



# CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

## Experiment 2

**Name:** Madhav

**Semester:** 6th

**UID:** 23BAI70107

**Date of Performance:** 21/01/2026

**Branch:** BE-CSE-AIML

**Subject Name:** FULLStack2

**Section/Group:** 23AIT\_KRG-1

**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:

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

### 3. Implementation/Code:

#### a. Routes.jsx

```
import React from 'react'
import { useAuth
} from '../context/auth' import {Navigate} from 'react-router-dom';
```

```
export default function AuthProtectedRoutes ({ children }) {
  const { isAuthenticated } = useAuth();

  return isAuthenticated ? children : <Navigate to="/login"/>
}
```

## b. Dashboard\_layout.jsx\

```
import React from 'react'
import { Outlet } from 'react-router-dom'

export default function Dashboard_layout() { return
(
  <>
    <h1>Dashboard_layout</h1>

    <Outlet/>
  </>
)
}
```

## c. Coxtest.jsx

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

const AuthContext = createContext(null); export co

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

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

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

4. Output:

Ecco Track

[Summary](#) [Analytics](#) [Login](#) [Logout](#)

Login

Login Now

Dashboard\_layout

DashboardSummary

Dashboard\_layout

DashboardAnalytics

## Ecco Track

[Summary](#) [Analytics](#) [Login](#) [Logout](#)

**Log Out**

[Logout Now](#)

### 5. Learning Outcome:

- a. I learnt how to make react routes.
- b. How the react routes actually works.
- c. I learnt how to use the outlet tags along with nested routes.