



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 2

Student Name: Yeshika

UID: 23BDA70132

Branch: AIT-CSE

Section: 23AIT_KRG-1

Semester: 6

Date of Performance: 23/01/2026

Subject Name: Full Stack II

Subject Code: 23CSH-382

- **Aim:**

To implement an authentication system and protected routing mechanism in a React application to control access to dashboard features and ensure secure navigation.

- **Objectives:**

The main objectives of this experiment are as follows:

1. To implement global authentication state management using React Context API.
2. To restrict access to protected pages using protected route components.
3. To create a nested dashboard layout for organizing multiple dashboard views.
4. To implement programmatic navigation using React Router hooks.
5. To configure advanced routing using React Router with nested and protected routes.

- **Implementation:**

The following general steps were followed to implement authentication and protected routing:

1. An authentication context was created in AuthContext.jsx to manage global login state using React Context API.
2. The authentication provider was wrapped around the application to make login state accessible throughout the app.
3. A protected routes component was created to restrict access to private pages for unauthenticated users.
4. Unauthenticated users were redirected to the Login page when trying to access protected routes.
5. A Login.jsx page was implemented to handle user authentication and update the global login state.
6. Programmatic navigation was implemented using React Router hooks to redirect users after login and logout actions.
7. A DashboardLayout.jsx component was created to provide a common dashboard layout with sub-navigation.
8. Nested routes were configured for Summary, Analytics, and Settings dashboard views.
9. Individual dashboard view components were created for summary, analytics, and settings sections.

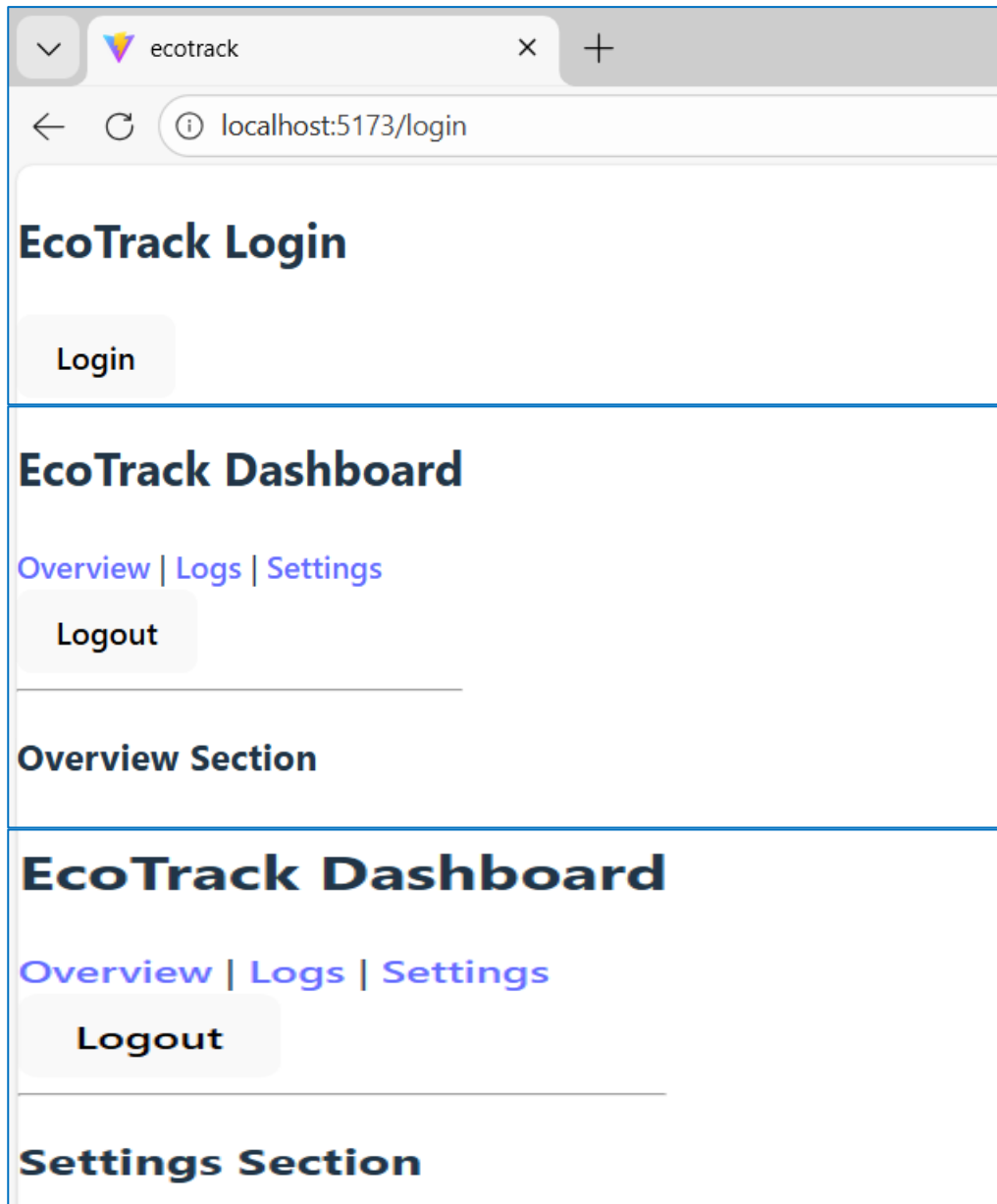


DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

10. A protected Logs page was implemented to display high-carbon activities.
11. The main routing structure was configured in App.jsx using BrowserRouter with protected and nested routes.

- **Output:**





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

- **Results:**

The authentication system and protected routing were successfully implemented. Unauthorized users were correctly redirected to the Login page. Nested dashboard views loaded properly within the main dashboard layout. The routing system verified secure access control and smooth navigation across the application.

- **Learning Outcomes:**

After completing this experiment, I have learnt to:

1. Implement authentication using React Context API for global state management.
2. Create protected routes to secure private components and pages.
3. Configure nested routing using React Router.
4. Use programmatic navigation for login and logout workflows.
5. Design structured dashboard layouts with multiple nested views.
6. Manage application-wide access control in a React application.