



COLLEGE CODE : 9623

COLLEGE NAME : Amrita College Of Engineering And Technology

DEPARTMENT : Computer Science and Engineering

STUDENT NMD-ID :539E1EAAF72C0DEABDE4E70731865422

ROLL NO : 962323104701

DATE :12-09-2025

Completed the project named as phase_01_ReactJS routing with login protection

NAME :ReactJS routing with login protection

SUBMITTED BY,

NAME :sathya.S.V

MOBILE NO :63839 01669

TITLE SLIDE

Build a Multi-Page React App with Route
Protection

- ◆ Using React Router & Authentication Logic

PROBLEM STATEMENT(PART 1)

- Most web apps need multiple pages (Home, About, Dashboard, Profile).
- Without routing, React apps feel limited to just one page.
- Users need smooth navigation without full page reloads.

PROBLEM STATEMENT(PART2)

- Some pages must be private (e.g., Dashboard, Profile).
- If routes are not protected, sensitive data may be exposed.
- A secure yet user-friendly authentication system is required.

USER AND STAKEHOLDERS(PART1)

- Primary Users
- End users who log in to access private content.
- Visitors exploring public pages like Home or About.

USERS AND STAKEHOLDERS(PART 2)

- Stakeholders
- Developers building scalable React apps.
- Businesses ensuring secure access to customer data.
- Admins who need to manage user access rights

USER STORIES(PART 1)

- “As a user, I want to...”
- Navigate between Home, About, Dashboard, and Login smoothly.
- Log in with one click and access my dashboard.

USER STORIES(PART2)

- “As a developer, I want to...”
- Protect certain routes so only logged-in users can access them.
- Redirect unauthorized users to the login page automatically.

MVP FEATURES(PART 1)

- Multi-page navigation with React Router.
- Public pages: Home, About, Login.
- Private pages: Dashboard.

MVP FEATURES(PART 2)

- Basic authentication (Login/Logout).
- ProtectedRoute wrapper for secure access.
- Redirect logic for unauthorized users.

WIREFRAMES (CONCEPTUAL)

- Navigation bar: links to Home, About, Dashboard, Login/Logout.
- Home Page: welcome text.
- Login Page: simple login button.
- Dashboard: visible only after login.

API ENDPOINT LIST (FOR EXTENSION)

- (Even though our MVP is frontend-only, future APIs can be planned)
- POST /login → User authentication.
- POST /logout → End session.
- GET /dashboard → Fetch private user data.

ACCEPTANCE CRITERIA PART 1

- User can navigate between pages without reloading.
- Public routes are always accessible.
- Private routes are hidden unless logged in.

ACCEPTANCE CRITERIA PART 2

- Unauthorized users are redirected to /login.
- Login button updates authentication state.
- Logout button resets authentication state.

FUTURE ENCHANTMENT

- Persistent login using localStorage.
- Role-based access (e.g., admin vs. normal user).
- Integration with real backend authentication (JWT, OAuth).

CLOSING SLIDE

- Outcome:
- A working React app with multiple pages.
- Smooth navigation using React Router.
- Protected routes with simple authentication.