COLLEGE CODE : 9623
COLLEGE NAME : Amrita College Of Engineering And Technology
DEPARTMENT : Computer Science and Engineering
STUDENT NMD-ID :539E1EAAF72C0DEABDE4E70731865422

ROLL NO : 962323104701

DATE :13-10-2025

Completed the project named as phase_05_ReactJS routing with login protection

PROJECT_NAME : ReactJS routing with login protection

SUBMITTED BY,
NAME : sathya.S.V
MOBILE NO :6383901669

Project Overview

- The project “ReactJS Routing with Login Protection” demonstrates how to implement secure navigation between multiple pages in a React application.It focuses on React Router for page navigation and Login Authentication to restrict access to certain routes unless the user is logged in.

Key points:

- Built using ReactJS and React Router v6.
- Implements Login authentication using local storage/session storage.
- Protects routes by redirecting unauthorized users to the login page.
- Provides a clean user interface for login and navigation.
- Ensures better user experience and security in single-page applications (SPA).

Essential Topics Covered:

- React Components & Hooks (useState, useNavigate, useEffect)
- React Router (Navigation, Route Protection)

- Conditional Rendering
- Authentication & Route Guards

Project Report

The project begins with creating a ReactJS application using create-react-app.
The following steps outline the development process:

1. Project Setup:
    - Initialized a React project and installed required dependencies like react-router-dom.
2. Component Creation:
    - Login Component: Handles user authentication.
    - Home, Dashboard, About Pages: Accessible only after login.
    - ProtectedRoute Component: Checks if a user is authenticated before allowing access.

3. Routing:

- Implemented routes using BrowserRouter, Routes, and Route components.
- Added a PrivateRoute mechanism that prevents users from accessing protected pages without logging in.

4. Login Protection Logic:

- On successful login, a user token is stored in local storage.
- Each protected route checks for this token before rendering.
- If not found, the user is redirected to the login page.

5. Testing and Validation:

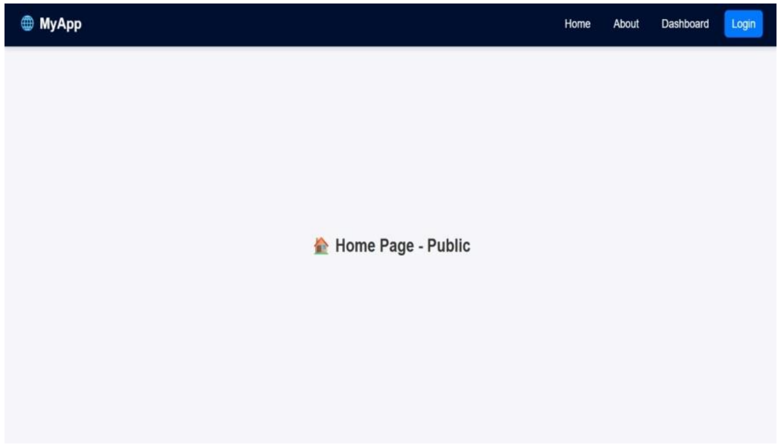
- Verified navigation between routes and ensured login protection works correctly.

Outcome:

- A fully functional ReactJS web app with secure routing and user authentication.

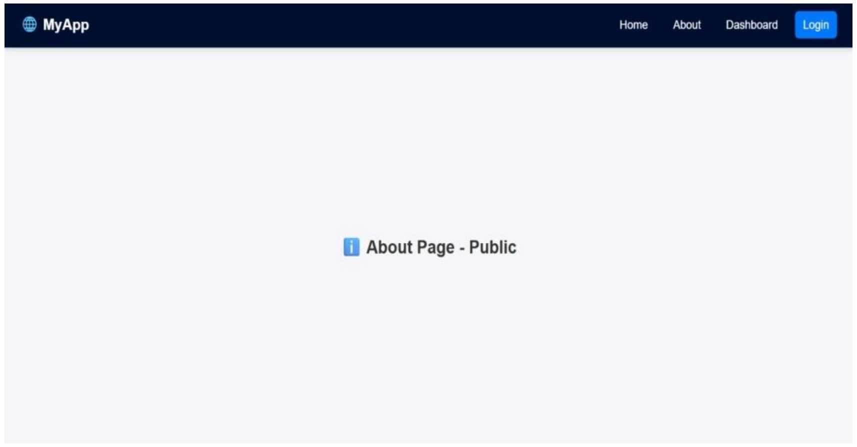
3. Screenshots

1)web version



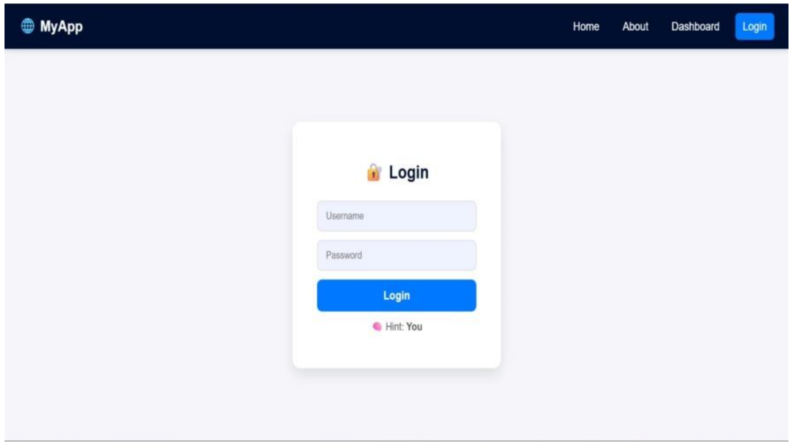
🏠 Home Page - Public

a) Home page



ℹ️ About Page - Public

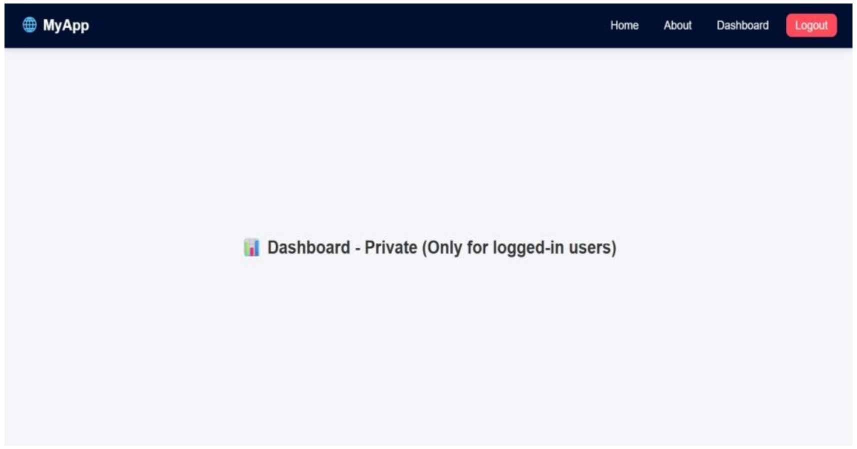
b) About page



🔒 Login

Hint: You

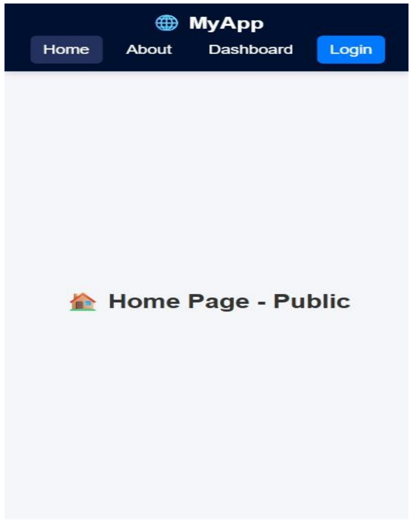
c) Dashboard page before Login



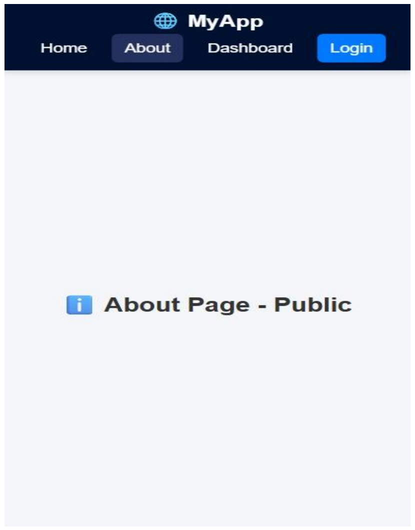
📊 Dashboard - Private (Only for logged-in users)

d) Dashboard page after Login

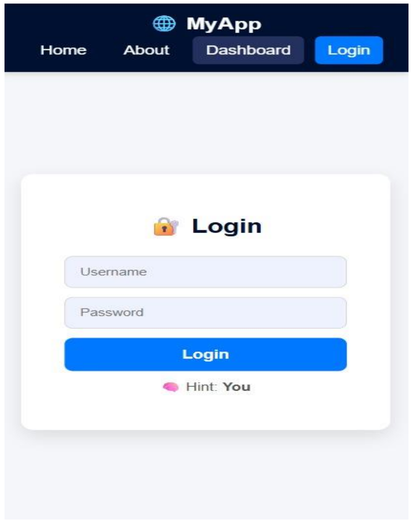
2)web app version



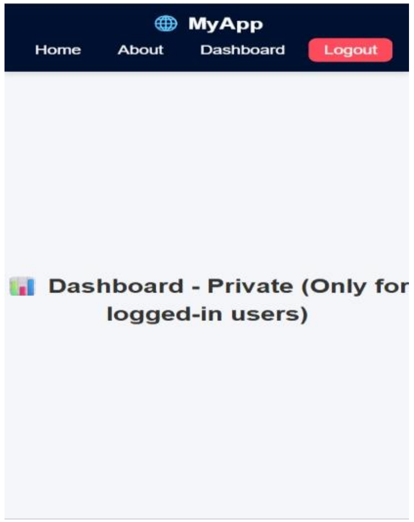
a)Home page



b)about page



c) Dashboard page before Login



d) Dashboard page after Login

4. Challenges & Solutions

Challenge	Solution
Implementing protected routes	Used custom ProtectedRoute component that checks authentication state before rendering.
Handling user session	Stored authentication status in local storage and verified it in each protected route.
Redirecting unauthenticated users	Utilized useNavigate hook from React Router to redirect to login.
State management	Managed app-wide login state using React hooks and props.

5. GitHub Link

https://github.com/BLUE-DEMON-7/NM-PROJECT-2-

Conclusion

This project demonstrates how ReactJS routing with login protection enhances user experience and security in modern web applications. It shows practical implementation of React Router, authentication, and state management, all essential concepts for real-world React development.

All the project phases (Phase 1 to Phase 5) have been uploaded to the GitHub repository mentioned above.