

Day 3 Task: React + Redux Dashboard

Setup

1. Create a new project using **Vite + React + TypeScript**.
 2. Install dependencies:
 - @reduxjs/toolkit
 - react-redux
 3. Set up your Redux store and connect it to your React app with the `<Provider>` component.
-

Login Page

1. Create a login page with a form for username and password.
 2. Store a predefined username and password locally (no API needed).
 3. On successful login, navigate to the dashboard.
 4. Manage the login state globally with Redux (auth slice).
 5. Display username in the dashboard header when logged in.
 6. Add a logout button that clears the authentication state.
-

Dashboard

After login, render a dashboard containing **four cards** that demonstrate Redux usage.

Counter Card

- Show a numeric value (the counter).
- Add buttons to increment and reset.
- The counter state must be stored in Redux.

Timer Card

- Can **Start**, **Pause**, and **Reset**.
- Display elapsed seconds.
- When started, it should increment time every second.
- Timer running state and value must be stored in Redux.

Notes Card

- Allow the user to add notes.
- Each note can be marked as **Done**.
- When a note is done, change its color to green and apply a strikethrough style.

- Notes should be managed in Redux as an array of objects.

Users List Card (New)

- Fetch users from `https://jsonplaceholder.typicode.com/users`.
- Display their names and emails in a list.
- Handle loading and error states.
- Store fetched users in Redux using `createAsyncThunk`.
- add an input that can fetch user by id and if not found (response.status === 404) show not found string and if the input is empty just fetch the full user list.