



Experiment 2

Student Name: Anuj Yadav

UID: 23BIA50011

Branch: BE-AIT-CSE

Section/Group: 23AML_KRG-G2

Semester: 6th

Date of Performance: 25 Jan 2026

Subject Name: Full Stack II

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:

- Configure client-side routing in a React application using React Router
- Implement SPA navigation without full page reloads
- Design and apply protected routes using route-guard patterns
- Manage shared authentication state using React Context API
- Implement nested routing to build dashboard-style layouts
- Extend existing nested routes by adding new dashboard sections
- Implement logout functionality by updating shared context state
- Analyze route access behavior and explain redirection logic
- Understand the role of Context API in shared state management and its comparison with Redux at an introductory level

3. Implementation/Code:

main.jsx:

```
import { StrictMode } from 'react'
import { createRoot } from 'react-dom/client'
import './index.css'
import { BrowserRouter } from 'react-router-dom'
import App from './App.jsx'
import { AuthProvider } from './context/AuthContext.jsx'

createRoot(document.getElementById('root')).render(
  <StrictMode>
    <BrowserRouter>
      <AuthProvider>
        <App />
      </AuthProvider>
    </BrowserRouter>
  </StrictMode>,
)
```

ProtectedRoute.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:

```
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:

```
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;
```

DashboardSummary.jsx:

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

DashboardLayout.jsx:

```
import { Link, Outlet } from "react-router-dom";  
  
✓ const DashboardLayout = () => {  
✓   return (  
✓     <>  
      <h3>Dashboard</h3>  
  
✓     <nav>  
        <Link to = "summary">Summmmary</Link>  
        <Link to = "analytics">Analytics</Link>  
      </nav>  
  
      <hr />  
      <Outlet />  
    </>  
  )  
}  
  
export default DashboardLayout;
```

DashboardAnalytics.jsx:

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

Logs.js;

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

AuthContext.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);
```

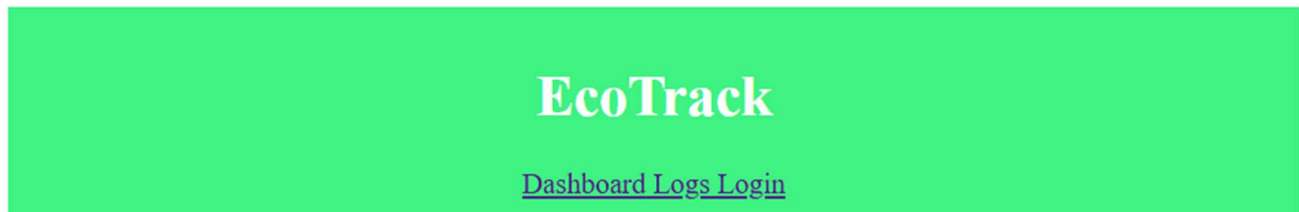
Header.jsx:

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

const Header = () => {
  return (
    <header style = {{
      padding: '10px',
      backgroundColor: '#3ef381',
      color : 'white',
      textAlign: 'center',
    }}>
      <h1>EcoTrack</h1>
      <Link to = "/">Dashboard</Link>
      <Link to = "/logs">Logs</Link>
      <Link to = "/login">Login</Link>
    </header>
  )
}

export default Header;
```

4. Output:



Login

Login

EcoTrack

[Dashboard](#) [Logs](#) [Login](#)

Dashboard

[Summary](#) [Analytics](#)

This is a Summary

5. Learning Outcome:

- **Implement client-side routing for seamless SPA navigation.**
- **Secure routes using context-based authentication and route guards.**
- **Manage shared state across components with React Context API.**
- **Build nested routes for modular dashboard layouts.**
- **Handle login and logout using context state.**
- **Analyze route access behavior and redirection logic.**
- **Compare Context API with Redux for state management.**
- **Extend dashboard features by adding new nested routes.**