

**Lesson 5-Building RESTful APIs with MERN Stack and Redux Toolkit**

**Activity 6 – Protecting Routes**

The objectives of this activitiy are to:

1. Understand the importance of route protection in web applications such as securing sensitive pages like user profiles or admin dashboards.
2. Use useSelector from Redux Toolkit to access user authentication states (such as **isLogin** flag) from the Redux store.
3. Utilize useEffect and useNavigate from react-router-dom to redirect users based on their authentication status.

**Instructions:**

1. The following routes and its corresponding components should be protected and should not be viewed by authorized users.
   1. Home, Profile
2. In Home component import the following:

import { useSelector } from "react-redux";

import { useNavigate } from "react-router-dom";

import { useEffect } from "react";

1. Retrieve the current value of the **isLogin** flag from the state in the Redux Store.

const Home = () => {

  const { isLogin} = useSelector(state => state.users)

…..

}

1. Create a useEffect function.

  useEffect(() => {

  }, []);

1. In the useEffect function, check if there is a value for the email from the state. If none, it means that there is no prior login and authentication. Redirect the user to the login route.

useEffect(() => {

    if (!isLogin) { navigate("/login") }

  },[isLogin])

1. Do the same logic in the Profile component.