



Experiment 2

Student Name: Mohit Sengar

Branch: CSE

Semester: 6th

Subject Name: Full Stack Development – II

UID: 23BCS10331

Section/Group: KRG 3-A

Date of Performance: 27/01/2026

Subject Code: 23CSH-309

1. Aim: To implement Single Page Application (SPA) routing in the EcoTrack application using React Router, secure application routes using protected routing with Context API-based authentication, and manage shared authentication state across components.

2. Objective:

- To configure client-side routing using React Router
- To implement SPA navigation without page reloads
- To protect routes using authentication-based route guards
- To manage shared authentication state using React Context API
- To implement login and logout functionality
- To restrict unauthorized access to protected pages
- To understand redirection logic in protected routing
- To analyze the role of Context API in state management

3. Implementation / Code:

Tools & Technologies Used:

- AWS Free Tier Account
- Web Browser (Google Chrome / Firefox)
- Amazon EC2 Service
- RDP Client (Microsoft Remote Desktop)
- Internet-enabled Laptop/Desktop

Implementation Description:

- The EcoTrack application is enhanced by implementing client-side routing using React Router, enabling seamless navigation between different pages without full page reloads.
- An authentication system is implemented using React Context API, which stores and manages the authentication state (isAuthenticated) across the entire application.
- A ProtectedRoute component is created to restrict access to sensitive pages such as Dashboard, Logs, and Data. If the user is not authenticated, they are automatically redirected to the Login page.
- Login functionality updates the authentication state using context, while logout functionality resets the authentication state and redirects the user back to the login page.

- This approach ensures secure navigation, centralized state management, and a smooth SPA user experience.

Sample Code Snippet:

```
Header.js U eco-track\src\components\Header.js\ Header
1 import {Link} from 'react-router-dom';
2 import { UserAuth } from '../context/AuthContext';
3
4 const Header = () => {
5   const { isAuthenticated, setIsAuthenticated } = UserAuth();
6
7   return (
8     <header style = {{ display: 'flex', justifyContent: 'space-between', alignItems: 'center', padding: '10px 20px', backgroundColor: 'white' }}>
9       <h1>EcoTrack</h1>
10      <nav>
11        <Link to="/" style = {{ color: 'white', marginRight: '1rem' }} > Dashboard </Link>
12        <Link to="/logs" style = {{ color: 'white', marginRight: '1rem' }} > Logs </Link>
13        <Link to="/login" style = {{ color: 'white', marginRight: '1rem' }} > Login </Link>
14      </nav>
15    </header>
16  );
17 };
18
19 export default Header;
20
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

To create a production build, use `npm run build`.

webpack compiled **successfully**

```
DashboardLayout.js U eco-track\src\pages\DashboardLayout.js\ DashboardLayout
1 import {Link, Outlet} from 'react-router-dom';
2
3 const DashboardLayout = () => {
4   return (
5     <div style = {{padding: "1rem"}}>
6       <h3>
7         Dashboard
8       </h3>
9
10      <nav>
11        <Link to="/summary">Summary</Link> | {" "}
12        <Link to="/analytics">Analytics</Link>
13      </nav>
14
15      <hr/>
16      <Outlet />
17    </div>
18  );
19 };
20
21
22
23 export default DashboardLayout;
```

```
AuthContext.js U eco-track\src\context\AuthContext.js AuthProvider
1 import { createContext, useContext, useState } from "react";
2
3 const AuthContext = createContext(null);
4
5 export const AuthProvider = ({ children }) => {
6   const [isAuthenticated, setIsAuthenticated] = useState(false);
7
8   return (
9     <AuthContext.Provider value = {{ isAuthenticated, setIsAuthenticated }} >
10       {children}
11     </AuthContext.Provider>
12   );
13 };
14
15
16 export const UserAuth = () => useContext(AuthContext);
17
```

```
login.js U eco-track\src\pages\login.js...
1 import React from "react";
2 import { UserAuth } from "../context/AuthContext";
3 import { useNavigate } from "react-router-dom";
4
5 const Login = () => {
6
7   const {setIsAuthenticated} = UserAuth();
8   const navigate = useNavigate();
9
10   const handleLogin = () => {
11     // Simulate authentication process
12     setIsAuthenticated(true);
13     navigate("/");
14   };
15
16   return (
17     <div>
18       <h2>Login</h2>
19       <button onClick={handleLogin}>Login</button>
20     </div>
21   );
22 };
23
24
25
26 export default Login;
27
28
```

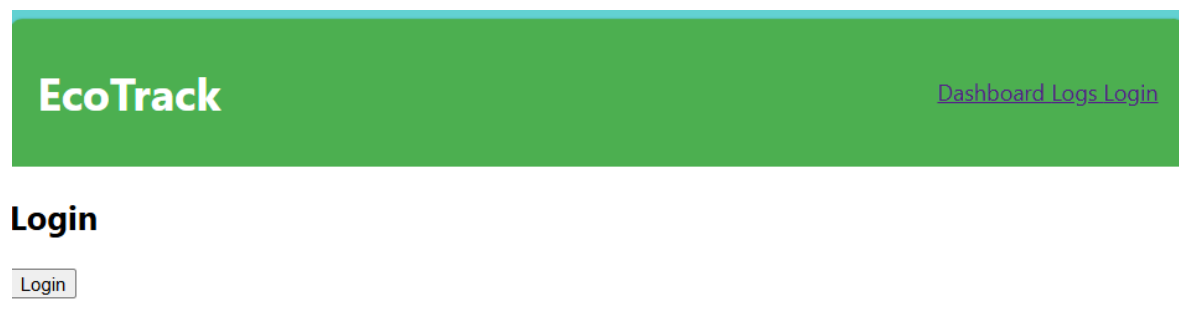
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

To create a production build, use `npm run build`.

webpack compiled **successfully**

4. Output:

- The EcoTrack application successfully implements SPA routing
- Navigation occurs without full page reloads
- Unauthorized users are redirected to the login page
- Authenticated users can access Dashboard, Logs, and Data pages
- System logs and environmental data are displayed dynamically
- Logout functionality securely ends the session
- Proper route protection is verified using ProtectedRoute



5. Learning Outcomes (What I Have Learnt)

After completing this experiment, the student is able to:

- Implement SPA routing using React Router
- Secure application routes using protected routing
- Manage shared authentication state using Context API
- Implement login and logout functionality
- Understand route redirection logic

Compare Context API with Redux at an introductory level

Build scalable and secure React applications



DEPARTMENT OF **COMPUTER SCIENCE & ENGINEERING**

Discover. Learn. Empower.