

Task: User Management Application

Objective: Create a CRUD (Create, Read, Update, Delete) application using React to manage users. Use the JSONPlaceholder API (<https://jsonplaceholder.typicode.com/>) to fetch, create, update, and delete user data.

Tasks:

1. **Fetch Users:**
 - Fetch a list of users from the JSONPlaceholder API.
 - Display the users in a list or table format on the page.
 - Each user should display basic information.
2. **Create User:**
 - Implement a form to create a new user with proper validations.
 - Use a drawer or modal to display the form.
 - Perform a POST request to the JSONPlaceholder API to simulate creating a new user.
 - Update the UI with the new user data returned from the API.
3. **Update User:**
 - Each user should have an "Edit" button.
 - Clicking "Edit" should open a drawer or modal with the user's data pre-filled.
 - Allow the user to update fields.
 - Perform a PUT request to the JSONPlaceholder API to simulate updating the user.
 - Reflect the updated user data in the UI.
4. **Delete User:**
 - Each user should have a "Delete" button.
 - Clicking "Delete" should open a confirmation modal.
 - Perform a DELETE request to the JSONPlaceholder API to simulate deleting the user.
 - Remove the deleted user from the UI.

Additional Requirements:

1. **Table UI:**
 - Create a table based ui for showing the users details in different columns.
 - The table displaying the users should have alternating row colors [example](#).
2. **React Functional Components & Hooks:**
 - Utilize React functional components and Hooks (useState, useEffect, etc.).
3. **Routing using react-router-dom:**
 - Home view to list all users.
 - Detailed view for each user.
4. **Responsive Design:**

- Ensure the application is styled and has a responsive layout (works well on both desktop and mobile).
5. **Form Validations:**
- Display appropriate validation messages for each field.
 - **Name:** Required, minimum 3 characters.
 - **Email:** Required, must be a valid email format.
 - **Phone:** Required, must be a valid phone number.
 - **Username:** Required, minimum 3 characters. The field should be auto filled with the format (**USER-name**) and should be a non-editable field. When updating a user, the username should remain the same.
 - **Address (Street, City):** Required.
 - **Company Name:** Optional, but if provided, must be at least 3 characters.
 - **Website:** Optional, must be a valid URL if provided.
6. **Error Handling:**
- Implement proper error handling.
 - Notify users if any API request fails (e.g., using toasts or alerts).
7. **Code Quality:**
- Comment your code for clarity.
 - Ensure smooth and intuitive user interactions.

Optional Advanced Features (Bonus Points):

- Implement a search functionality to allow users name and filter out the user from the table.
- Implement a loading spinner or skeleton screen while fetching data.
- Use TypeScript for type safety throughout the application.

Deployment:

- Push the code to a GitHub repository.
- Deploy the application using a service like Vercel or Netlify.
- Provide the live URL for the hosted application.