**Assignment 10**

Top of Form

Bottom of Form

Up to this point, component communication has primarily been achieved through states and props. In this assignment, you will have the opportunity to manage state using Redux.

**Building Admin functionalities:**

1. Changes in POST: /user/create endpoint

In addition to providing full name, email, and password details, you will also need to include a new field called type. This field must have a value of either "employee" or "admin" and cannot be left empty or accept any other value besides these two options. Also, alter get all users API to fetch all the details for users except password. Additionally, modify the get all users API to retrieve all user details except for passwords.

1. Building admin page “Employees”

You are tasked with developing a straightforward admin page that will showcase all users in a table format along with their details such as email, name, and type.

1. Changes in Login API and Routing

The routing will be restricted based on the type of user. For example, when an admin logs in, they will only have access to the admin page. Conversely, if an employee logs in, they will only be able to view the pages developed in the previous assignment

Top of Form

Bottom of Form

**Functionalities for Admin Portal:**

1. Create a API: /create/jobTop of Form

Create an API with fields Company name, job title, description, and salary. While you have the freedom to include additional fields, it's recommended to maintain simplicity.Bottom of Form

1. Construct an "Add Jobs" page within the admin portal. On this page, you'll create a form that includes all the fields provided by the /create/job API. Upon saving the form, a new record should be added to the collections.

**Functionalities for Employee Portal:**

1. Create a API: get/jobs

This API should retrieve the list of available jobs from the collection that was posted by the admin.

1. Create a "Jobs" page within the employee portal. On this page, display all the available jobs retrieved from the database.

Note: The "Add Jobs" page should only be accessible to Admin users, while the "Jobs" page should only be visible to Employee users.

**Add-ons (optional):**

1. Include a spinner or loading indicator during API calls from the UI. This will provide users with a visual cue that they need to wait for a moment while the data is being processed.
2. You can add pagination for Jobs page.

**Instructions:**

* No pages should be accessible without any login.
* Make use of redux for all the state management.
* Make use of Material UI or React Bootstrap to build the components such as nav bar, card etc.

**Helpful Links:**

Routing:

1. <https://hygraph.com/blog/routing-in-react>
2. <https://www.youtube.com/watch?v=oUZjO00NkhY>
3. https://stackoverflow.com/questions/69851678/correct-way-to-redirect-user-to-different-routes-based-on-authentication-in-reac

Material UI and React Bootstrap:

1. <https://mui.com/material-ui/all-components/>
2. <https://react-bootstrap.netlify.app/>

Redux:

1. <https://react-redux.js.org/tutorials/quick-start>