

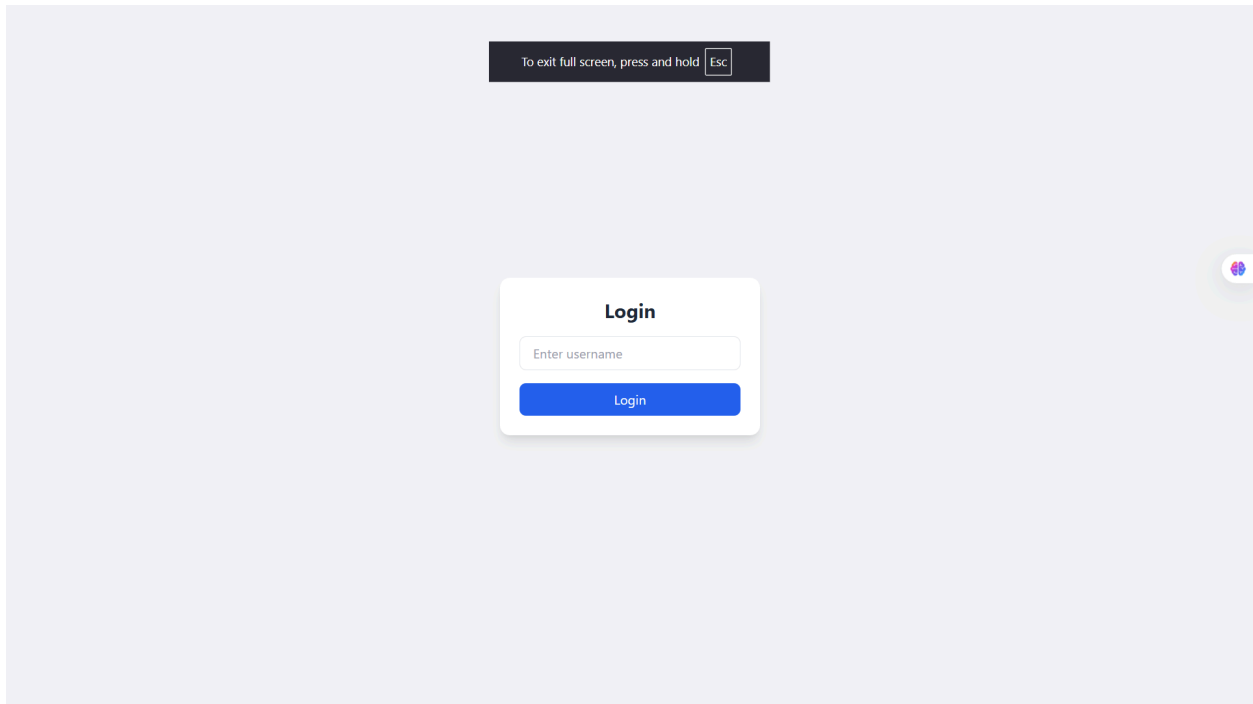
Task 3: Redux User Authentication

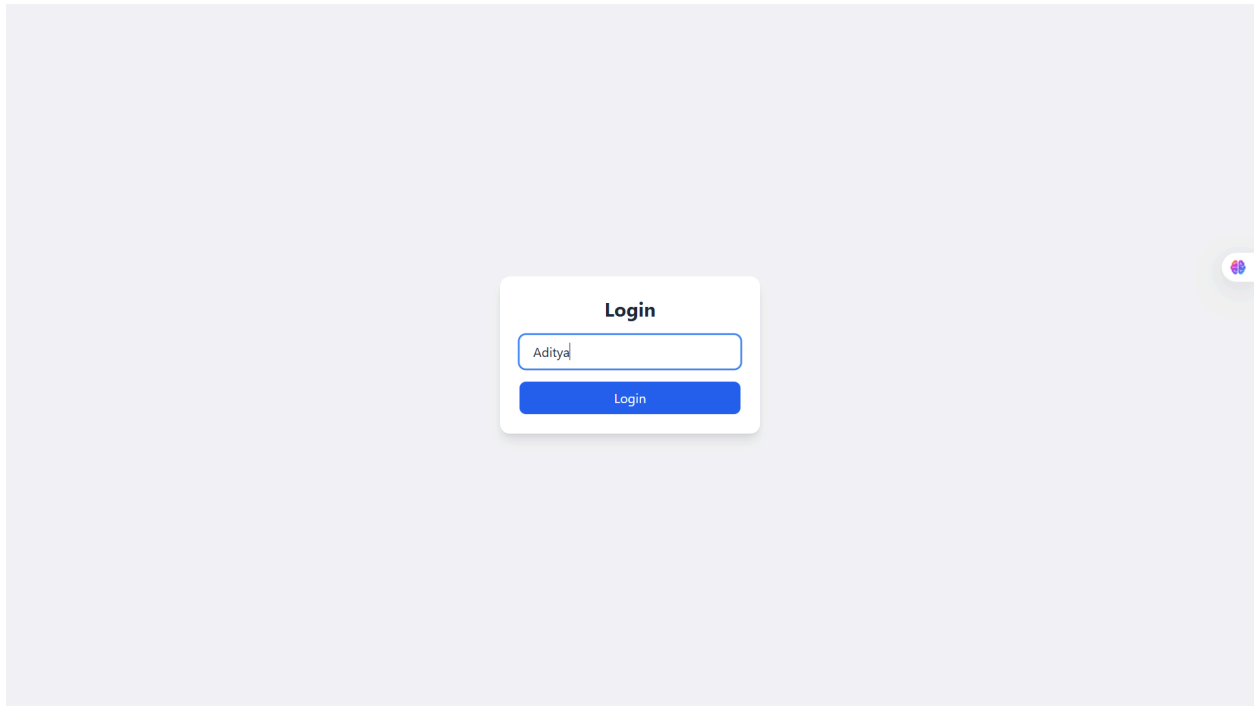
1. Task Description

A simple React project demonstrating **user authentication flow** using **Redux Toolkit** for global state management and **TailwindCSS** for styling.

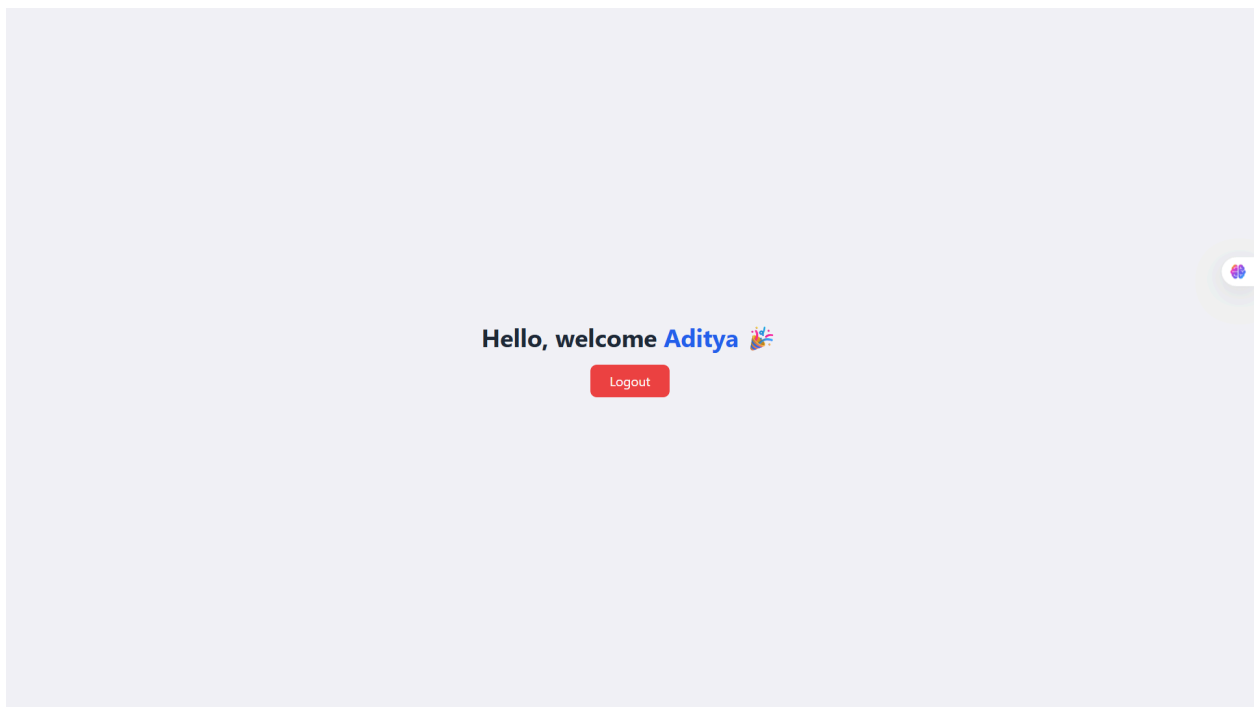
2. Task Output Screenshot

Before Login





After Login



3 . File Structure and Approach

I structured the project in a modular and scalable way, using Redux Toolkit for global state management and TailwindCSS for styling.

1. redux/authSlice.js

This slice manages authentication state (user info and login status). Contains actions like login, logout. Stores username in state after login.

2. redux/store.js

Configures the Redux store using configureStore. Combines the authSlice reducer. Makes the global state accessible across the app.

3. pages/Login.jsx

Represents the login form. On form submission, dispatches the login action with the entered username. After successful login, redirects the user to the Home page. Styled using TailwindCSS for a clean UI.

4. pages/Home.jsx

Initially shows a “Login to access features” message and a Login button that redirects to the login page.

5. App.jsx

The root component where routes are defined using React Router DOM. Wraps the app with Provider from react-redux so that Redux state is available globally. Handles navigation between Home and Login pages.