Front-End Development Exercise: Todo List Application

Objective

Create a front-end only Todo List application using React, Next.js, React Hook Form, and Yup for form validation. The application should interact with a third-party API to manage todos, using react-query to handle API interactions.

Requirements

1. **Framework and Libraries:**
   - **React**: A JavaScript library for building user interfaces. It allows developers to create reusable UI components and manage the state of their applications effectively.
   - **Next.js**: A React framework that provides infrastructure and simple development experience.
   - **React Hook Form**: A library for managing form state and validation in React applications. It simplifies handling form inputs and reduces the need for boilerplate code.
   - **Yup**: A JavaScript schema builder for value parsing and validation. It is often used with form libraries like React Hook Form to define validation schemas.
   - **react-query**: A library for fetching, caching, and updating asynchronous data in React applications. It simplifies data fetching and helps manage server state.
2. **Data Management:**
   - Use the dummyjson API to get/create/update/delete todos.
   - Since this is a dummy api, create / update / delete won’t change anything on the list of todos. You will just need to test that the result of the api calls are successful or not.

3. **Features:**
    - Display a list of todos.
    - Create a new todo.
    - Update an existing todo.
    - Delete a todo.

Deliverables

- A Next.js project with the above components.
- A functional Todo List application that interacts with the dummyjson API.
- Clear and well-commented code.
- Proper separation of concerns, with API interactions, form handling, and UI components modularized appropriately.

Evaluation Criteria

- Correct use of React, Next.js, React Hook Form, Yup, and react-query.
- Proper interaction with the dummyjson API.
- Functionalities work as expected (create, read, update, delete todos).
- Code quality and organization, with clear separation of concerns.
- Use of best practices in React and Next.js development.

Steps to Complete the Exercise

1. **Setup Next.js Project with All Libraries Using npm**
2. **Create a List Page:**
    - Create a page where the user can see a list of todos retrieved from the API.
    - Fetch todos using react-query.
    - Display the todos in a list format.

3. **Create a Create Page:**
    - Create a page where the user can create a new todo.
    - Use React Hook Form and Yup for form handling and validation.
    - After successfully creating a todo, log the result of the api call or show a toast that say if there call was successful or if there was an error.
    - Redirect the user to the list page if the api call is successful.
4. **Create an Update Page:**
    - Create a page where the user can see the data of an existing todo in a form.
    - Use React Hook Form and Yup for form handling and validation.
    - Allow the user to update the todo and call the API to save changes. Log the result of the api call or show a toast that say if there call was successful or if there was an error.
    - Redirect the user to the list page if the api call is successful.
5. **Add a Delete Button on the Todo Card:**
    - On the list page, add a delete button on each todo card.
    - Call the API to remove the todo when the delete button is clicked. Log the result of the api call or show a toast that say if there call was successful or if there was an error.

GitHub Management

1. **Create a GitHub Repository:**
    - Create a new repository on GitHub for the Todo List application.
    - Add **ThomasNight** and **tahirNight** as collaborators on the repository so they can review the code.
        - To add collaborators, go to the repository settings on GitHub, navigate to the "Collaborators" section, and add their GitHub usernames.

2. **Commit Guidelines:**
    - Make a commit for at minimum each completed step in the exercise.
    - Use clear and descriptive commit messages that reflect the changes made.
    - Example commit messages: