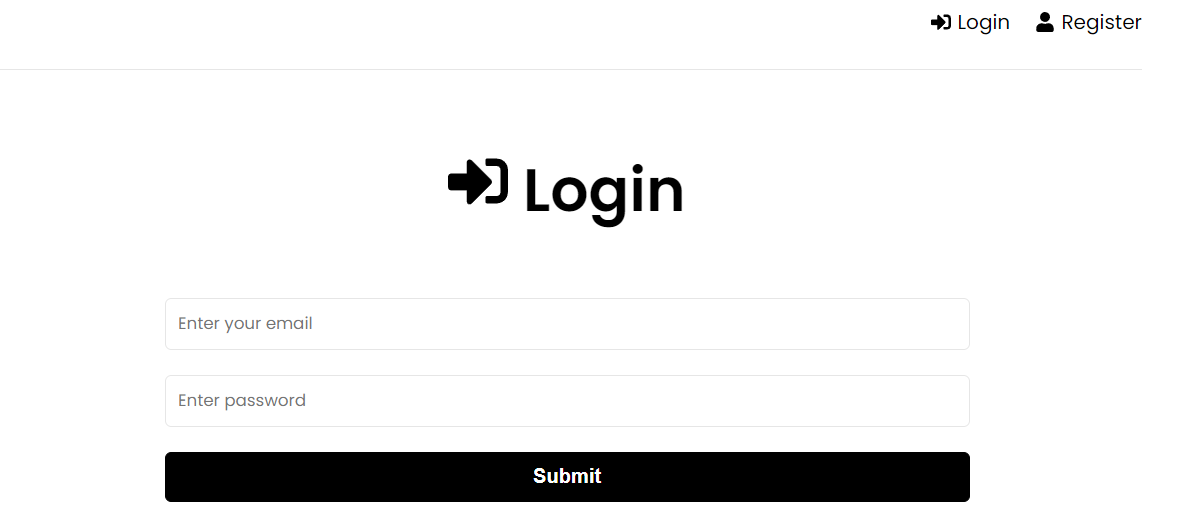
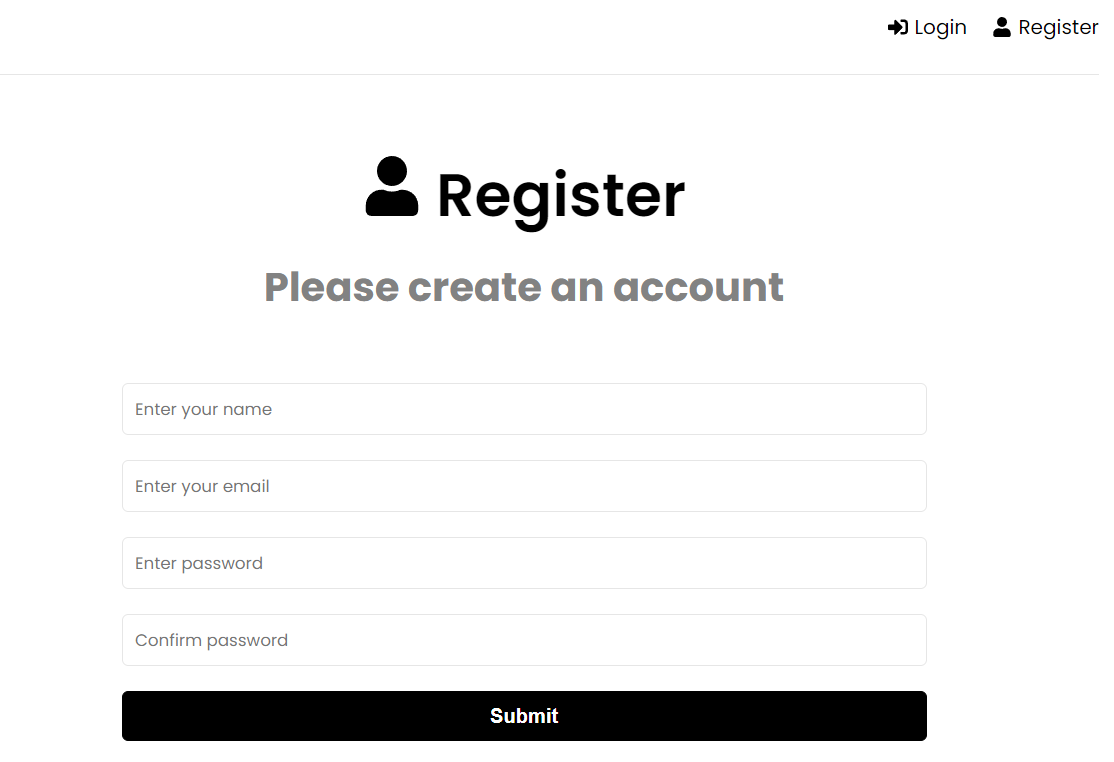
**Redux auth routing**

**Develop a Redux store to manage user authentication.**

In a React-Redux application, **authorization** and **routing** are crucial for controlling access to different parts of the application based on the user's authentication state and role. Integrating **Redux** for state management helps manage user authentication status across the application, while **React Router** allows for efficient navigation. Together, they create a secure and structured flow.

**Output Screenshot**





**Widget Used In Task**

* **Login/Logout Button Widgets:** Displays a login or logout button depending on the user’s authentication state.
* **Loading Spinner Widgets**: Display a loading spinner while checking the authentication state or fetching user data.
* **Error Message Widgets**:Display error messages if the user encounters issues during authentication (e.g., wrong password, unauthorized access).
* **Redirect Widgets**:Automatically redirect users based on their authentication state (e.g., from the login page to the dashboard after a successful login).