Frontend Assignment: SPA with Navigation and API Integration

Objective:

Create a Single Page Application (SPA) using a frontend framework or library of your choice (e.g., React, Vue, Angular, Svelte). The application will feature basic navigation, API integration, and dynamic routing.

Note: Using Angular v17 will be considered a plus, as it aligns with the technology stack used in our organization.

Requirements:

1. **Navigation:**
    - **The application should have two navigation items: "Home" and "Users List."**
    - **The navigation should be accessible from any page in the application.**
2. **Home Page:**
    - **When the "Home" navigation item is clicked, the user should be taken to the Home page.**
    - **The Home page should display a simple text: "Hello, World."**
3. **Users List Page:**
    - **When the "Users List" navigation item is clicked, the user should be taken to the Users List page.**
    - **The Users List page should display a table of users fetched from the Reqres API.**
        - **API Endpoint: https://reqres.in/api/users**
    - **The table should include at least the following columns:**
        - **ID**
        - **Name (First + Last Name)**
        - **Email**
4. **User Details Page:**
    - **When a user in the table is clicked, the application should navigate to a User Details page for that specific user.**
    - **The User Details page should display detailed information about the user, fetched from the Reqres API.**

        - API Endpoint: https://reqres.in/api/users/{id}
    - The details should include at least the following information:
        - User ID
        - First Name
        - Last Name
        - Email
        - Avatar (Profile Picture)
    - Include a "Back to Users List" button/link to navigate back to the Users List page.

Additional Guidelines:

- Routing: Implement client-side routing to manage the different views/pages (Home, Users List, User Details).
- Design: You can use any CSS framework (e.g., Bootstrap, Tailwind) or write custom styles, but the app should be visually clean and responsive.
- Error Handling: Implement basic error handling for the API requests, such as showing an error message if the data fails to load.
- Optional Enhancements:
    - Add a loading indicator while fetching data from the API.
    - Include pagination on the Users List page if desired.

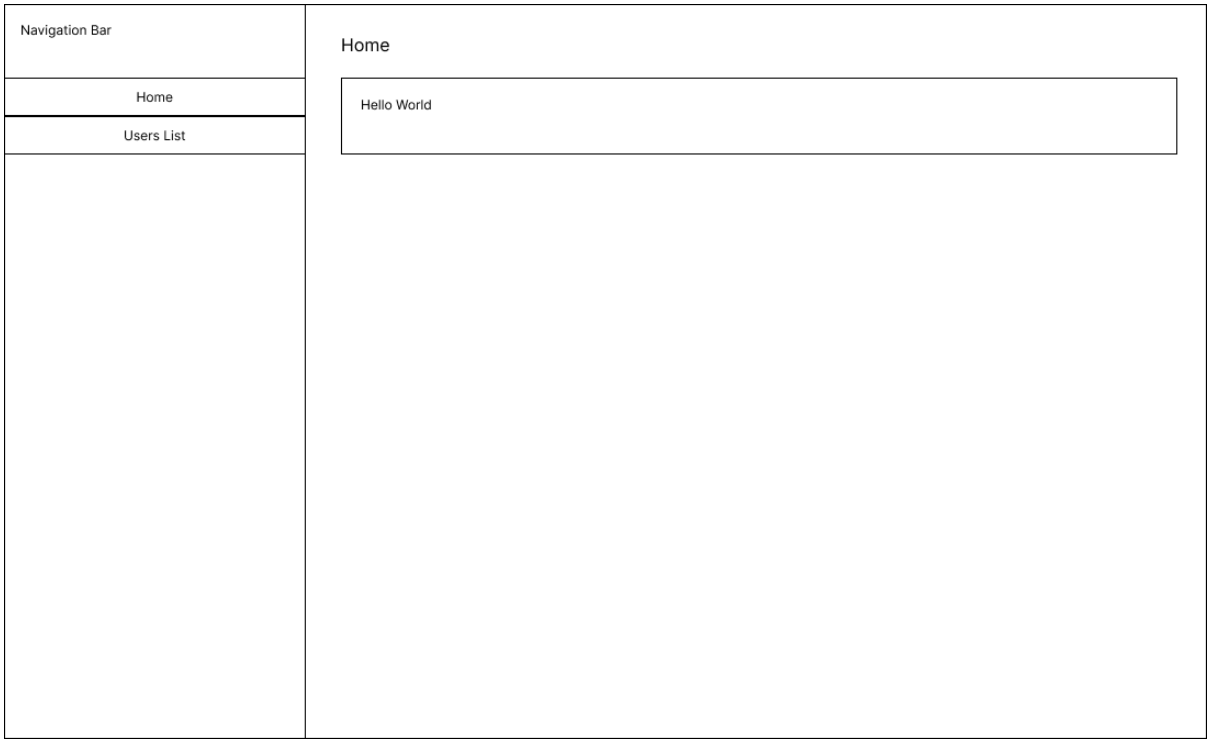
Mockups for Inspiration:

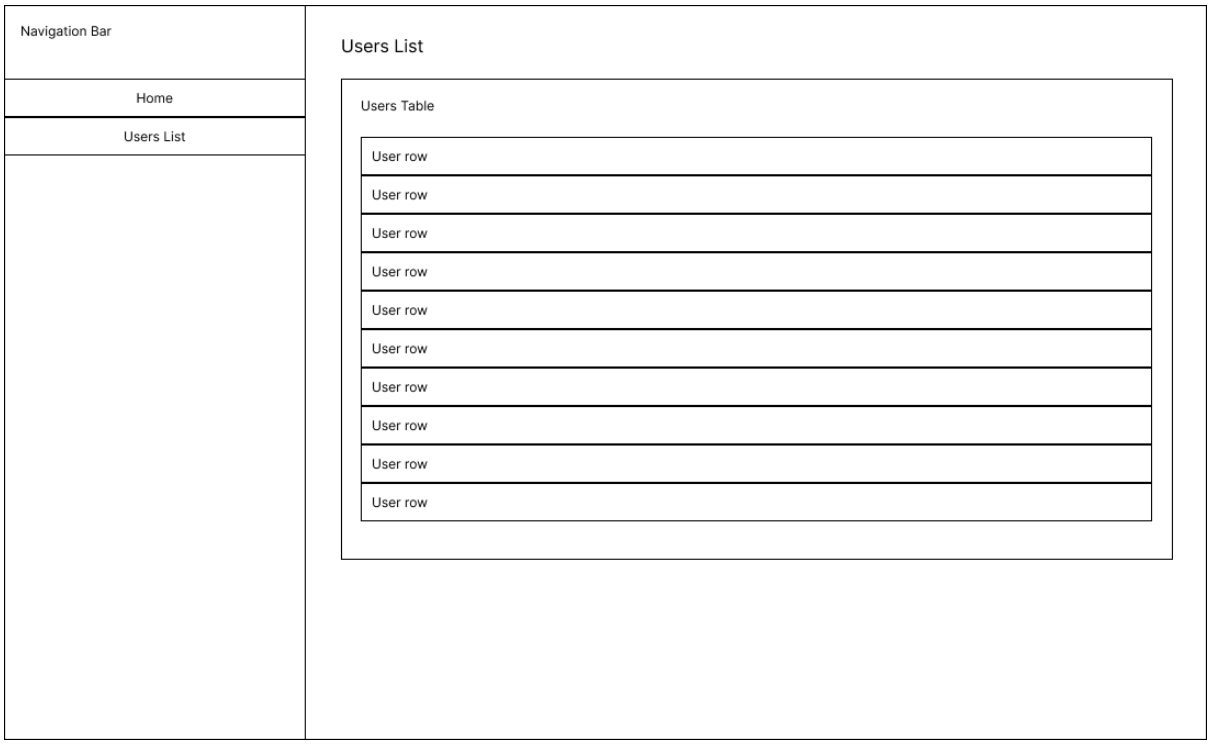
To help guide your design and implementation, we've included some mockups that illustrate the general layout and flow of the application. These mockups are meant to inspire you and provide a visual reference for what the final product might look like. Feel free to customize the design as you see fit, but ensure that the core functionality as described above is implemented.

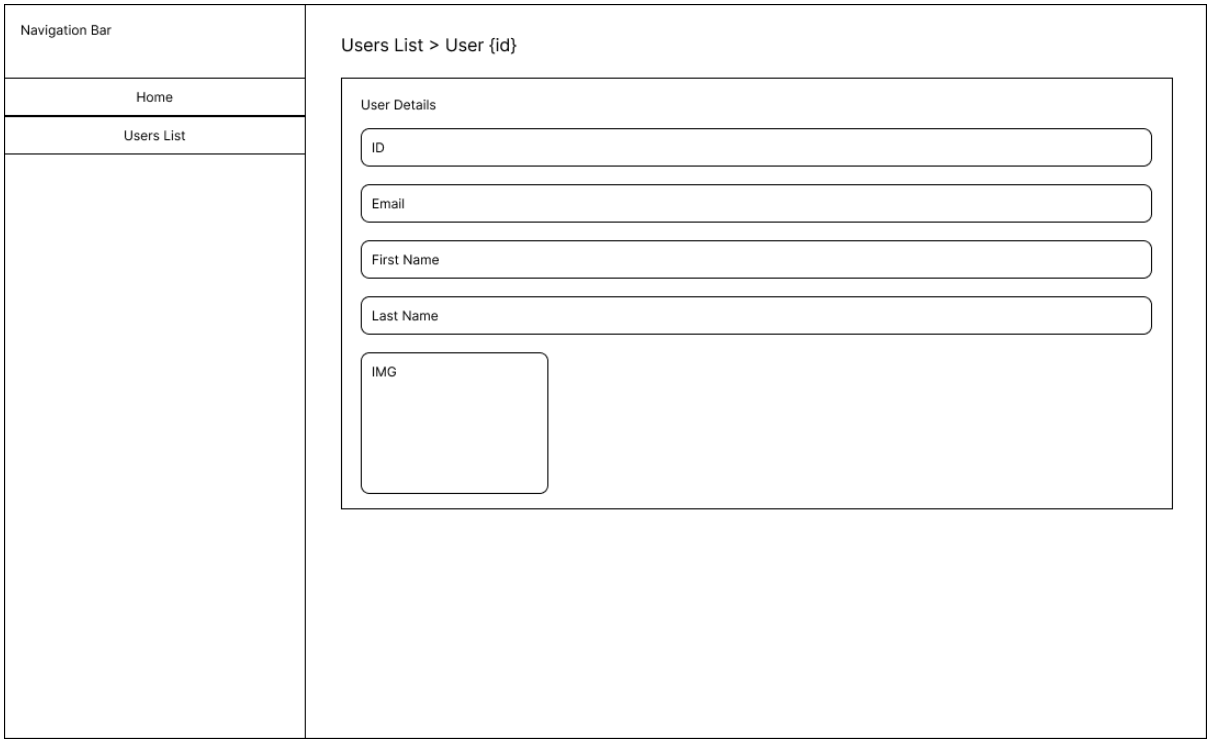
Submission:

- Provide a link to a GitHub repository with your code.
- Include instructions in the README on how to install dependencies and run the project locally.

This assignment is designed to showcase your ability to build a functional SPA with dynamic routing, API integration, and the use of modern frontend frameworks. Good luck!


Navigation Bar
Home
Users List
Home
Hello World


Navigation Bar
Home
Users List
Users List
Users Table
User row
User row
User row
User row
User row
User row
User row
User row
User row
User row


Navigation Bar
Home
Users List
Users List > User {id}
User Details
ID
Email
First Name
Last Name
IMG