EXPERIMENT- 2

Student Name: ASHLIN JAMES

UID: 23BAI70722

Branch: BE CSE AIML

Section/Group: 23AIT_KRG-G1

Semester: 6

Date of Performance: 21/01/26

Subject Name: FULLSTACK-II

Subject Code: 23CSH-382

AIM:

To implement Single Page Application (SPA) navigation in the EcoTrack application using React Router, secure application routes using context-based authentication, and extend nested dashboard routing through follow-up enhancements.

OBJECTIVES:

After completing this experiment and its follow-up tasks, the student will be able to:

1.   Configure client-side routing in a React application using React Router

2.   Implement SPA navigation without full page reloads

3.   Design and apply protected routes using route-guard patterns

4.   Manage shared authentication state using React Context API

5.   Implement nested routing to build dashboard-style layouts

6.   Extend existing nested routes by adding new dashboard sections

7.   Implement logout functionality by updating shared context state

8.   Analyze route access behavior and explain redirection logic

9.   Understand the role of Context API in shared state management and its comparison with Redux at an introductory level

IMPLEMENTATION/CODE:

Logs.js

```
1   const logs = [
2       { id: 1, activity: "Car Travel", carbon: 4 },
3       { id: 2, activity: "Electricity Usage", carbon: 6 },
4       { id: 3, activity: "Cycling", carbon: 0 },
5   ];
6   export default logs;
```

Logs.jsx

```
1   import logs from "../data/logs";
2   const Logs = () => {
3       return (
4           <div>
5               <h1>Logs</h1>
6               <ol>
7                   {logs.map((log) => (
8                       <li key={log.id}>
9                           {log.activity}: {log.carbon} kg CO2
10                      </li>
11                  ))}
12              </ol>
13          </div>
14      )
15  }
16  export default Logs;
```

Login.jsx

```
1   import { useAuth } from "../context/AuthContext";
2   import { useNavigate } from "react-router-dom";
3   const Login = () => {
4       const {setIsAuthenticated} = useAuth();
5       const navigate = useNavigate();
6
7       const handleLogin = () => {
8           setIsAuthenticated(true);
9           navigate("/");
10      }
11      return (
12          <>
13          <h3>Login</h3>
14          <button onClick={handleLogin}>Login</button>
15          </>
16      )
17  }
18  export default Login;
```

Logout.jsx

```jsx
1  import { useAuth } from "../context/AuthContext";
2  import { useNavigate } from "react-router-dom";
3  import { useEffect } from "react";
4
5  const Logout = () => {
6    const { setIsAuthenticated } = useAuth();
7    const navigate = useNavigate();
8
9    useEffect(() => {
10     setIsAuthenticated(false);
11     navigate("/login");
12   }, [setIsAuthenticated, navigate]); // ✅ dependencies added
13
14   return null;
15 };
16 export default Logout;
```

App.jsx

```jsx
1  import { Route, Routes } from "react-router-dom";
2  import Login from "./pages/Login";
3  import DashboardAnalytics from "./pages/DashboardAnalytics";
4  import DashboardLayout from "./pages/DashboardLayout";
5  import DashboardSummary from "./pages/DashboardSummary";
6  import DashboardSettings from "./pages/DashboardSettings";
7  import ProtectedRoute from "./routes/ProtectedRoute";
8  import Logs from "./pages/Logs";
9  import Logout from "./pages/Logout";
10 import Header from "./components/Header";
11
12 function App() {
13   return (
14     <>
15       <Header />
16       <Routes>
17         <Route path="/login" element={<Login />} />
18         <Route path="/logout" element={<Logout />} />
19
20         <Route
21           path="/"
22           element={
23             <ProtectedRoute>
24               <DashboardLayout />
25             </ProtectedRoute>
26           }
```

```
27            >
28              <Route index element={<DashboardSummary />} />
29              <Route path="settings" element={<DashboardSettings />} |
30              <Route path="summary" element={<DashboardSummary />} />
31              <Route path="analytics" element={<DashboardAnalytics />
32            </Route>
33
34            <Route
35              path="/logs"
36              element={
37                <ProtectedRoute>
38                  <Logs />
39                </ProtectedRoute>
40              }
41            />
42          </Routes>
43        </>
44      );
45    }
46
47    export default App;
```

AuthContext.jsx

```
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, setIsAut
10                {children}
11            </AuthContext.Provider>
12         )
13    }
14
15    export const useAuth = () => useContext(AuthContext);
```

OUTPUT:

1)LOGIN

2)DASHBOARD SUMMARY

3)DASHBOARD ANALYTICS

4)LOGS

EcoTrack

Dashboard | Logs | Login | Logout

Logs

1. Car Travel: 4 kg CO2
2. Electricity Usage: 6 kg CO2
3. Cycling: 0 kg CO2
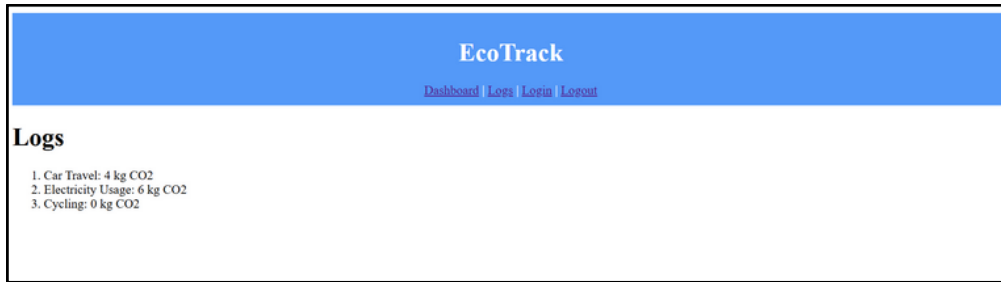
<u>LEARNING OUTCOME</u>S:

1. Gained a solid understanding of React authentication workflows, including login, logout, and protected routes.
2. Implemented the Context API to manage and share global authentication state across the application.
3. Secured application routes using guarded routes with React Router.
4. Utilized useNavigate() for programmatic navigation and secure user redirection.
5. Identified and resolved common React issues such as blank screens and ES module export errors.