



Experiment 2

Student Name: Tushar

Branch: AIT_CSE

Semester: 6th

Subject Name: Full Stack II

UID: 23BAI70332

Section/Group: 23AIT_KRG_G2

Date of Performance: 25/01/26

Subject Code: 23CSH-382

1. 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.

2. 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.

3. Implementation/Code:

=>App.jsx

```
import { BrowserRouter, Routes, Route, Navigate } from "react-router-dom";
import Header from "../components/Header";
import Login from "../pages/Login";
import Dashboard from "../pages/Dashboard";
import DashboardSummary from "../pages/DashboardSummary";
```

```

import DashboardAnalytics from "../pages/DashboardAnalytics";
import Logs from "../pages/Logs";
import ProtectedRoute from "../routes/ProtectedRoute";
import { useAuth } from "../context/AuthContext";

const HomeRedirect = () => {
  const { isAuthenticated } = useAuth();
  return (
    <Navigate to={isAuthenticated ? "/dashboard" : "/login"} replace />
  );
};

const App = () => {
  return (
    <BrowserRouter>
      <Header />

      <Routes>
        { /* Home */ }
        <Route path="/" element={ <HomeRedirect /> } />

        { /* Login */ }
        <Route path="/login" element={ <Login /> } />

        { /* Protected routes */ }
        <Route
          path="/dashboard"
          element={
            <ProtectedRoute>
              <Dashboard />
            </ProtectedRoute>
          }
        />

        <Route
          path="/dashboard/summary"
          element={
            <ProtectedRoute>
              <DashboardSummary />
            </ProtectedRoute>
          }
        />
      </Routes>
    </BrowserRouter>
  );
};

```

```

<Route
  path="/dashboard/analytics"
  element={
    <ProtectedRoute>
      <DashboardAnalytics />
    </ProtectedRoute>
  }
/>

```

```

<Route
  path="/logs"
  element={
    <ProtectedRoute>
      <Logs />
    </ProtectedRoute>
  }
/>

```

```

  { /* fallback */ }
  <Route path="*" element={<Navigate to="/" replace />} />
</Routes>
</BrowserRouter>
);
};

```

```
export default App;
```

=>ProtectedRoute.jsx

```

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

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

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

  return children;
}

```

```
};
```

```
export default ProtectedRoute;
```

⇒ AuthContext.js

```
import React, { createContext, useContext, useState } from "react";
const AuthContext = createContext();
export const AuthProvider = ({ children }) => {
  const [isAuthenticated, setIsAuthenticated] = useState(false);
  const login = () => setIsAuthenticated(true);
  const logout = () => setIsAuthenticated(false);
  return React.createElement(
    AuthContext.Provider,
    { value: { isAuthenticated, login, logout } },
    children
  );
};
export const useAuth = () => useContext(AuthContext);
```

⇒ Login.jsx

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

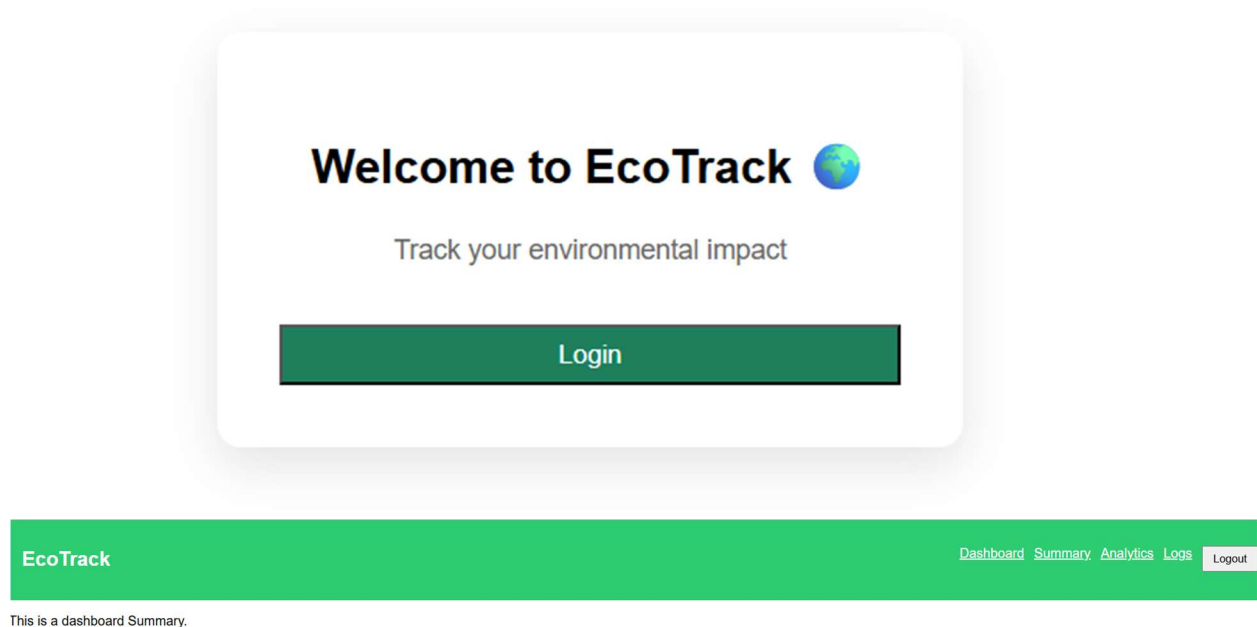
const Login = () => {
  const { login } = useAuth();
  const navigate = useNavigate();

  const handleLogin = () => {
    login();
    navigate("/dashboard");
  };

  return (
    <div style={{ padding: "2rem" }}>
      <h2>Login</h2>
      <button onClick={handleLogin}>Login</button>
    </div>
  );
};

export default Login;
```

4. Output



5. Learning Outcome

- Understood React project structure and component-based architecture
- Implemented routing and navigation using React Router
- Applied authentication using Context API
- Created protected routes for secure access
- Designed a clean and user-friendly UI