EmployWise Assignment

Assignment Overview:

You are tasked with creating a **React application** that integrates with the **Reqres API** to perform basic user management functions. The assignment is divided into three levels of increasing complexity.

BASE URL = https://reqres.in/

Level 1: Authentication Screen

- Create a basic authentication screen where the user can log in using credentials.
- Use the following endpoint for authentication:
  - **POST** `/api/login` with `email` and `password` in the body.
  - Email : eve.holt@reqres.in
  - Password : cityslicka
- On successful login, store the token (returned by the API) and navigate to the **Users List** page (Level 2).

Level 2: List All Users

- After logging in, display a paginated list of users.
- Use the following endpoint to fetch user data:
  - **GET** `/api/users?page=1`
- Display the user's first name, last name, and avatar in a structured layout (table or card format).
- Implement pagination to navigate through different pages of users.
- Pagination or lazy loading both are accepted

Level 3: Edit, Delete, and Update Users

- Each user in the list should have options to **Edit** or **Delete** their details.
- **Edit**:
  - Clicking **Edit** should open a form pre-filled with the user's data.
  - Allow updating the user's first name, last name, and email.
  - Use the following endpoint for updates:
    - **PUT** `/api/users/{id}`
- **Delete**:
  - Clicking **Delete** should remove the user from the list.
  - Use the following endpoint for deletion:
    - **DELETE** `/api/users/{id}`
- Show appropriate success or error messages based on the outcome of each operation.

Rules and Guidelines:

1. **Framework & Libraries**:
    - You must use **React** as the frontend framework.
    - Feel free to use state management tools like **Redux** or **Context API**, but it is not mandatory.
    - For HTTP requests, use **Axios** or the native `fetch` API.
2. **User Interface**:
    - The UI should be user-friendly and responsive (work well on both desktop and mobile).
    - Use any CSS framework (e.g., **Bootstrap**, **Material-UI**, **Tailwind CSS**), or write custom CSS.
3. **Error Handling**:
    - Handle API errors gracefully, displaying appropriate error messages to the user.
    - Ensure form validation, especially in the login and edit screens.
4. **Persistence**:
    - Persist the login token in local storage or session storage.
    - Redirect the user to the login page if the token is missing or expired.
5. **Code Quality**:
    - Write clean, modular, and well-documented code.
    - Ensure proper usage of React components, hooks, and lifecycle methods.
6. **Submission**:
    - Provide a **GitHub repository** link containing your code.
    - Include a **README** file explaining how to run the project, install dependencies, and any assumptions or considerations made.
7. **AI Usage**
    - You are permitted to use ChatGPT, but you'll earn extra points if you complete the assignment on your own.
8. **Bonus Points**:
    - Implement client-side search and filtering on the users' list.
    - Use **React Router** for navigation between pages (Login, User List, Edit User).
    - Host on any free server like heroku or any other. Provide the link in the readme file.