



Frontend

Technical Task: Interactive User Profile Interface with Testing

Project Overview

You are tasked with developing a user profile management component for a social networking application. This component is essential for allowing users to view, edit, and manage their personal information.

Functional Requirements:

1. User Profile Display:

- Display user information including name, email, bio, and profile picture.
- Fetch user data from a mock API (<https://mswjs.io/>) using React Query.

2. Edit Profile Functionality:

- Enable users to edit their name, email, and bio.
- Validate form inputs for correctness (e.g., valid email format, non-empty name).
- Update user data locally or via a mock API submission.

3. Toggle Between Views:

- Implement a mechanism to toggle between view and edit modes.
- Ensure transitions are smooth and the interface remains responsive.

4. Unit and Integration Testing (optional):

- Write unit tests for each component (e.g., profile display, edit form).
- Develop integration tests that cover the at least one workflow

Technical and Design Specifications:

- **React and TypeScript** for application development.
- **React Query** for handling data fetching and updates.

- **Responsive Design** to ensure usability across devices.
- **Modern, User-friendly UI** with a focus on ease of use.
- **Jest, React Testing Library and Playwright** for writing and executing tests.
(optional)

Submission Guidelines:

- Host the project on GitHub with a clear commit history.
- Include a README detailing setup, functionality, and test execution.
- Provide explanations for design choices and the approach to testing.
- Deploy project to the hosting service and provide URL (optional)

Evaluation Criteria:

- **Adherence to Functional Requirements:** Complete implementation of the specified features.
- **Code Quality:** Well-structured, clean, and commented code.
- **User Interface:** Implementation of a responsive and intuitive UI.
- **Testing:** Comprehensive unit and integration tests demonstrating the reliability of the component (optional)
- **Best Practices:** Use of best practices in React, TypeScript, and testing methodologies.