owner
288:     </button>
289: ) : null}
290: </div>
291: </div>
292: </div>
293: ))}
294: </div>
295:
296: {items.length === 0 && (
297:     <div className="text-center py-20">
298:         <p className="text-gray-500 text-lg">No items found.</p>
299:         <p className="text-gray-400 text-sm mt-2">
300:             Try adjusting your search or filter criteria.
301:         </p>
302:     </div>
303: )}
304:
305: {/* Message Modal */}
306: {showMessageBox && currentItem && (
307:     <div className="fixed inset-0 bg-black bg-opacity-50 px-4 flex items-center justif
| y-center z-50">
308:         <div className="bg-white p-6 rounded-xl shadow-xl w-full max-w-md space-y-4">
309:             <h3 className="text-lg font-bold text-blue-600">Send Message</h3>
310:             <p className="text-sm text-gray-600">
311:                 To: <strong>{currentItem.creatorName}</strong> <br />
312:                 Item: <strong>{currentItem.name}</strong>
313:             </p>
314:             <textarea
315:                 rows={4}
316:                 className="w-full border border-gray-300 rounded-lg px-3 py-2 focus:outline-
| none focus:ring-2 focus:ring-blue-500 resize-none"
317:                 placeholder="Type your message... (max 1000 characters)"
318:                 value={messageText}
319:                 onChange={(e) => setMessageText(e.target.value)}
320:                 maxLength={1000}
321:             />
322:             <div className="text-xs text-gray-500 text-right">
323:                 {messageText.length}/1000 characters
324:             </div>
325:             <div className="flex justify-end gap-2">
326:                 <button
327:                     onClick={() => {
328:                         setShowMessageBox(false);
329:                         setMessageText("");
330:                         setCurrentItem(null);
331:                     }}
332:                     className="text-sm px-4 py-2 bg-gray-200 rounded-lg hover:bg-gray-300 tran
| sition"
333:                 >
334:                     Cancel
335:                 </button>
336:                 <button
337:                     onClick={handleSendMessage}
338:                     disabled={sendingMessage || !messageText.trim()}
339:                     className="text-sm px-4 py-2 bg-blue-600 text-white rounded-lg hover:bg-bl
| ue-700 transition disabled:opacity-50 disabled:cursor-not-allowed"
340:                 >

```

```

341:         {sendingMessage ? "Sending..." : "Send"}
342:     </button>
343: </div>
344: </div>
345: </div>
346: )}
347:
348: { /* Claim Modal */}
349: {showClaimModal && currentItem && (
350:     <div className="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-cen
| ter z-50 px-4">
351:         <div className="bg-white p-6 rounded-xl shadow-xl w-full max-w-md space-y-4">
352:             <h3 className="text-lg font-bold text-blue-600">
353:                 Claim Item: {currentItem.name}
354:             </h3>
355:             <p className="text-sm text-gray-600">
356:                 Your registered name and email will be used for this claim.
357:                 The item owner will be notified.
358:             </p>
359:             <div className="flex justify-end gap-2">
360:                 <button
361:                     onClick={() => {
362:                         setShowClaimModal(false);
363:                         setCurrentItem(null);
364:                     }}
365:                     className="text-sm px-4 py-2 border border-gray-300 rounded-lg hover:bg-gr
| ay-50 transition"
366:                 >
367:                     Cancel
368:                 </button>
369:                 <button
370:                     onClick={() => handleClaim(currentItem.id)}
371:                     disabled={claimingItem}
372:                     className="text-sm px-4 py-2 bg-blue-600 text-white rounded-lg hover:bg-bl
| ue-700 transition disabled:opacity-50 disabled:cursor-not-allowed"
373:                 >
374:                     {claimingItem ? "Claiming..." : "Confirm Claim"}
375:                 </button>
376:             </div>
377:         </div>
378:     </div>
379: )}
380: </div>
381: );
382: }

```

■ File: src\pages\LoginPage.js

```

=====
1: // src/pages/LoginPage.js
2: import { useState } from "react";
3: import { useNavigate } from "react-router-dom";
4: import { API_BASE_URL, API_ENDPOINTS } from "../config/api";
5: import toast, { Toaster } from "react-hot-toast";
6:
7: export default function LoginPage() {
8:     const navigate = useNavigate();
9:     const [email, setEmail] = useState("");
10:    const [password, setPassword] = useState("");
11:    const [loading, setLoading] = useState(false);
12:
13:    const handleLogin = async (e) => {
14:        e.preventDefault();
15:        setLoading(true);
16:
17:        try {
18:            const res = await fetch(`${API_BASE_URL}${API_ENDPOINTS.LOGIN}`, {
19:                method: "POST",
20:                headers: {
21:                    "Content-Type": "application/json",
22:                },
23:                credentials: "include", // Important: This allows cookies to be set
24:                body: JSON.stringify({ email, password }),
25:            });
26:
27:            const data = await res.json();
28:
29:            if (!res.ok || !data.success) {
30:                toast.error(data.message || "Invalid email or password.");
31:                return;
32:            }
33:
34:            toast.success("Login successful!");
35:

```

```

36: // Backend sets HTTP-only cookies, no need to store tokens in localStorage
37: // Redirect based on user role
38: if (data.user?.role === "ADMIN") {
39:   navigate("/admin");
40: } else {
41:   navigate("/home");
42: }
43: } catch (err) {
44:   console.error("Login error:", err);
45:   toast.error("Login failed. Please try again.");
46: } finally {
47:   setLoading(false);
48: }
49: };
50:
51: return (
52:   <div className="flex items-center justify-center min-h-screen bg-gradient-to-br from-b
| lue-50 to-indigo-100 px-4">
53:     <Toaster position="top-center" />
54:     <div className="bg-white p-8 rounded-2xl shadow-xl w-full max-w-md">
55:       <div className="text-center mb-6">
56:         <h1 className="text-3xl font-bold text-blue-600 mb-2">Lost & Found</h1>
57:         <h2 className="text-xl font-semibold text-gray-700">Welcome Back</h2>
58:         <p className="text-sm text-gray-500 mt-1">Sign in to your account</p>
59:       </div>
60:
61:       <form onSubmit={handleLogin} className="space-y-4">
62:         <div>
63:           <label className="block text-sm font-medium text-gray-700 mb-1">
64:             Email Address
65:           </label>
66:           <input
67:             type="email"
68:             placeholder="your@email.com"
69:             className="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-
| none focus:ring-2 focus:ring-blue-500 focus:border-transparent"
70:             value={email}
71:             onChange={(e) => setEmail(e.target.value)}
72:             required
73:           />
74:         </div>
75:
76:         <div>
77:           <label className="block text-sm font-medium text-gray-700 mb-1">
78:             Password
79:           </label>
80:           <input
81:             type="password"
82:             placeholder="???????"
83:             className="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-
| none focus:ring-2 focus:ring-blue-500 focus:border-transparent"
84:             value={password}
85:             onChange={(e) => setPassword(e.target.value)}
86:             required
87:           />
88:         </div>
89:
90:         <button
91:           type="submit"
92:           disabled={loading}
93:           className="w-full bg-blue-600 text-white py-2.5 rounded-lg font-medium hover:b
| g-blue-700 transition disabled:opacity-50 disabled:cursor-not-allowed"
94:         >
95:           {loading ? "Signing in..." : "Sign In"}
96:         </button>
97:       </form>
98:
99:       <div className="mt-6 text-center text-sm text-gray-600">
100:         Don't have an account?{" "}
101:         <button
102:           onClick={() => navigate("/register")}
103:           className="text-blue-600 font-medium hover:underline"
104:         >
105:           Create one
106:         </button>
107:       </div>
108:     </div>
109:   </div>
110: );
111: }

```

■ File: src\pages\RegisterPage.js

```
=====
1: // src/pages/RegisterPage.js
2: import { useState } from "react";
3: import { useNavigate } from "react-router-dom";
4: import { API_BASE_URL, API_ENDPOINTS } from "../config/api";
5: import toast, { Toaster } from "react-hot-toast";
6:
7: export default function RegisterPage() {
8:   const navigate = useNavigate();
9:   const [formData, setFormData] = useState({
10:     name: "",
11:     email: "",
12:     password: "",
13:     confirmPassword: "",
14:   });
15:   const [loading, setLoading] = useState(false);
16:
17:   const handleChange = (e) => {
18:     setFormData((prev) => ({
19:       ...prev,
20:       [e.target.name]: e.target.value,
21:     }));
22:   };
23:
24:   const handleSubmit = async (e) => {
25:     e.preventDefault();
26:
27:     // Validation
28:     if (formData.password !== formData.confirmPassword) {
29:       toast.error("Passwords do not match.");
30:       return;
31:     }
32:
33:     // Password validation to match backend requirements
34:     const passwordRegex = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[^\w-zA-Z\d]).{8,}$/;
35:     if (!passwordRegex.test(formData.password)) {
36:       toast.error(
37:         "Password must contain at least one uppercase letter, one lowercase letter, one nu
38: | mber, and one special character."
39:       );
40:       return;
41:     }
42:
43:     // Name validation
44:     if (formData.name.length < 2 || formData.name.length > 100) {
45:       toast.error("Name must be between 2 and 100 characters.");
46:       return;
47:     }
48:
49:     const nameRegex = /^[a-zA-Z\s]+$/;
50:     if (!nameRegex.test(formData.name)) {
51:       toast.error("Name should only contain letters and spaces.");
52:       return;
53:     }
54:
55:     setLoading(true);
56:
57:     try {
58:       const res = await fetch(`${API_BASE_URL}${API_ENDPOINTS.REGISTER}`, {
59:         method: "POST",
60:         headers: {
61:           "Content-Type": "application/json",
62:         },
63:         credentials: "include", // Important for cookies
64:         body: JSON.stringify(formData),
65:       });
66:
67:       const data = await res.json();
68:
69:       if (!res.ok || !data.success) {
70:         // Handle validation errors
71:         if (data.errors) {
72:           Object.values(data.errors).forEach((error) => toast.error(error));
73:         } else {
74:           toast.error(data.message || "Registration failed.");
75:         }
76:         return;
77:       }
78:
79:       toast.success("Registration successful! Redirecting to home...");
80:
81:       // Backend auto-logs in and sets HTTP-only cookies
82:       // No need to store tokens in localStorage
83:       setTimeout(() => navigate("/home"), 2000);
84:     } catch (error) {
85:       toast.error("Registration failed. Please try again later.");
86:     }
87:   };
88:
89:   return (
90:     <div>
91:       <h2>Register</h2>
92:       <form>
93:         <input type="text" value={formData.name} onChange={handleChange} />
94:         <input type="email" value={formData.email} onChange={handleChange} />
95:         <input type="password" value={formData.password} onChange={handleChange} />
96:         <input type="password" value={formData.confirmPassword} onChange={handleChange} />
97:         <button type="button" disabled={loading}>Register</button>
98:       </form>
99:     </div>
100:   );
101: }
=====
```

```

83:     } catch (err) {
84:         console.error("Registration error:", err);
85:         toast.error("Something went wrong. Please try again.");
86:     } finally {
87:         setLoading(false);
88:     }
89: };
90:
91: return (
92:     <div className="min-h-screen flex items-center justify-center bg-gradient-to-br from-b
| lue-50 to-indigo-100 px-4">
93:         <Toaster position="top-center" />
94:         <div className="bg-white p-8 rounded-2xl shadow-xl w-full max-w-md">
95:             <div className="text-center mb-6">
96:                 <h1 className="text-3xl font-bold text-blue-600 mb-2">Lost & Found</h1>
97:                 <h2 className="text-xl font-semibold text-gray-700">Create Account</h2>
98:                 <p className="text-sm text-gray-500 mt-1">Join our community today</p>
99:             </div>
100:
101:             <form onSubmit={handleSubmit} className="space-y-4">
102:                 <div>
103:                     <label className="block text-sm font-medium text-gray-700 mb-1">
104:                         Full Name
105:                     </label>
106:                     <input
107:                         type="text"
108:                         name="name"
109:                         placeholder="John Doe"
110:                         className="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-
| none focus:ring-2 focus:ring-blue-500 focus:border-transparent"
111:                         onChange={handleChange}
112:                         value={formData.name}
113:                         required
114:                     />
115:                 </div>
116:
117:                 <div>
118:                     <label className="block text-sm font-medium text-gray-700 mb-1">
119:                         Email Address
120:                     </label>
121:                     <input
122:                         type="email"
123:                         name="email"
124:                         placeholder="your@email.com"
125:                         className="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-
| none focus:ring-2 focus:ring-blue-500 focus:border-transparent"
126:                         onChange={handleChange}
127:                         value={formData.email}
128:                         required
129:                     />
130:                 </div>
131:
132:                 <div>
133:                     <label className="block text-sm font-medium text-gray-700 mb-1">
134:                         Password
135:                     </label>
136:                     <input
137:                         type="password"
138:                         name="password"
139:                         placeholder="???????"
140:                         className="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-
| none focus:ring-2 focus:ring-blue-500 focus:border-transparent"
141:                         onChange={handleChange}
142:                         value={formData.password}
143:                         required
144:                     />
145:                     <p className="text-xs text-gray-500 mt-1">
146:                         Min 8 chars with uppercase, lowercase, number & special character
147:                     </p>
148:                 </div>
149:
150:                 <div>
151:                     <label className="block text-sm font-medium text-gray-700 mb-1">
152:                         Confirm Password
153:                     </label>
154:                     <input
155:                         type="password"
156:                         name="confirmPassword"
157:                         placeholder="???????"
158:                         className="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-
| none focus:ring-2 focus:ring-blue-500 focus:border-transparent"
159:                         onChange={handleChange}
160:                         value={formData.confirmPassword}
161:                         required
162:                     />

```

```

163:         </div>
164:
165:         <button
166:             type="submit"
167:             disabled={loading}
168:             className="w-full bg-blue-600 text-white py-2.5 rounded-lg font-medium hover:b
| g-blue-700 transition disabled:opacity-50 disabled:cursor-not-allowed"
169:         >
170:             {loading ? "Creating Account..." : "Create Account"}
171:         </button>
172:     </form>
173:
174:     <div className="mt-6 text-center text-sm text-gray-600">
175:         Already have an account?{" "}
176:         <button
177:             onClick={() => navigate("/")}
178:             className="text-blue-600 font-medium hover:underline"
179:         >
180:             Sign in
181:         </button>
182:     </div>
183: </div>
184: </div>
185: );
186: }

```

■ File: tailwind.config.js

```

1: /** @type {import('tailwindcss').Config} */
2: module.exports = {
3:   content: [
4:     "./src/**/*..{js,jsx,ts,tsx}", // very important for React
5:   ],
6:   theme: {
7:     extend: {},
8:   },
9:   plugins: [],
10: };

```