



Experiment 2

Student Name: Ashutosh Yadav

Branch: CSE

Semester: 6th

Subject Name: Full Stack Development – II

UID: 23BCS11023

Section/Group: KRG-3-A

Date of Performance: 20/01/2026

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:



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:

```
src > App.jsx > App
4   import { BrowserRouter,Route,Routes } from 'react-router-dom'
5   import ProtectedRoute from './routes/ProtectedRoute'
6   import DashboardSummary from './pages/DashboardSummary'
7   import DashboardAnalytics from './pages/DashboardAnalytics'
8   import Login from './pages/Login'
9   import Logs from './pages/Logs'
10
11  function App() {
12
13    return (
14      <main>
15        <>
16          <BrowserRouter>
17            <Header title={"EcoTrack"} />
18
19            <Routes>
20
21              <Route path="/login" element={<Login/>} />
22
23              <Route path="/" element={
24                <ProtectedRoute>
25                  <Dashboard/>
26                </ProtectedRoute>
27              }>
28
29              <Route index element={<DashboardSummary/>} />
30              <Route path="summary" element={<DashboardSummary/>} />
31              <Route path="analytics" element={<DashboardAnalytics/>} />
32
33            </Route>
34            <Route path="logs" element={<Logs/>} />
35
36          </Routes>
37
38        </BrowserRouter>
39      </main>
40    )
41  }
42
43
44
45 }
```

```
src > context >  AuthContext.jsx >  AuthProvider
1  import {createContext,useState,useContext} from "react";
2
3  const AuthContext = createContext(null);
4
5
6  export const AuthProvider= ({children}) =>{
7
8      const [isAuthenticated,setIsAuthenticated] = useState(false);
9
10
11     return (
12
13         <AuthContext.Provider value={{isAuthenticated,setIsAuthenticated}}>
14             {children}
15         </AuthContext.Provider>
16     )
17
18
19
20
21 }
22
23
24 export const useAuth = () => useContext(AuthContext);
```

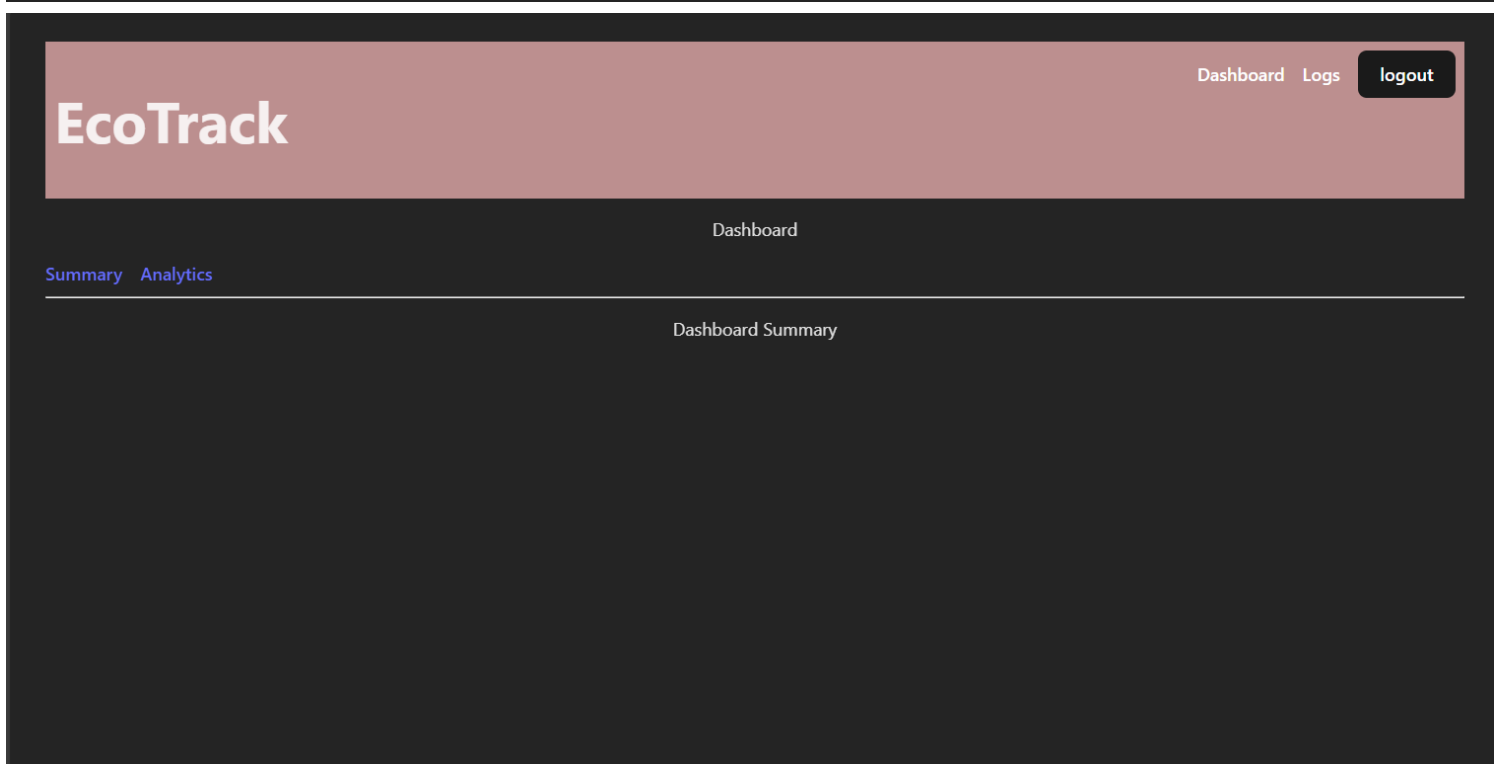
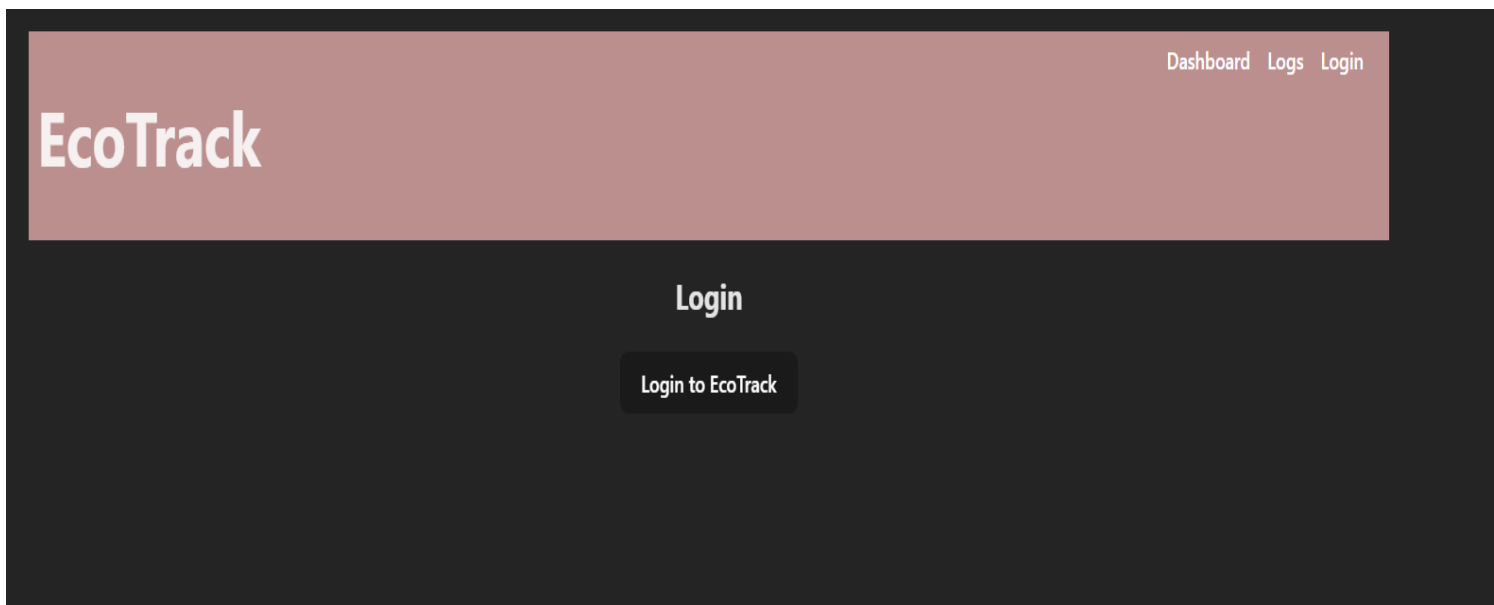
```
src > routes >  ProtectedRoute.jsx >  ProtectedRoute
1  import { Navigate, replace } from "react-router-dom";
2  import { useAuth } from "../context/AuthContext";
3
4  const ProtectedRoute = ({children} ) =>{
5
6      const {isAuthenticated} = useAuth();
7
8      if(!isAuthenticated) {
9          return <Navigate to="/login" replace />;
10     }
11
12
13     return children;
14
15 }
16
17 export default ProtectedRoute;
```

```
src > components > Header.jsx > Header
1
2 import { Link } from "react-router-dom";
3 import { useAuth } from "../context/AuthContext";
4 const Header = ({title}) =>{
5
6     const {isAuthenticated,setIsAuthenticated} = useAuth()
7     return [
8         <header style={{padding:"0.5rem",backgroundColor:"rosybrown",position:"sticky",top:"0",zInd
9
10         <h1>{title}</h1>
11
12         <nav>
13
14             <Link to="/" style={{marginRight:"1rem",color:"white"}} >
15             Dashboard
16             </Link>
17             <Link to="/logs" style={{marginRight:"1rem",color:"white"}} >
18             Logs
19             </Link>
20             {isAuthenticated ? <button onClick={()=>setIsAuthenticated(false)}> logout </button> :
21             <Link to="/login" style={{marginRight:"1rem",color:"white"}} >
22             Login
23             </Link>}
24
25         </nav>
26
27         </header>
28     ]
29 }
30
31 }
32
33
34 export default Header;
```



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





1. Learning Outcomes

- 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 applicatio