Assessment Task: Building a User Management Dashboard

Overview:

You are tasked with building a simple User Management Dashboard using React.js and Next.js. You will use a fake online REST API to fetch and display user data. The dashboard should include features for viewing a list of users, adding new users, updating user information, and deleting users.

Requirements:

Setup Project: Create a new Next.js project. You can use create-next-app.

1. **API Integration:** Use a fake online REST API for user data. You can use a service like JSONPlaceholder (https://jsonplaceholder.typicode.com/) or any other fake REST API you prefer. The API should include endpoints for getting users, adding users, updating user information, and deleting users.

2. **User List:** Create a page to display a list of users. Fetch and display user data from the API on this page. Each user in the list should display their name, email, and a button to view more details.

3. **Pagination:** Introduce user pagination on the user list page, limiting the number of users displayed per page and allowing users to navigate through different pages.

4. **Sorting and Filtering:** Allow users to sort and filter the user list based on various criteria, such as name, email, or registration date. Use state management to manage dynamic sorting and filtering options.

5. **User Details:** Create a user details page that displays more information about a selected user, including their name, email, phone number, and any other relevant data from the API. Include a button to return to the user list.

6. **Add User:** Implement a page or form for adding new users. Include fields for name, email, and any other required information. When a user is added, send a POST request to the API and display the newly added user in the user list.

7. **Update User:** Implement a page or form for updating user information. Users should be able to edit the name and email of an existing user. When changes are saved, send a PUT request to the API to update the user's information.

8. **Delete User:** Implement a feature to delete a user. This can be done with a confirmation dialog. When a user is deleted, send a DELETE request to the API, and update the user list accordingly.

9. **Styling:** Apply basic CSS styles to make the dashboard visually appealing and user-friendly. You can use CSS frameworks like Tailwind CSS or Material-UI or write your custom CSS.

10. **Routing:** Use Next.js routing to navigate between different pages (e.g., user list, user details, add user, update user).
11. **Error Handling:** Implement basic error handling for API requests and display appropriate error messages to the user.

Evaluation:
Code structure and organization.
Proper usage of React and Next.js.
API integration and data fetching.
CRUD (Create, Read, Update, Delete) functionality.
Error handling and validation.
Use of version control (e.g., Git).

Deliverables:
A GitHub repository containing the code for the project.
A live demo of the User Management Dashboard (you can deploy it to Vercel or another hosting service).