

Note : You will be disqualified by default if the code is copied directly from AI tools like chatGPT. This is a test of your knowledge, not of AI's.

Task 1: Build a React User Dashboard with Goal Tracking

Create a **React.js dashboard** for managing users and tracking their goals with the following features:

1. **User List:** Display a list of users with name, email, and the number of goals they have.
2. **Search and Sort:** Add a search bar to filter users by name or email and allow sorting by name.
3. **Add User:** Include a form to add a new user (name, email).
4. **User Details:** Clicking on a user shows their goals in a sidebar or modal, with each goal showing its title, deadline, and status (e.g., "In Progress," "Completed").
5. **Goal Tracking Dashboard:** Add a summary section displaying total goals, completed goals, and a percentage of completion for all users combined.

Ensure the app is responsive, visually appealing, and uses React hooks for state management. Submit the project via GitHub with a README that includes setup instructions and details on your design decisions.

Submission Must Have's:

- Share via GitHub the code.
 - Share the hosted application url, feel free to choose any free hosting like vercel.
 - Attach the screenshot of your app in Github ReadMe.
-

Task 2: Backend Application

Develop a **Spring Boot backend** with APIs for user onboarding.

Requirements:

1. **APIs:**
 - **Sign Up:** User registration.
 - **Sign In:** User authentication.
 - **Forgot Password:** Mock password reset functionality.
2. **Database:**
 - Use an **H2 in-memory database** with tables for storing user details.
3. **Code Quality:**
 - Include **unit tests** for all APIs.
 - Follow clean coding practices with modular structure and error handling.
4. **Documentation:**

- Clearly explain how to run the application and test the APIs (e.g., via Postman).

Submission:

- Share via GitHub or as a downloadable zip file with setup instructions.
-

Evaluation Criteria

1. **Frontend:** Modern, Neat UI
2. **Backend:** API functionality, unit test coverage, and clean, maintainable code.
3. **Documentation:** Clarity and completeness for both tasks.

Note : Do not be discouraged even if you don't get this right 100%, submit whatever best you can achieve as long as it is presented in ReadMe clearly with screenshots.