Assignment 1: React Router
Awareness, Application, Mastery, Influence

Gaurab Wagle

July 11, 2025

Overview of the Assignment

For this assignment, Participants will learn to set up multi-page React applications using React Router. They will understand routing, navigation, route parameters, and nested routes.

GitHub Repository Link

Get the complete code and project details on my GitHub repository:
`https://github.com/gxaurab/Tangible/tree/main/Ass2/Assignment2`
   Deployed Site:
`https://tangible-production.up.railway.app/ass1/awareness`

1 Awareness

- **Rubric:** Navigate between simple pages using ¡Route¿ and ¡Link¿ components.

- **Proof of Work**: Build navigation between Home, Login, and Register pages.

1.1 Approach

I built a simple Routing between Home, Login and Register page.

1.2   Outcome

Figure 1: Awareness Section.

2   Application

- **Rubric:** Use dynamic route parameters and nested routes to display specific content.

- **Proof of Work:** Implement dynamic routes like /post/:id to display individual post details fetched from state or mock data.

2.1   Approach:

For this assignment, I an array of object that had id and the Name of the person, which when clicked would redirect me to the another page that was dynamic with the last url parameter as the object id. If you look closely, you can see there are two different titles.

I used the json placeholder API to retrieve a random text.
"https://jsonplaceholder.typicode.com/todos/"
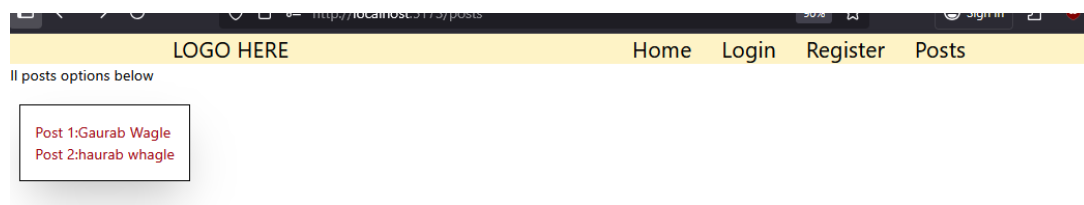
2.2   Outcome

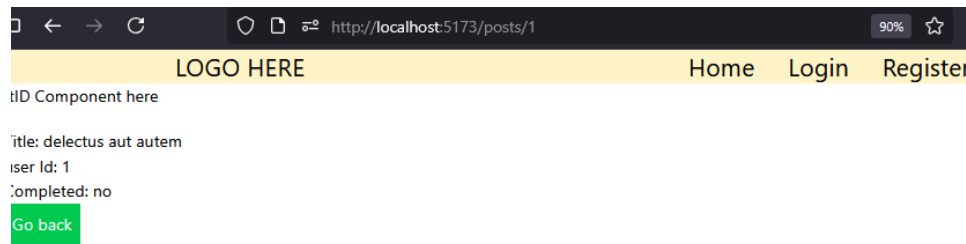Figure 2: Dynamic Components

Figure 3: Dynamic Id, 1

3   Mastery

- **Rubric:** Create protected routes that restrict access based on user authentication state.

- **Proof of Work:** Build an admin dashboard with auth-guarded routes requiring login.

3.1   Approach

For this assignment, I used **Firebase Authentication** and **Zustand** for auth state management.

When the user tries to access to the /dashboard route then they get redirected to the /login where they need to login via Google. After loggin in, they now can access to the dashboard route like shown in the figure below.
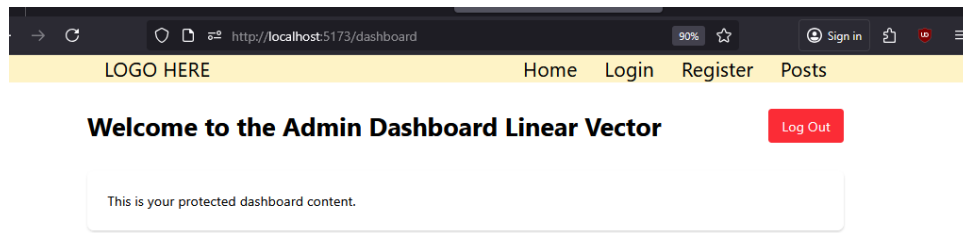
[H]

Figure 5: Dashboard Route being open when logged in
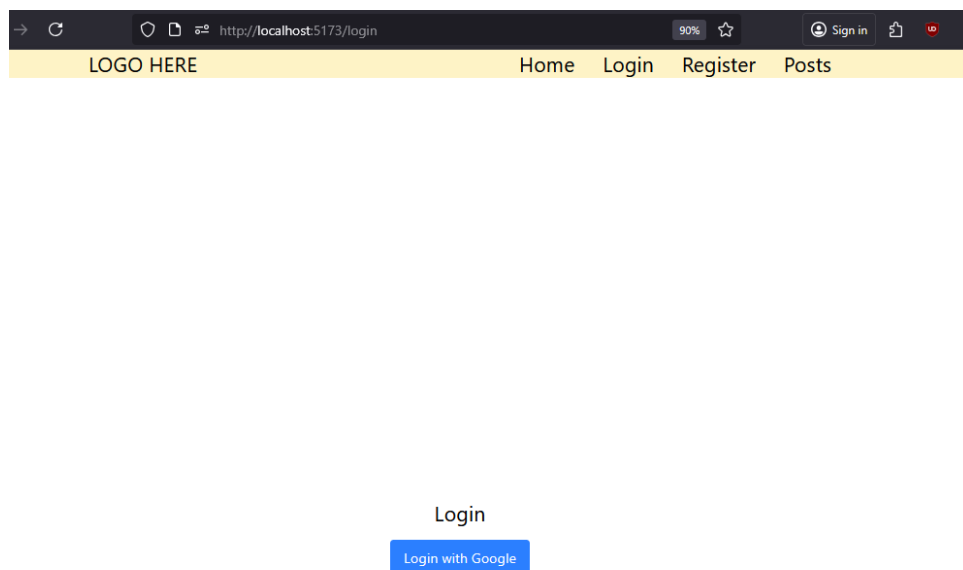
3.2   Outcome

Figure 4: "/dashboard" redirecting to the "/login" route

4   Influence

- **Rubric:** Optimize route-based code splitting and implement role-based layouts for different user types.

- **Proof of Work:** Develop a multi-role application with lazy loading and dynamic

navigation menus based on user roles.
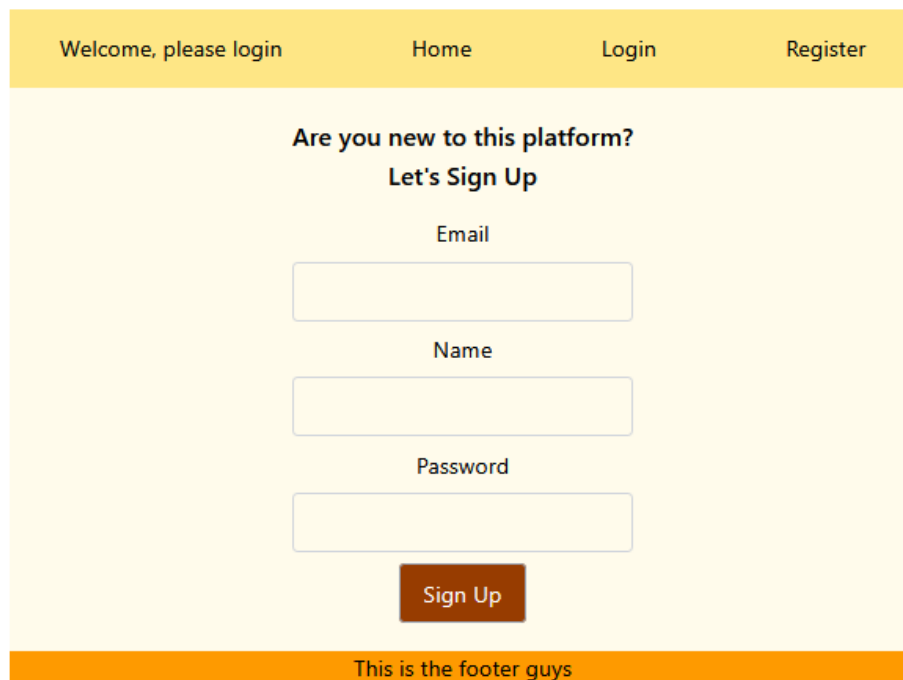
4.1   Approach

Workflow of the task:

- Home page with protected Navbar options.

- User Registers the account

- User Logs Inside

- If User is Admin, he sees a Dashboard Navigation, if a user, he is a normal person. Dashboard Navigation option is hidden

- Logs Out

4.2   What Happens Inside?

- User Registers: After registering the user, it then creates an entry in firestore with the user's UID, name, email and a default role of "user".

- User Logs In: When the user logs in, I then fetch the user detail from the firestore by comparing it with the UID and then the zustand is updated with the values: *role* and *name*.

- User now can access dashboard if his role is admin, else he is just exposed to the home page where his name is shown as greeting.

- When clicked the logout button, the zustand values are set to null.

4.3   Outcome

: Below is the Screenshot of the working Influence component.

Figure 6: Register

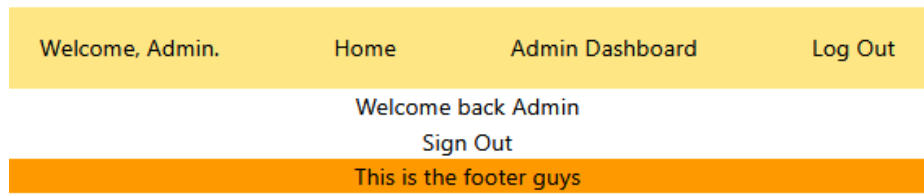
Figure 7:   Login Page

Figure 8: Admin Dashboard visible for admin

4.4   Conclusion:

Overall, this assignment helped me get a very good knowledge in the routing, zustand and protected Routes. I also had a good time with firebase and confident on it.