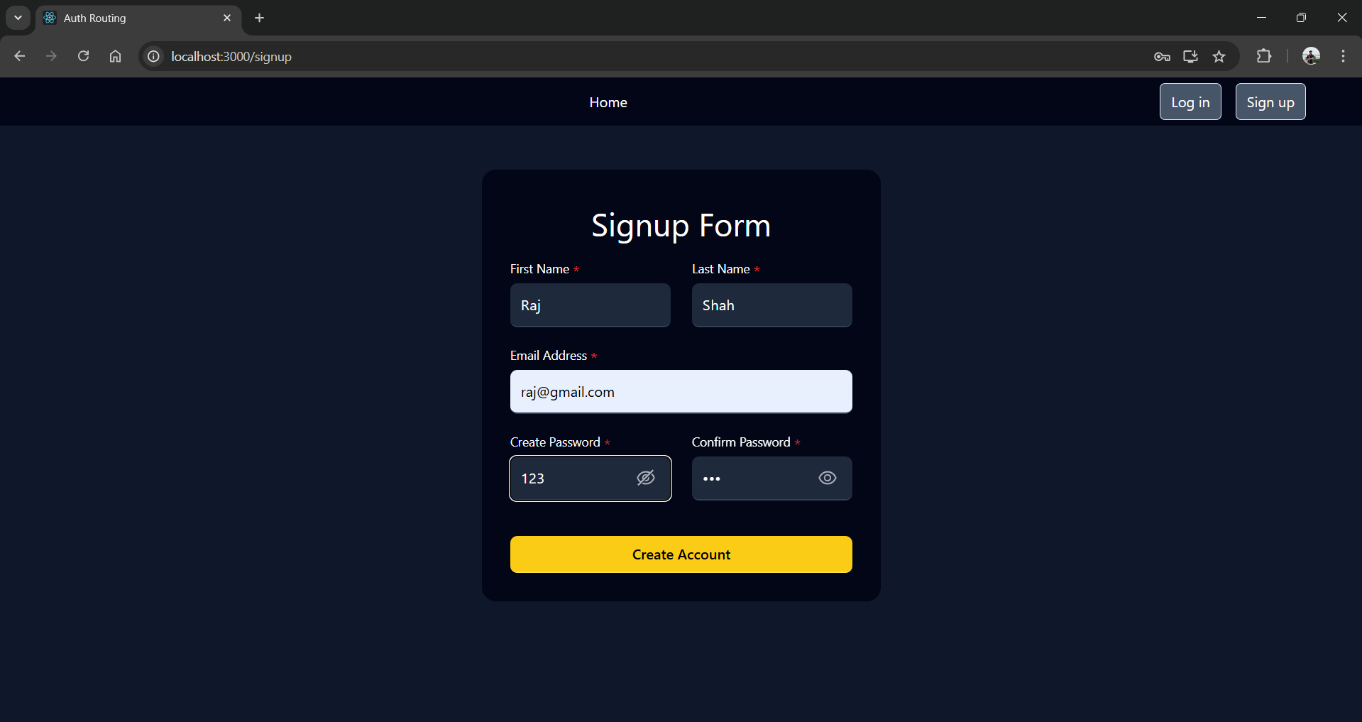
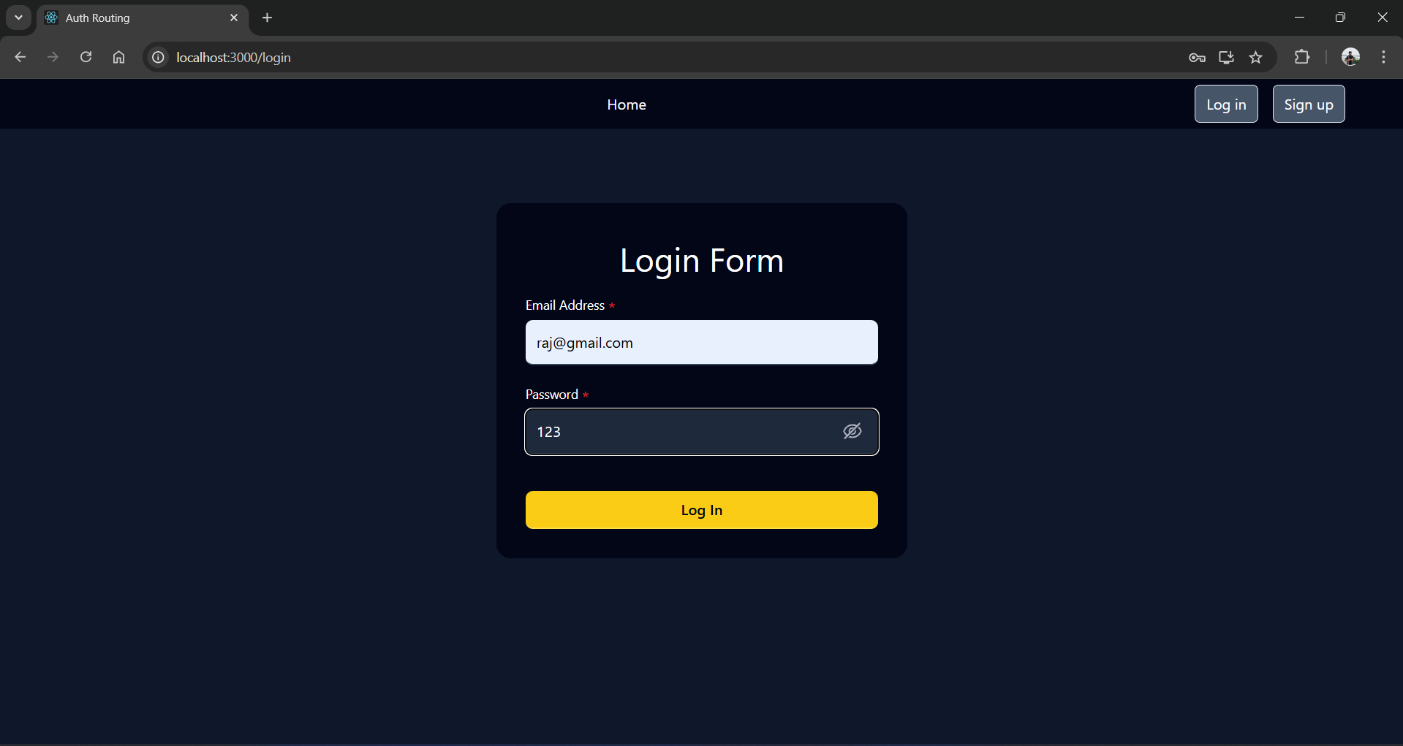
**Task : Auth Routing**

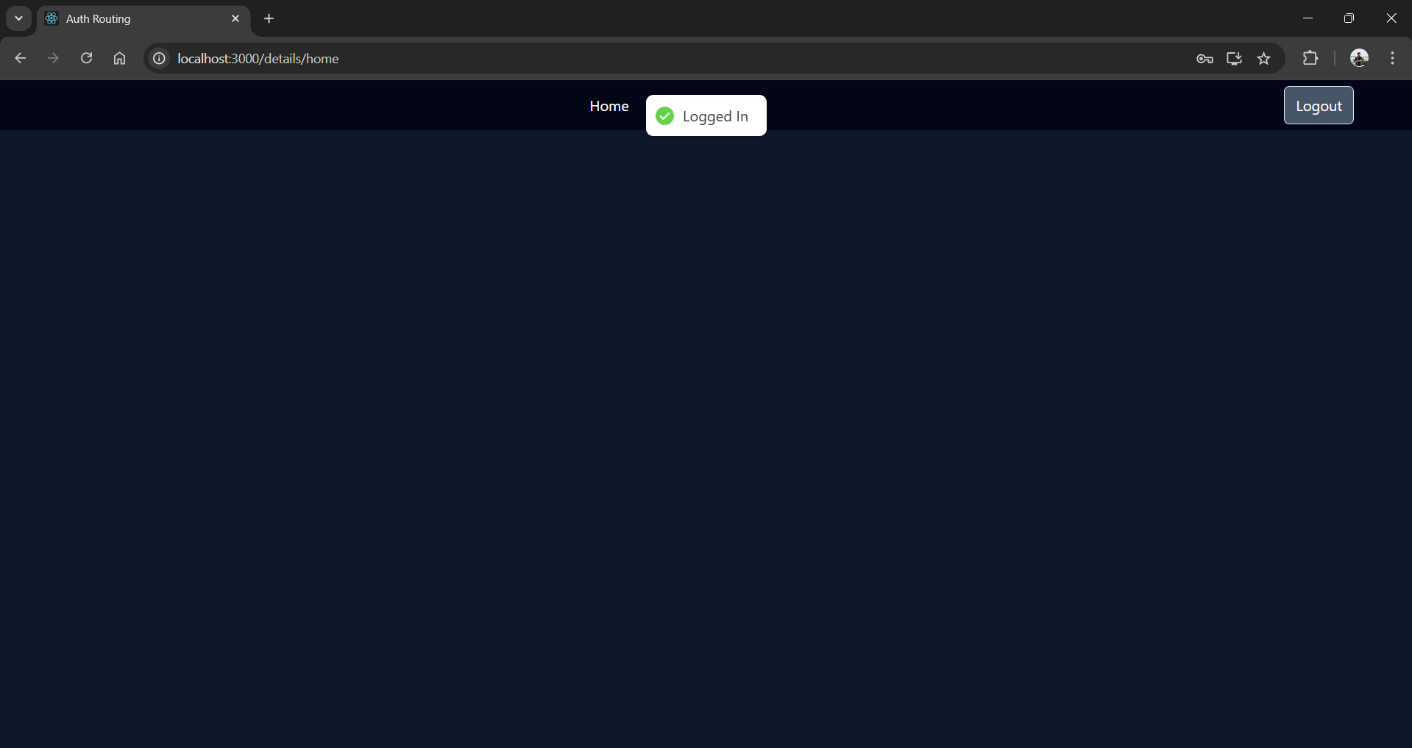
**Task Description :** Develop an authentication-based routing system to restrict access to specific routes. If the user is logged in, they can access the protected route. If not logged in, they are redirected to the Login page.

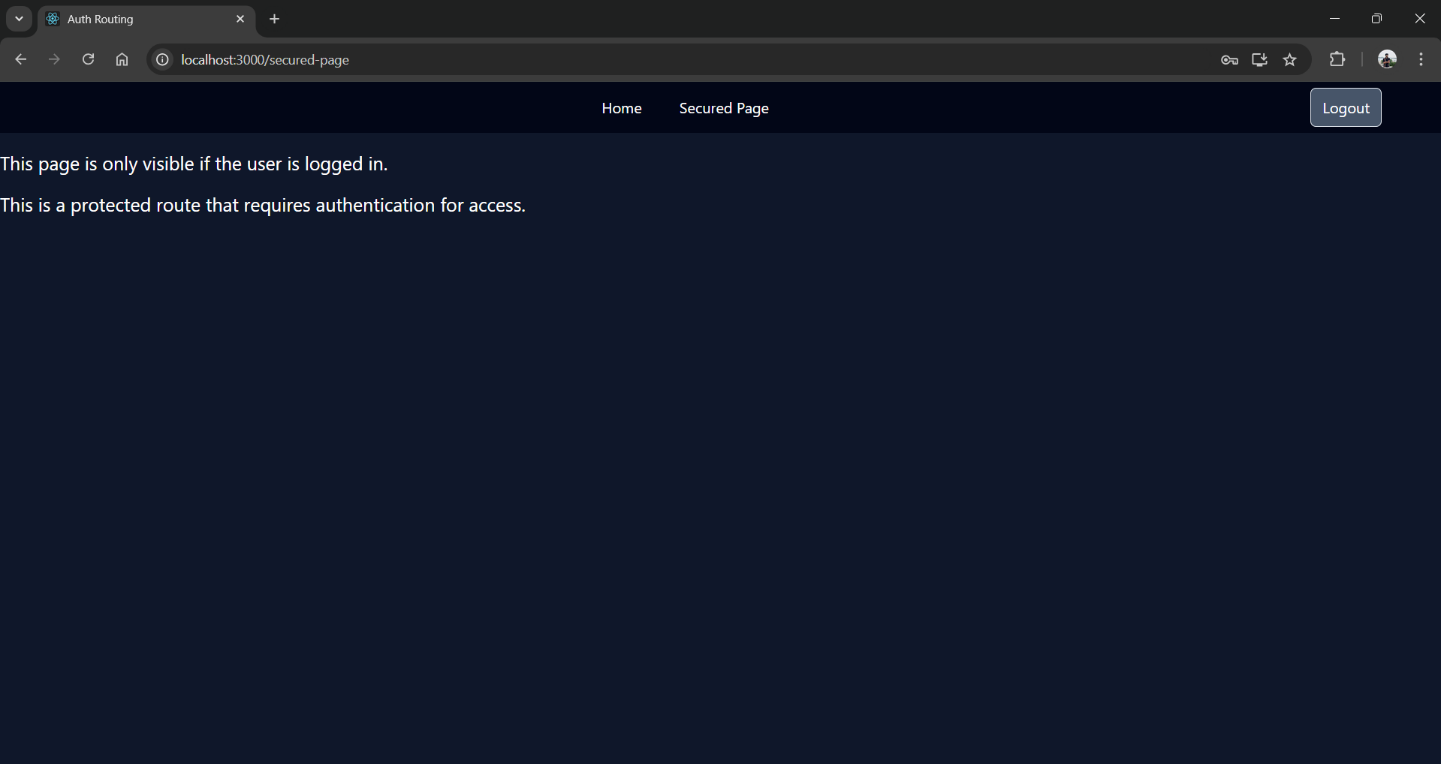
Technology :- React.js, Node.js, Express.js, MongoDB, Redux, Tailwind CSS

**Task Output Screenshot :**









**Widget/Algorithm used in task :**

1. React Router for Navigation :

* Used React Router to handle routing between different components/pages, including open and protected routes.

1. JWT for Authentication :

* Implemented JWT (JSON Web Tokens) for user authentication. After logging in, a token is stored and used to verify the user’s identity.

1. Private Routes :

* Created a PrivateRoute component that checks if a user is authenticated before allowing access to certain routes.

1. Redirects for Unauthorized Users :

* User without authentication are redirected to a login page using the Navigate from React Router.

1. bcrypt for Password Hashing :

* Used bcrypt to securely hash passwords before storing them in the background, ensuring passwords are never stored in plain text.

1. Toasters :

* Integrated react-hot-toast to display toast notifications for events like successful login, unauthorized access attempts or errors.

1. Backend Authentication Management :

* Authentication -based access is enforced server-side by checking the user information stored in the database.

1. State Management :

* Manages the authentication state and passed it down through components using React Redux.