

## ■ 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

```
=====
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
| 0">
18:               <tr>
19:                 {headers.map((h, idx) => (
20:                   <th key={idx} className="px-5 py-3 border-b font-bold">
21:                     {h}
22:                   </th>
23:                 )))
24:               </tr>
25:             </thead>
26:
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:                key={idx}
38:                className="border-t text-sm text-gray-800 hover:bg-gray-50 transition">
39:                <td className="px-4 py-3 font-medium">{claim.itemName}</td>
40:                <td className="px-4 py-3">{claim.location}</td>
41:                <td className="px-4 py-3 text-sm text-gray-600">
42:                  {formatDate(claim.claimedAt)}
43:                </td>
44:              </tr>
45:            )));
46:          )
47:        </tbody>
48:      </table>
49:    </div>
50:  );
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("An error occurred while submitting feedback.");
30:    }
31:  };
32:
33:  return (
34:    <div>
35:      <form onSubmit={handleSubmit}>
36:        <input type="text" value={text} onChange={(e) => setText(e.target.value)} />
37:        <button type="submit" disabled={loading}>Submit</button>
38:      </form>
39:    </div>
40:  );
41: }
```

```

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:                method: "GET",
21:                credentials: "include", // Send cookies with request
22:                headers: {
23:                    Accept: "application/json",
24:                },
25:            });
26:
27:            if (res.ok) {
28:                const response = await res.json();
29:                // Backend returns: { success: true, message: "...", data: { id, name, email, role
| } }
30:                if (response.success && response.data) {
31:                    setUser(response.data);
32:                } else {
33:                    setUser(null);
34:                }
35:            } else {
36:                // Don't redirect here - just clear user state
37:                setUser(null);
38:            }
39:        } catch (err) {
40:            console.error("Error fetching user:", err);
41:            setUser(null);
42:        } finally {

```

```

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

```

```

125:             <Link
126:               to="/"
127:               className="hover:text-blue-600 transition font-medium"
128:             >
129:               Login
130:             </Link>
131:             <Link
132:               to="/register"
133:               className="bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-blue-700
| transition font-medium"
134:               >
135:                 Register
136:               </Link>
137:             </>
138:           )
139:         </nav>
140:
141:         {/* Mobile Hamburger */}
142:         <button
143:           className="md:hidden text-gray-800 focus:outline-none"
144:           onClick={() => setSidebarOpen(true)}
145:           aria-label="Open menu"
146:         >
147:           <svg className="w-6 h-6 fill="none" stroke="currentColor">
148:             <path
149:               strokeLinecap="round"
150:               strokeLinejoin="round"
151:               strokeWidth={2}
152:               d="M4 6h16M4 12h16M4 18h16"
153:             />
154:           </svg>
155:         </button>
156:       </div>
157:     </header>
158:
159:     {/* Sidebar overlay */}
160:     {sidebarOpen &&
161:       <div
162:         className="fixed inset-0 bg-black bg-opacity-40 z-40"
163:         onClick={() => setSidebarOpen(false)}
164:       ></div>
165:     }
166:
167:     {/* Sidebar */}
168:     <div
169:       className={`fixed top-0 left-0 h-full w-64 bg-white shadow-lg z-50 transform trans
| ition-transform duration-300 ${

170:         sidebarOpen ? "translate-x-0" : "-translate-x-full"
171:       } md:hidden`}
172:     >
173:       <div className="p-4 flex items-center justify-between border-b">
174:         <h2 className="text-lg font-bold text-blue-600">Menu</h2>
175:         <button
176:           className="text-gray-600 text-2xl"
177:           onClick={() => setSidebarOpen(false)}
178:           aria-label="Close menu"
179:         >
180:           x
181:         </button>
182:       </div>
183:       <div className="flex flex-col gap-4 p-4 text-sm">
184:         {user ? (
185:           <>
186:             <div className="pb-3 border-b">
187:               <p className="text-xs text-gray-500">Logged in as</p>
188:               <p className="font-medium text-gray-800">{user.name}</p>
189:             </div>
190:             <Link
191:               to="/home"
192:               onClick={() => setSidebarOpen(false)}
193:               className="hover:text-blue-600 py-2"
194:             >
195:               Home
196:             </Link>
197:             <Link
198:               to="/register-item"
199:               onClick={() => setSidebarOpen(false)}
200:               className="hover:text-blue-600 py-2"
201:             >
202:               Register Item
203:             </Link>
204:             <Link
205:               to="/lost-found-items"
206:               onClick={() => setSidebarOpen(false)}
207:               className="hover:text-blue-600 py-2"
208:             >
209:               Lost Found Items
210:             </Link>
211:           </>
212:         ) : (
213:           <div>
214:             <h3>Welcome to the App!</h3>
215:             <p>Please log in or register to use the app.</p>
216:           </div>
217:         )
218:       </div>
219:     </div>
220:   </div>
221: </body>
222: </html>

```

```

208:         >
209:             Browse Items
210:             </Link>
211:             {user.role === "ADMIN" && (
212:                 <Link
213:                     to="/admin"
214:                     onClick={() => setSidebarOpen(false)}
215:                     className="hover:text-blue-600 py-2"
216:                 >
217:                     Admin Dashboard
218:                     </Link>
219:                 )}
220:                 <button
221:                     onClick={() => {
222:                         handleLogout();
223:                         setSidebarOpen(false);
224:                     }}
225:                     className="text-left text-red-600 hover:underline py-2"
226:                 >
227:                     Logout
228:                     </button>
229:                 </>
230:             ) : (
231:                 <>
232:                     <Link
233:                         to="/"
234:                         onClick={() => setSidebarOpen(false)}
235:                         className="hover:text-blue-600 py-2"
236:                     >
237:                         Login
238:                         </Link>
239:                         <Link
240:                             to="/register"
241:                             onClick={() => setSidebarOpen(false)}
242:                             className="hover:text-blue-600 py-2"
243:                         >
244:                             Register
245:                             </Link>
246:                         </>
247:                     )
248:                 </div>
249:             </div>
250:         </>
251:     );
252: }

```

---

## ■ 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:shad
| 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:outline-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-dxl transition disabled:opacity-50 disabled:cursor-not-allowed"
149:             onClick={() => handleReply(msg.id, msg.senderId, msg.itemId)}
150:             disabled={
151:                 !replies[msg.id]?.trim() || sendingReply[msg.id]
152:             }
153:         >
154:             {sendingReply[msg.id] ? "Sending..." : "Send Reply"}
155:         </button>
156:     </div>
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:     // Only attempt refresh for non-auth endpoints
69:     if (response.status === 401 &&
70:       !endpoint.includes('/api/auth/') &&
71:       endpoint !== API_ENDPOINTS.REFRESH) {
72:
73:       // Try to refresh token
74:       try {
75:         const refreshResponse = await fetch(`#${API_BASE_URL}${API_ENDPOINTS.REFRESH}`, {
76:           method: 'POST',
77:           credentials: 'include',
78:         });
79:
80:         if (refreshResponse.ok) {
81:           // Retry original request
82:           return fetch(`#${API_BASE_URL}${endpoint}`, finalOptions);
83:         } else {
84:           // Refresh failed, redirect to login
85:           window.location.href = '/';
86:           throw new Error('Session expired');
87:         }
88:       } catch (refreshError) {
89:         console.error('Token refresh failed:', refreshError);
90:         window.location.href = '/';
91:         throw new Error('Session expired');
92:       }
93:     }
94:
95:     return response;
96:   } catch (error) {
97:     console.error('API call error:', error);
98:     throw error;
99:   }
100: }

```

---

## ■ 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:         method: 'GET',
26:         credentials: "include",
27:         headers: {
28:           'Accept': 'application/json',
29:         },
30:       });
31:
32:       if (!res.ok) {
33:         toast.error("Authentication required");
34:         navigate("/");
35:         return false;
36:       }
37:
38:       const response = await res.json();
39:       const user = response.data;
40:
41:       if (!user || user.role !== "ADMIN") {
42:         toast.error("Access denied. Admin privileges required.");
43:         navigate("/");
44:         return false;
45:       }
46:       return true;
47:     } catch (err) {
48:       console.error("Admin validation error:", err);
49:       toast.error("Error validating admin access.");
50:       navigate("/");
51:       return false;
52:     }
53:   };
54:
55:   // Fetch Admin Dashboard Data
56:   const fetchAdminData = async () => {
57:     try {
58:       const res = await fetch(`/${API_BASE_URL}${API_ENDPOINTS.ADMIN_DASHBOARD}`, {
59:         method: 'GET',
60:         credentials: "include",
61:         headers: {
62:           'Accept': 'application/json',
63:         },
64:       });
65:
66:       if (!res.ok) {
67:         if (res.status === 403) {
68:           toast.error("Access denied");
69:           navigate("/");
70:         }
71:         throw new Error("Failed to load dashboard");
72:       }
73:
74:       const data = await res.json();
75:
76:       if (data.success) {
77:         setItems(data.items || []);
78:         setClaims(data.claims || []);
79:         setUsers(data.users || []);
80:         setFeedback(data.feedback || []);
81:         setStats(data.stats || {});
82:       }
83:     } catch (err) {
```

```

84:     toast.error("Failed to load admin data.");
85:     console.error(err);
86:   } finally {
87:     setLoading(false);
88:   }
89: };
90:
91: useEffect(() => {
92:   const init = async () => {
93:     const isAdmin = await validateAdminAccess();
94:     if (isAdmin) await fetchAdminData();
95:   };
96:   init();
97: }, []);
98:
99: // Delete Handler with Confirmation
100: const handleDelete = async (type, id) => {
101:   const confirmed = await MySwal.fire({
102:     title: `Delete this ${type}?`,
103:     text: `This action cannot be undone.`,
104:     icon: "warning",
105:     showCancelButton: true,
106:     confirmButtonColor: "#d33",
107:     cancelButtonColor: "#3085d6",
108:     confirmButtonText: "Yes, delete it!",
109:     cancelButtonText: "Cancel",
110:   });
111:
112:   if (!confirmed.isConfirmed) return;
113:
114:   let endpoint;
115:   switch (type) {
116:     case "item":
117:       endpoint = API_ENDPOINTS.ADMIN_DELETE_ITEM(id);
118:       break;
119:     case "claim":
120:       endpoint = API_ENDPOINTS.ADMIN_DELETE_CLAIM(id);
121:       break;
122:     case "user":
123:       endpoint = API_ENDPOINTS.ADMIN_DELETE_USER(id);
124:       break;
125:     case "feedback":
126:       endpoint = API_ENDPOINTS.ADMIN_DELETE_FEEDBACK(id);
127:       break;
128:     default:
129:       return;
130:   }
131:
132:   try {
133:     const res = await fetch(`/${API_BASE_URL}${endpoint}`, {
134:       method: "DELETE",
135:       credentials: "include",
136:       headers: {
137:         'Accept': 'application/json',
138:       },
139:     });
140:
141:     const data = await res.json();
142:
143:     if (res.ok && data.success) {
144:       toast.success(data.message || `${type} deleted successfully.`);
145:       await fetchAdminData(); // Refresh data
146:     } else {
147:       toast.error(data.message || `Failed to delete ${type}.`);
148:     }
149:   } catch (err) {
150:     console.error(err);
151:     toast.error(`Error deleting ${type}.`);
152:   }
153: };
154:
155: const formatDate = (dateString) => {
156:   if (!dateString) return "N/A";
157:   const date = new Date(dateString);
158:   return date.toLocaleDateString("en-US", {
159:     year: "numeric",
160:     month: "short",
161:     day: "numeric",
162:   });
163: };
164:
165: if (loading) {
166:   return (
167:     <div className="min-h-screen flex items-center justify-center">
168:       <div className="text-center">

```

```

169:         <div className="animate-spin rounded-full h-12 w-12 border-b-2 border-blue-600 m
170: | x-auto mb-4"></div>
171:     <p className="text-gray-600">Verifying admin access...</p>
172:     </div>
173:   );
174: }
175:
176: return (
177:   <div className="max-w-7xl mx-auto px-6 py-8 space-y-8">
178:     <Toaster position="top-center" />
179:
180:     <div className="text-center">
181:       <h1 className="text-3xl font-bold text-blue-600 mb-2">
182:         Admin Dashboard
183:       </h1>
184:       <p className="text-gray-600">Manage all platform activities</p>
185:     </div>
186:
187:     /* Statistics Cards */
188:     {stats && (
189:       <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-4">
190:         <div className="bg-blue-50 border border-blue-200 rounded-xl p-4">
191:           <h3 className="text-sm font-medium text-blue-600">Total Items</h3>
192:           <p className="text-2xl font-bold text-blue-700 mt-1">
193:             {stats.totalItems || 0}
194:           </p>
195:         </div>
196:         <div className="bg-green-50 border border-green-200 rounded-xl p-4">
197:           <h3 className="text-sm font-medium text-green-600">Total Claims</h3>
198:           <p className="text-2xl font-bold text-green-700 mt-1">
199:             {stats.totalClaims || 0}
200:           </p>
201:         </div>
202:         <div className="bg-purple-50 border border-purple-200 rounded-xl p-4">
203:           <h3 className="text-sm font-medium text-purple-600">Total Users</h3>
204:           <p className="text-2xl font-bold text-purple-700 mt-1">
205:             {stats.totalUsers || 0}
206:           </p>
207:         </div>
208:         <div className="bg-orange-50 border border-orange-200 rounded-xl p-4">
209:           <h3 className="text-sm font-medium text-orange-600">Total Feedback</h3>
210:           <p className="text-2xl font-bold text-orange-700 mt-1">
211:             {stats.totalFeedback || 0}
212:           </p>
213:         </div>
214:       </div>
215:     )})
216:
217:     /* Items Table */
218:     <AdminTable
219:       title="Registered Items"
220:       type="item"
221:       data={items}
222:       headers={['ID', 'Name', 'Location', 'Status', 'Created By', 'Actions']}
223:       rowRenderer={(item) => [
224:         item.id,
225:         item.name,
226:         item.location,
227:         <span
228:           className={`inline-block px-2 py-1 rounded-full text-xs font-medium ${item.status === "FOUND" ? "bg-green-100 text-green-700" : item.status === "LOST" ? "bg-yellow-100 text-yellow-700" : "bg-blue-100 text-blue-700"}`}
229:         >
230:           {item.status}
231:         </span>,
232:         item.creatorName,
233:         <button
234:           onClick={() => handleDelete("item", item.id)}
235:           className="text-red-600 hover:text-red-800 text-sm font-medium"
236:         >
237:           Delete
238:         </button>,
239:       ]})
240:     />
241:
242:     /* Claims Table */
243:     <AdminTable
244:       title="Claims"
245:       type="claim"
246:       data={claims}

```

```

253:     headers={[ "ID", "Item", "Claimant", "Email", "Claimed At", "Actions" ]}
254:     rowRenderer={(claim) => [
255:       claim.id,
256:       claim.itemName,
257:       claim.claimantName,
258:       claim.claimantEmail,
259:       formatDate(claim.claimedAt),
260:       <button
261:         onClick={() => handleDelete("claim", claim.id)}
262:         className="text-red-600 hover:text-red-800 text-sm font-medium"
263:       >
264:         Delete
265:       </button>,
266:     ]}
267:   />
268:
269: /* Users Table */
270: <AdminTable
271:   title="Users"
272:   type="user"
273:   data={users}
274:   headers={[ "ID", "Name", "Email", "Role", "Registered", "Actions" ]}
275:   rowRenderer={(user) => [
276:     user.id,
277:     user.name,
278:     user.email,
279:     <span
280:       className={`inline-block px-2 py-1 rounded-full text-xs font-medium ${user.role === "ADMIN" ? "bg-red-100 text-red-700" : "bg-gray-100 text-gray-700"}`}
281:     }>
282:     {user.role}
283:   </span>,
284:   formatDate(user.createdAt),
285:   user.role === "ADMIN" ? (
286:     <span className="text-gray-400 text-sm">Protected</span>
287:   ) : (
288:     <button
289:       onClick={() => handleDelete("user", user.id)}
290:       className="text-red-600 hover:text-red-800 text-sm font-medium"
291:     >
292:       Delete
293:     </button>
294:   ),
295:   ],
296:   1
297:   />
298:   ],
299:   1
300:   />
301:
302: /* Feedback Table */
303: <AdminTable
304:   title="Feedback"
305:   type="feedback"
306:   data={feedback}
307:   headers={[ "ID", "User", "Feedback", "Submitted", "Actions" ]}
308:   rowRenderer={(fb) => [
309:     fb.id,
310:     fb.userName || "Anonymous",
311:     <div className="max-w-md truncate">{fb.feedbackText}</div>,
312:     formatDate(fb.submittedAt),
313:     <button
314:       onClick={() => handleDelete("feedback", fb.id)}
315:       className="text-red-600 hover:text-red-800 text-sm font-medium"
316:     >
317:       Delete
318:     </button>,
319:   ],
320:   1
321:   />
322:   );
323: }

```

---

## ■ 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:   }, []);
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:         headers: {
31:           'Accept': 'application/json',
32:         },
33:       });
34:
35:       if (!res.ok) {
36:         if (res.status === 401 || res.status === 403) {
37:           // Unauthorized - redirect to login
38:           toast.error("Please login to continue");
39:           navigate("/");
40:         } else {
41:           toast.error("Failed to load dashboard");
42:         }
43:       }
44:     }
45:
46:     const data = await res.json();
47:
48:     if (data.user) {
49:       setName(data.user.name);
50:       setRole(data.user.role);
51:       setItems(data.items || []);
52:       setClaims(data.claims || []);
53:       setMessages(data.messages || []);
54:     } else {
55:       navigate("/");
56:     }
57:   } catch (err) {
58:     console.error("Failed to fetch dashboard:", err);
59:     toast.error("Network error. Please try again.");
60:     navigate("/");
61:   } finally {
62:     setLoading(false);
63:   }
64: };
65:
66: const handleFeedbackSubmit = async (e) => {
67:   e.preventDefault();
68:
69:   if (!feedbackText.trim()) {
70:     toast.error("Please enter your feedback");
71:     return;
72:   }
73:
74:   if (feedbackText.trim().length < 10) {
75:     toast.error("Feedback must be at least 10 characters long");
76:     return;
77:   }
78:
79:   if (feedbackText.trim().length > 2000) {
80:     toast.error("Feedback cannot exceed 2000 characters");
81:     return;
82:   }
83:
84:   setSubmittingFeedback(true);
85:
86:   try {
87:     const res = await fetch(`/${API_BASE_URL}${API_ENDPOINTS.FEEDBACK}`, {
88:       method: "POST",
89:       credentials: "include", // Send cookies
90:       headers: {
91:         "Content-Type": "application/json",
92:       },
93:       body: JSON.stringify({ feedback: feedbackText.trim() }),
94:     });

```

```

95:
96:     const data = await res.json();
97:
98:     if (res.ok && data.success) {
99:         toast.success(data.message || "Thank you for your feedback!");
100:        setFeedbackText("");
101:    } else {
102:        toast.error(data.message || "Failed to submit feedback.");
103:    }
104: } catch (err) {
105:     console.error(err);
106:     toast.error("Something went wrong while submitting feedback.");
107: } finally {
108:     setSubmittingFeedback(false);
109: }
110: }
111:
112: if (loading) {
113:     return (
114:         <div className="min-h-screen flex items-center justify-center text-gray-600">
115:             <div className="text-center">
116:                 <div className="animate-spin rounded-full h-12 w-12 border-b-2 border-blue-600 m
| x-auto mb-4"></div>
117:                 <p>Loading your dashboard...</p>
118:             </div>
119:         </div>
120:     );
121: }
122:
123: return (
124:     <div className="p-6 max-w-6xl mx-auto space-y-10 pb-20">
125:         <Toaster position="top-center" />
126:
127:         <div className="text-center">
128:             <h1 className="text-3xl font-bold text-gray-800">Welcome, {name}!</h1>
129:             <p className="text-sm text-gray-500 mt-1">
130:                 Here's an overview of your account and activity.
131:             </p>
132:         </div>
133:
134:         /* My Registered Items */
135:         <section className="bg-white p-6 rounded-2xl shadow-sm border border-gray-200">
136:             <h2 className="text-xl font-semibold text-gray-700 mb-4 border-b pb-2">
137:                 My Registered Items
138:             </h2>
139:             <ItemTable items={items} />
140:         </section>
141:
142:         /* My Claims */
143:         <section className="bg-white p-6 rounded-2xl shadow-sm border border-gray-200">
144:             <h2 className="text-xl font-semibold text-gray-700 mb-4 border-b pb-2">
145:                 My Claims
146:             </h2>
147:             <ClaimsTable claims={claims} />
148:         </section>
149:
150:         /* Messages */
151:         <section className="bg-white p-6 rounded-2xl shadow-sm border border-gray-200">
152:             <h2 className="text-xl font-semibold text-gray-700 mb-4 border-b pb-2">
153:                 Messages
154:             </h2>
155:             <MessageList messages={messages} />
156:         </section>
157:
158:         /* Platform Feedback - Only for non-admin users */
159:         {role !== "ADMIN" && (
160:             <section className="bg-white p-6 rounded-2xl shadow-sm border border-gray-200">
161:                 <h2 className="text-xl font-semibold text-gray-700 mb-4 border-b pb-2">
162:                     Platform Feedback
163:                 </h2>
164:                 <form onSubmit={handleFeedbackSubmit} className="space-y-4">
165:                     <textarea
166:                         rows={4}
167:                         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"
168:                         placeholder="Share your experience with us... (minimum 10 characters)"
169:                         value={feedbackText}
170:                         onChange={(e) => setFeedbackText(e.target.value)}
171:                         required
172:                         maxLength={2000}
173:                     />
174:                     <div className="flex justify-between items-center">
175:                         <span className="text-xs text-gray-500">
176:                             {feedbackText.length}/2000 characters
177:                         </span>

```

```

178:             <button
179:                 type="submit"
180:                 disabled={submittingFeedback || feedbackText.trim().length < 10}
181:                 className="bg-blue-600 text-white px-6 py-2 rounded-xl hover:bg-blue-700 transition disabled:opacity-50 disabled:cursor-not-allowed"
182:             >
183:                 {submittingFeedback ? "Submitting..." : "Submit Feedback"}
184:             </button>
185:         </div>
186:     </form>
187:     </section>
188:   )
189: </div>
190: );
191: }

```

---

## ■ 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:                 <CameraIcon className="w-5 h-5" />
230:                 Register Item
231:             </>
232:         )}
233:     </button>
234:   </div>
235: </form>
236:
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 [sendMessage, setSendMessage] = 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:        method: "GET",
36:        credentials: "include",
37:        headers: {
38:          Accept: "application/json",
39:        },
40:      });
41:      if (res.ok) {
42:        const data = await res.json();
43:        if (data.success && data.data) {
44:          setCurrentUser(data.data); // Backend returns data in 'data' field
45:        }
46:      }
47:    } catch (err) {
48:      console.error("Failed to fetch user", err);
49:    }
50:  };
51:
52:  const fetchItems = async () => {
53:    try {
54:      const params = new URLSearchParams();

```

```

55:     if (search) params.append("search", search);
56:     if (filter) params.append("status", filter);
57:
58:     const res = await fetch(
59:       `${API_BASE_URL}${API_ENDPOINTS.ITEMS}?${params.toString()}`,
60:       {
61:         method: "GET",
62:         credentials: "include",
63:         headers: {
64:           Accept: "application/json",
65:         },
66:       }
67:     );
68:
69:     if (!res.ok) throw new Error("Failed to fetch items");
70:
71:     const data = await res.json();
72:     setItems(data.items || []);
73:   } catch (err) {
74:     console.error(err);
75:     toast.error("Failed to load items");
76:   } finally {
77:     setLoading(false);
78:   }
79: };
80:
81: // Trigger search when search or filter changes
82: useEffect(() => {
83:   const debounce = setTimeout(() => {
84:     fetchItems();
85:   }, 500);
86:   return () => clearTimeout(debounce);
87: }, [search, filter]);
88:
89: const handleClaim = async (itemId) => {
90:   setClaimingItem(true);
91:   try {
92:     const res = await fetch(
93:       `${API_BASE_URL}${API_ENDPOINTS.CLAIM_ITEM(itemId)}`,
94:       {
95:         method: "POST",
96:         credentials: "include",
97:         headers: {
98:           Accept: "application/json",
99:         },
100:       }
101:     );
102:
103:     const data = await res.json();
104:
105:     if (res.ok && data.success) {
106:       toast.success(data.message || "Item claimed successfully!");
107:       setShowClaimModal(false);
108:       fetchItems(); // Refresh items list
109:     } else {
110:       toast.error(data.message || "Failed to claim item.");
111:     }
112:   } catch (err) {
113:     console.error("Claim failed", err);
114:     toast.error("Something went wrong.");
115:   } finally {
116:     setClaimingItem(false);
117:   }
118: };
119:
120: const handleSendMessage = async () => {
121:   if (!currentItem || !messageText.trim()) {
122:     toast.error("Please enter a message");
123:     return;
124:   }
125:
126:   if (messageText.trim().length > 1000) {
127:     toast.error("Message cannot exceed 1000 characters");
128:     return;
129:   }
130:
131:   setSendingMessage(true);
132:
133:   try {
134:     const res = await fetch(`${API_BASE_URL}${API_ENDPOINTS.MESSAGES}`, {
135:       method: "POST",
136:       headers: {
137:         "Content-Type": "application/json",
138:         Accept: "application/json",
139:       },

```

```

140:     credentials: "include",
141:     body: JSON.stringify({
142:       receiverId: currentItem.createdBy,
143:       itemId: currentItem.id,
144:       message: messageText.trim(),
145:     }),
146:   });
147:
148:   const data = await res.json();
149:
150:   if (res.ok && data.success) {
151:     toast.success(data.message || "Message sent!");
152:     setShowMessageBox(false);
153:     setMessageText("");
154:     setCurrentItem(null);
155:   } else {
156:     toast.error(data.message || "Failed to send message.");
157:   }
158: } catch (err) {
159:   console.error("Message failed", err);
160:   toast.error("Something went wrong.");
161: } finally {
162:   setSendingMessage(false);
163: }
164: };
165:
166: const formatDate = (dateString) => {
167:   if (!dateString) return "N/A";
168:   const date = new Date(dateString);
169:   return date.toLocaleDateString("en-US", {
170:     year: "numeric",
171:     month: "short",
172:     day: "numeric",
173:     hour: "2-digit",
174:     minute: "2-digit",
175:   });
176: };
177:
178: if (loading) {
179:   return (
180:     <div className="min-h-screen flex items-center justify-center">
181:       <div className="text-center">
182:         <div className="animate-spin rounded-full h-12 w-12 border-b-2 border-blue-600 m
| x-auto mb-4"></div>
183:         <p className="text-gray-600">Loading items...</p>
184:       </div>
185:     </div>
186:   );
187: }
188:
189: return (
190:   <div className="max-w-6xl mx-auto p-6 space-y-6">
191:     <Toaster position="top-center" reverseOrder={false} />
192:
193:     <h1 className="text-3xl font-bold text-blue-600 text-center">
194:       Lost and Found Items
195:     </h1>
196:
197:     /* Search and Filter */
198:     <div className="flex flex-col sm:flex-row gap-4 justify-between items-center">
199:       <input
200:         type="text"
201:         placeholder="Search by name, location, or description..."
202:         className="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-none
| focus:ring-2 focus:ring-blue-500"
203:         value={search}
204:         onChange={(e) => setSearch(e.target.value)}
205:       />
206:       <select
207:         value={filter}
208:         onChange={(e) => setFilter(e.target.value)}
209:         className="px-4 py-2 border border-gray-300 rounded-lg focus:outline-none focus:
| ring-2 focus:ring-blue-500 min-w-[150px]"
210:       >
211:         <option value="">All Items</option>
212:         <option value="LOST">Lost</option>
213:         <option value="FOUND">Found</option>
214:         <option value="CLAIMED">Claimed</option>
215:       </select>
216:     </div>
217:
218:     /* Items Grid */
219:     <div className="grid gap-6 grid-cols-1 sm:grid-cols-2 lg:grid-cols-3">
220:       {items.map((item) => (
221:         <div

```



```

| unded-lg hover:bg-blue-50 transition font-medium"
304:      >
305:          Message Owner
306:          </button>
307:          ) : null}
308:      </div>
309:      </div>
310:      </div>
311:    )}}
312:  </div>
313:
314:  {items.length === 0 && (
315:    <div className="text-center py-20">
316:      <p className="text-gray-500 text-lg">No items found.</p>
317:      <p className="text-gray-400 text-sm mt-2">
318:        Try adjusting your search or filter criteria.
319:      </p>
320:    </div>
321:  )}

322:
323: /* Message Modal */
324: {showMessageBox && currentItem && (
325:   <div className="fixed inset-0 bg-black bg-opacity-50 px-4 flex items-center justify-
| y-center z-50">
326:     <div className="bg-white p-6 rounded-xl shadow-xl w-full max-w-md space-y-4">
327:       <h3 className="text-lg font-bold text-blue-600">Send Message</h3>
328:       <p className="text-sm text-gray-600">
329:         To: <strong>{currentItem.creatorName}</strong> <br />
330:         Item: <strong>{currentItem.name}</strong>
331:       </p>
332:       <textarea
333:         rows={4}
334:         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"
335:         placeholder="Type your message... (max 1000 characters)"
336:         value={messageText}
337:         onChange={(e) => setMessageText(e.target.value)}
338:         maxLength={1000}
339:       />
340:       <div className="text-xs text-gray-500 text-right">
341:         {messageText.length}/1000 characters
342:       </div>
343:       <div className="flex justify-end gap-2">
344:         <button
345:           onClick={() => {
346:             setShowMessageBox(false);
347:             setMessageText("");
348:             setCurrentItem(null);
349:           }}
350:           className="text-sm px-4 py-2 bg-gray-200 rounded-lg hover:bg-gray-300 tran
| sition"
351:         >
352:           Cancel
353:         </button>
354:         <button
355:           onClick={handleSendMessage}
356:           disabled={sendingMessage || !messageText.trim()}
357:           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"
358:         >
359:           {sendingMessage ? "Sending..." : "Send"}
360:         </button>
361:       </div>
362:     </div>
363:   </div>
364: }

365:
366: /* Claim Modal */
367: {showClaimModal && currentItem && (
368:   <div className="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-cen
| ter z-50 px-4">
369:     <div className="bg-white p-6 rounded-xl shadow-xl w-full max-w-md space-y-4">
370:       <h3 className="text-lg font-bold text-blue-600">
371:         Claim Item: {currentItem.name}
372:       </h3>
373:       <p className="text-sm text-gray-600">
374:         Your registered name and email will be used for this claim. The
375:         item owner will be notified.
376:       </p>
377:       <div className="flex justify-end gap-2">
378:         <button
379:           onClick={() => {
380:             setShowClaimModal(false);
381:             setCurrentItem(null);
382:           }}>

```

```

383:         className="text-sm px-4 py-2 border border-gray-300 rounded-lg hover:bg-gr
| ay-50 transition"
384:             >
385:                 Cancel
386:             </button>
387:             <button
388:                 onClick={() => handleClaim(currentItem.id)}
389:                 disabled={claimingItem}
390:                 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"
391:             >
392:                 {claimingItem ? "Claiming..." : "Confirm Claim"}
393:             </button>
394:         </div>
395:     </div>
396:     </div>
397:   )
398: </div>
399: );
400: }

```

---

## ■ 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">Welcom 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)(?=.*[^a-zA-Z\d]).{8,}$/;
35:     if (!passwordRegex.test(formData.password)) {
36:         toast.error(
37:             "Password must contain at least one uppercase letter, one lowercase letter, one nu
| mber, and one special character."
38:         );
39:         return;
40:     }
41:
42:     // Name validation
43:     if (formData.name.length < 2 || formData.name.length > 100) {
44:         toast.error("Name must be between 2 and 100 characters.");
45:         return;
46:     }
47:
48:     const nameRegex = /^[a-zA-Z\s]+$/;
49:     if (!nameRegex.test(formData.name)) {
50:         toast.error("Name should only contain letters and spaces.");
51:         return;
52:     }
53:
54:     setLoading(true);
55:
56:     try {
57:         const res = await fetch(`${
58:             API_BASE_URL
59:         }${API_ENDPOINTS.REGISTER}`, {
60:             method: "POST",
61:             headers: {
62:                 "Content-Type": "application/json",
63:             },
64:             credentials: "include", // Important for cookies
65:             body: JSON.stringify(formData),
66:         });
67:
68:         const data = await res.json();
69:
70:         if (!res.ok || !data.success) {
71:             // Handle validation errors
72:             if (data.errors) {
73:                 Object.values(data.errors).forEach((error) => toast.error(error));
74:             } else {
75:                 toast.error(data.message || "Registration failed.");
76:             }
77:             return;
78:         }
79:
80:         toast.success("Registration successful! Redirecting to home...");
81:
82:         setTimeout(() => navigate("/home"), 2000);
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-
111:             none focus:ring-2 focus:ring-blue-500 focus:border-transparent"
112:             onChange={handleChange}
113:             value={formData.name}
114:             required
115:         />
116:     </div>
117:
118:     <div>
119:         <label className="block text-sm font-medium text-gray-700 mb-1">
120:             Email Address
121:         </label>
122:         <input
123:             type="email"
124:             name="email"
125:             placeholder="your@email.com"
126:             className="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-
127:                 none focus:ring-2 focus:ring-blue-500 focus:border-transparent"
128:                 onChange={handleChange}
129:                 value={formData.email}
130:                 required
131:             />
132:         </div>
133:
134:         <div>
135:             <label className="block text-sm font-medium text-gray-700 mb-1">
136:                 Password
137:             </label>
138:             <input
139:                 type="password"
140:                 name="password"
141:                 placeholder="????????"
142:                 className="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-
143:                     none focus:ring-2 focus:ring-blue-500 focus:border-transparent"
144:                     onChange={handleChange}
145:                     value={formData.password}
146:                     required
147:                 />
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-
159:                         none focus:ring-2 focus:ring-blue-500 focus:border-transparent"
160:                         onChange={handleChange}
161:                         value={formData.confirmPassword}
162:                         required
163:                     />
164:             </div>
165:
166:             <button
167:                 type="submit"
168:                 disabled={loading}
169:                 className="w-full bg-blue-600 text-white py-2.5 rounded-lg font-medium hover:b-
170:                     g-blue-700 transition disabled:opacity-50 disabled:cursor-not-allowed"
171:                 >
172:                     {loading ? "Creating Account..." : "Create Account"}
173:                 </button>
174:             </form>
175:
176:             <div className="mt-6 text-center text-sm text-gray-600">
177:                 Already have an account?{" "}
178:                 <button
179:                     onClick={() => navigate("/")}>
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: };
```

---