Experiment 2

Student Name: Harsh   **UID:** 23BAI706474

Branch: BE CSE AIML   **Section/Group:** 23AIT-KRG G2

Semester: 6th   **Date of Performance:** January 2026

Subject Name: Full Stack   **Subject Code:** 23CSH-382

1. **Title:**

   Routing, Protected Routes & Context-Based State Management in EcoTrack

2. **Aim:**

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

3. **Objective:**

   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

4. **Implementation/Code:**

- AuthContext.jsx:

```jsx
import { createContext, useContext, useState } from "react";

const AuthContext = createContext(null);

export const AuthProvider = ({children}) => {
    const [isAuthenticated, setIsAuthenticated] = useState(false);

    return (
        <AuthContext.Provider value = {{isAuthenticated, setIsAuthenticated}}>
            {children}
        </AuthContext.Provider>
    )
}

export const useAuth = () => useContext(AuthContext);
```

- ProtectedRoute.jsx:

```jsx
import { Navigate } from "react-router-dom";
import { useAuth } from "../context/AuthContext";
import { children } from "react";

const ProtectedRoute = ({children}) => {
    const {isAuthenticated} = useAuth();

    if(!isAuthenticated) {
        return <Navigate to = "/login" replace/>
    }
    return children;
}

export default ProtectedRoute;
```

- Logs.jsx:

```jsx
import logs from "../data/logs";

const Logs = () => {
    return (
        <div>
            <h1>Logs</h1>
            <ol>
                {logs.map((log) => (
                    <li key={log.id}>
                        {log.activity}: {log.carbon} kg CO2
                    </li>
                ))}
            </ol>
        </div>
    )
}

export default Logs;
```

- Login.jsx:

```jsx
import { useAuth } from "../context/AuthContext";
import { useNavigate } from "react-router-dom";

const Login = () => {
    const {setIsAuthenticated} = useAuth();
    const navigate = useNavigate();

    const handleLogin = () => {
        setIsAuthenticated(true);
        navigate("/");
    }
    return (
        <>
        <h3>Login</h3>
        <button onClick={handleLogin}>Login</button>
        </>
    )
}
export default Login;
```

-

Logout.jsx:

```jsx
import { useEffect } from "react";
import { useNavigate } from "react-router-dom";
import { useAuth } from "../context/AuthContext";

const Logout = () => {
  const { setIsAuthenticated } = useAuth();
  const navigate = useNavigate();

  useEffect(() => {
    setIsAuthenticated(false);
    navigate("/login");
  }, [setIsAuthenticated, navigate]);

  return <p>Logging you out...</p>;
}

export default Logout;
```

- DashboardSettings, DashboardSummary, DashboardAnalytics:

-
```
const DashboardSettings = () => {
    return (
        <h3>These are the settings</h3>
    )
}

export default DashboardSettings;
```

```
const DashboardSummary = () => {
    return (
        <h3>This is a Summary</h3>
    )
}

export default DashboardSummary;
```

```
const DashboardAnalytics = () => {
    return (
        <h3>This is a Analysis</h3>
    )
}

export default DashboardAnalytics;
```

DashboardLayout.jsx:

```
import { Link, Outlet } from "react-router-dom";

const DashboardLayout = () => {
    return (
        <>
        <h3>Dashboard</h3>

        <nav>
            <Link to = "settings">Settings</Link>{" "}
            <Link to = "summary">Summmary</Link>{" "}
            <Link to = "analytics">Analytics</Link>
        </nav>

        <hr />
        <Outlet />
        </>

    )
}

export default DashboardLayout;
```

-
  - Header.jsx:

```jsx
import { Link } from "react-router-dom";
import { useAuth } from "../context/AuthContext";

const Header = () => {
    const {isAuthenticated} = useAuth();
    return (
        <header style = {{
            padding: '10px',
            backgroundColor: '#5499f8',
            color : 'white',
            textAlign: 'center',
        }}>
            <h1>EcoTrack</h1>
            <Link to = "/">Dashboard</Link>{" "}
            <Link to = "/logs">Logs</Link>{" "}
            {isAuthenticated ? (
                <Link to = "/logout">Logout</Link>
            ) : (
                <Link to = "/login">Login</Link>
            )}
        </header>
    )
}
export default Header;
```

App.jsx:

-

```jsx
import { Route, Routes } from "react-router-dom";
import Login from "./pages/Login";
import Logout from "./pages/Logout";
import DashboardAnalytics from "./pages/DashboardAnalytics";
import DashboardLayout from "./pages/DashboardLayout";
import DashboardSummary from "./pages/DashboardSummary";
import DashboardSettings from "./pages/DashboardSettings";
import ProtectedRoute from "./routes/ProtectedRoute";
import Logs from "./pages/logs";
import Header from "./components/Header";

function App() {
    return (
        <>
        <Header />
        <Routes>
            <Route path = "/Login" element = {<Login/>} />
            <Route path = "/Logout" element = {<Logout/>}/>
            <Route path = "/"
            element = {
                <ProtectedRoute>
                <DashboardLayout/>
                </ProtectedRoute>
            }>
            <Route index element = {<DashboardSummary/>}/>
            <Route path = "settings" element = {<DashboardSettings/>}/>
            <Route path = "summary" element = {<DashboardSummary/>}/>
            <Route path = "analytics" element = {<DashboardAnalytics/>}/>
            </Route>
            <Route path = "/logs"
            element = {
                <ProtectedRoute>
                <Logs/>
                </ProtectedRoute>
            }>
            </Route>
        </Routes>
        </>
    )
}
export default App;
```

-

5. Output

EcoTrack

Dashboard Logs Login

Login

Login

EcoTrack

Dashboard Logs Logout

Dashboard

Settings Summmary Analytics

This is a Summary

6. Learning Outcome

- We learnt about React Apps and how to create them.
- We learnt about reduce function and how it replaces map and filter.
- We learnt about the use of hooks.
- We learnt about Authentication and index pages
- We learnt the use of useContext and useState.
- We learnt about the flow of a React project.