**Product Engineering | UI - Hiring Exercise**

Authors: Dave Reidl

Hiring Manager : Ramakrishna Gunupuru

Date: January 2021

# **Exercise Goals**

* To learn more about how you think about UI in general
* To understand how you come up with a solution for a specific scenario
* To observe how you gather and present your ideas
* To review how you write basic feature specs

# **Exercise Instructions**

* This exercise is intended to take no more than 4 hours of preparation and does not assume that you are an expert in UI or in our product. Further, please do not spend any more than 4 hours on preparation.
* The review of this exercise will primarily be a discussion on the outcome of your results. You will need to prepare the applications and talking points as the deliverables. You can also present supporting wireframes and/or diagrams as necessary
* If you have any questions about this exercise while you are working on it, feel free to email the hiring manager
* You will present the results of your exercise in person to a select group within management during the interview process
* You will be required to zip the applications developed and send it across to Celigo post the timeline

# **Exercise Overview & Assignment**

## **Part 1**

Use React-Router-Dom module to create a sample SPA with the following routing:

**1. “/”**

Renders “hello world” and a simple appbar (link to ‘sign-in’ and ‘about’)

Example:

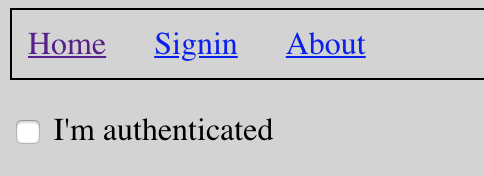


**Note** :-

* All other pages should have this same appbar, so this code should not be duplicated on every page.

**2**. **“/signin”**

Renders the same appbar, but also has a checkbox that toggles “authenticated” status. This should set a state property that can be used elsewhere (to simulate an auth flag)



**3**. **“/about”**

Renders a page with a left nar bar with the three links below (company, jobs, edit)

* /about/company
* /about/jobs
* /about/jobs/edit



**Note :-**

* When a user first comes to the “/about” page, the text in the right content area should read “About us”... If they click on the first 2 links (Company and Jobs) the content area should reflect content from those views.
* If the user clicks the “Edit Jobs” link, the code should be smart enough to check the authenticated flag and redirect them to the “/signin” page.

## **Part 2**

Convert the wireframe app from **part 1** into a “material-ui” branded app.

**Note :-**

* The whole app doesn't need to be converted, maybe the Appbar with the top-level nav links could be implemented as a MUI appbar, the sign-in route could render a dialog instead of a embedded form, etc