Okra Team
Startup 07 Submission

Special Server Setup Procedure

Our server does not require additional setup besides npm install and node src/server.js.

Main Feed- Nick Achin

The main feed now uses the ranking algorithm taking “Jane Doe’s” skills and interests and matching them on the job openings available. The main feed uses the “feed” collections for the feeds and jobItems/notificationItems to populate the feed.

BUGS: Unfortunately, our transition to the database did not go smoothly and the main feed is not properly populating the UI.

Job Board - Josh T.

The job board now uses the ranking algorithm for Jane Doe’s skills and interests, as well as pulling the job postings from all the available jobs in the mongodb database. Specifically, it pulls directly from the “jobItems” collection for all of the specific jobItems.

I worked with Ryan Minichiello to implement the ranking algorithm. While relatively simple, it should be effective for its purpose. The algorithm takes in the given user’s list of skills and interests, and finds the number of those skills and interests that appear in the given job posting’s tags. The percentage of job posting’s tags that appear in the user’s skills and interests is calculated, and the recommendation level decided based on the percentage of overlap. With the current three ranking levels, if the user shares ⅔ or more of the tags, they are given a gold recommendation. If the user shares only ⅓ or more of the tags, they are given a silver recommendation. If they share less than ⅓ of the tags, then they are given a bronze recommendation for that posting.

Bugs:

- There seems to be some problem with getting the user’s profile data from the database sometimes, which causes certain variables to be undefined and result in the feed being broken. This seems especially likely when refreshing the page multiple times. Waiting a few seconds before the refresh seems to fix this issue. It appears to be somewhat random, so I am having trouble finding what causes

it. Sometimes the method works, sometimes it doesn't. This is therefore most likely a problem with asynchornicity.

Project Page - Thomas Cloutier

The project page now pulls all of it's data from the online database. The specific collections that the page uses are the "project" collections for all of the specific project data, the "positions" collection for all of the open and filled position information for the project, and the "notificationItems" for all of the specific updates for the project.

Dropped Features:

-For this page the apply button does change upon interaction but fails to do anything based on the fact that we never implemented user sign-in and user creation figures. Ideally we would have like this to be used as a way for individual users to apply to projects with open positions that the user felt

- The following button was dropped for many of the same reasons. Also populating the user's feed with random project updates made the main-feed too cluttered

-Messages to projects were dropped as we felt that having users message projects was not a good feature. Also time constraints would not allow for it

Create Project - Ryan LeCours

The Create Project Page sends its entered information to the mongo database, however there is a small bug in this process. After following console.log checks to debug, the data makes it to JSon.stringify(body) in client/app/server, but body is somehow suddenly undefined, where it was defined up until this line, and a 501 error is thrown. We ran out of time before we could find out what caused this error.

Regardless, I added a remove tag button to the tag section, so a user can now add up to five tags and remove them if need be. The position section also used to let a user add too many positions, so I have gone back in and corrected that. All other functionality of my page is working, and all data entered is recorded when the Create button is clicked.

Profile Page - Ryan Minichiello

The profile page successfully pulls the user data from the mongo database. Specifically, the user collection for "Jane Doe" is pulled. It is also rendering the project pills just as the sidebar is.

Josh and I implemented the ranking algorithm, and I successfully ported it to the job board and the main feed. This was the central feature of the application, so it felt good to finally implement it and get it working the way we intended.

We as a team configured the database, and once the first domino of serving from mongo fell with profile and project, the rest of the components fell into place.

Dropped Features

WIth the dropping of the inbox feature, the user's messaging function will no longer work. User "Jane" is pretty hard coded throughout the application, but as a single user application requirement for demo, her data flows down to components not only used by her, but throughout the rest of the components in the application.

Side Navbar - Caroline Kim

- When the projects in side navbar are clicked, it will bring the user to the project page.
- The name of the project is pulled from the user data in the mongo database. From the user data, it looks for the IDs of projects that the user is associated with. And it goes to the project data to compare that ID and find the project name (identifier) to put it on the side navbar
- **BUGS**: clicking on any of the projects in the side navbar will always bring you to the same default project page

Inbox Page - Caroline Kim

- We ended up dropping the Inbox Page.
- But before it was dropped, it was sending chat data to mongo_express, and the banner on the right side wasn't static anymore.
- The main reason for dropping this page was because we couldn't implement the onPost or onClick to click around different chats or send any messages. Due to this problem, there was no purpose of having this inbox page. So we removed it from the top navbar and got rid of all the inbox components.

After getting rid of the inbox page, I worked on the side navbar, helped other people put up the data in mongodb and rendering from there.