

Task: Create an application with sidebar and a couple of routes as per the design attached below.

A. As a User, I want to be able to see my profile on the sidebar of the dashboard when I login.

B. I want to see the Blogs page along with sidebar visible through different pages of the application.

Task Requirements:

Frontend:

1. Create a simple 2-column layout that has a fixed sidebar (with user details and 2 links e.g., Dashboard and Blogs) and a content section.
2. Make an API call every time the app is loaded and call <http://localhost:6000/users/{random user between 1-10}> to fetch the user details and fill the sidebar user section e.g. <http://localhost:6000/users/2>
3. Make a custom hook to fetch the user's data from the user info API and use it on the components.
4. Implement 2 routes (Dashboard and Blogs) to change the content in the content section.
5. On Blogs page, make a REST API call to <http://localhost:6000/users/{random user}/posts> to fetch posts and display a list. E.g. <http://localhost:6000/users/1/posts>
6. List item should be clickable so that the details can be viewed on another page e.g., [app/posts/post-id](#) and should contain actions buttons like "Edit" and "Delete". For images, use any static image locally.
7. Clicking Edit should load that item in a form and saving after changing any text should make a PUT API call with the updated payload to <https://jsonplaceholder.typicode.com/posts>
8. Implement a simple unit test on sidebar to make sure that user detail is fetched.
9. Bonus Point: use <https://ant.design/> components for layout and sidebar etc.

Backend:

- Create a simple node application that has static sample data
- Create a GET service to serve the users list for frontend (<http://localhost:6000/users>)
- Create a GET to serve user details (<http://localhost:6000/users/1>)
- Create an endpoint to serve post for a user (<http://localhost:6000/users/1/posts>)
- Create a POST endpoint to accept post payload (<http://localhost:6000/users/1/post/1>)

Non-functional requirements:

- Make sure to use a .env file for Base API URLs.
- The system shall be implemented in React with Typescript and Styled-Components using Yarn.
- The final deliverable shall be shared with the Tech Lead through a Git repository with a readme file with instructions.

