Job Board

Fullstack Development

Overview

For the final project in this course, you will create a Fullstack app that mimics a simple job listing board. Users will be able to search for jobs. Furthermore, users will be able to log in to the site and then be able to post/edit/delete job listings, favorite posts and see their favorites.

For this assignment, you may work with one additional student, or you can also do this work by yourself.

Please note: you are welcome to come up with your own project idea, but you must contact the teaching staff beforehand to ensure that the code is sufficiently worthwhile and can reasonably meet the expectations of the rubric below. I am also open to you experimenting with other technologies in your stack, but again, please see us first.

Rubric

- Core Functionality - 30pts
- Working Github and Heroku App - 5pts
- Good Pages and Styling - 10pts
- RESTful APIs - 20pts
- MongoDB, Mongoose and Security Implementation - 15pts
- Well Written JavaScript - 10pts
- Writeup - 10pts
- Extra Credit - 16pts

Core Functionality

The goal of this project is to create an online job board posting similar to Indeed or similar sites. This section describes all the pages and features we will be looking for when grading.

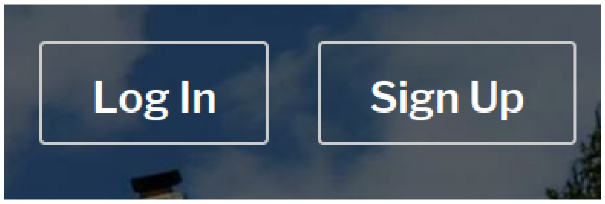
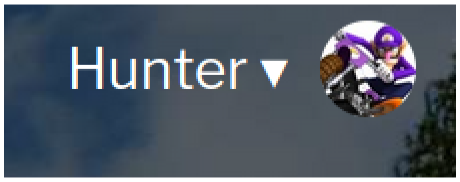
Homepage

When users first enter your page, they should be directed to your website's homepage. The homepage should just display the title of your website (name it whatever you please) in a fun font and with any images/icons you want. Under the name and (optional) logo, you should have a text form input field that acts as your search bar and a button that submits your search input. Clicking on this button will redirect users to the search page (discussed further below.)

The homepage should be visible to logged in and logged out users.

Navbar

There should be a nav bar at the top of every page. This navbar should contain a link to the home page, the favorites page and a link to a place where a user can login or register a new account. If the user is logged in, the navbar should contain the previous, but also a button to create a job listing, a button to log out and a slightly stylized reference to their username. The following example is from Redfin.com and shows how the login/register buttons switch to the username when the user is logged in.

	
Logged out	Logged in (clicking will take me to my settings page, which you do not have to deal with here)

Search Results Page

When a user clicks the “search” button on the homepage, the text in the input field will be used to search all job listings. For the sake of this assignment, assume that users will only use a single word in the search bar. When a user searches for a job, you should do a case insensitive search only in the job title and can match for incomplete words. So if there are 4 jobs in the system (“Accountant”, “Researcher”, “Software Engineer”, “Soft Drink Seller”) and the user searches for “soft”, then the “Software Engineer” and the “Soft Drink Seller” jobs should be shown.

Because users will want to see many jobs very quickly, you should not show the full job description at first, but rather only the job title, location and company name. Then a user clicks on the search result, this should redirect them to the Job Detail page. An example of job search results on Indeed is below.

Sort by: **relevance** - date

Page 1 of 806 jobs

new

Day Porter/Janitor

5 Star 5 Inc

Bellevue, WA

$20 - $25 an hour

Easily apply · Responsive employer · Urgently hiring

- Monday - Friday, 8 am to 5 pm. Having a reliable vehicle is a must.
- Day Porter position available in Bellevue, WA.
- Job Type: Full-time, Part Time available.

Active 5 days ago

new

Janitor

Steeler, Inc.

Seattle, WA 98178

$19 - $22 an hour

Easily apply

- Able to stand and move on foot for periods of time while working shift.
- Clean, sanitize, and stock restrooms and kitchenettes.
- Vacuum, sweep, and mop floors.

Active 5 days ago

Housekeeper Hiring Event All Shifts - Sign On 11. 17. 2021

Seattle Childrens Hospital **3.9** ★

Seattle, WA 98105 (Windermere area) +6 locations

Job Details Page

When a user clicks a job search result, they should be redirected to the Job Detail Page of the job they clicked. A job detail page will contain the following pieces on information:

- Job title
- Company name
- Location
- Description
- Employer email contact (formatted as a click anchor tag using the mailto feature)
- Optionally, a link to the company website (if one is not provided, do not show this field)
- Posting date (this should be calculated when inserted into the database)

Additionally on this page, there should be a Favorite button/icon. When a user clicks on this button, we should record in the backend that the user has "Favorited" this job description, and we will use this information later. If they click on this button again, we should then "Unfavorite" this job description. Because this can be confusing, make sure to indicate on the page somewhere (either on button itself or around it) if the user has Favorited this home or not.

If the user is logged out when they click the Favorite button, we should redirect them to the log in/registration page.

Login/Registration Page

When a logged out user sees the navbar, they should see the option to log in or register. These forms are very similar: when a user wants to sign in, they should be redirected to a login page that accepts a username and password. When the user wants to register, they should be redirected to a registration page that accepts a username, a password and a password verification form field.

You should show an error with a helpful message in the following cases:

- A user tries to sign in with a nonexistent user or attempts to log in with an invalid password
- A user tries to register an existing username
- A user tries to register but their password and password verification don't match

After a user successfully logs in or registers, the backend should set cookies for the user and the user should be redirected to the home page.

Creating/Editing/Deleting Job Detail Pages

After a user logs in, they should see a button to create a new Job Detail page in the header. When they click on this page, they should be brought to a form that allows users to submit the following values:

- Job title

- Company name
- Location
- Description
- Employer email contact
- Company website (optional)

All of these fields, except for company website, is required so users should get an error message if they are missing a value. When a new Job Listing is submitted, the backend or database will add the job posting date. After the page is submitted, the user should be redirected to the new Job Detail page.

Additionally, if a logged in user is looking at a page that *they* created, they should be able to see the option to edit or delete the detail page. If they click “edit”, they should be brought to a similar form as the Create Job Details page, with all the information they filled out from the existing job detail page. After editing the page, they should be redirected back to the Job Detail page. If the user clicks delete, the Job Detail page should not be accessible anymore on the website and the user should be redirected to the home page. If a user has not created this page, they should not have the option to edit or delete this listing.

Favorites Page

Finally, we should have the user Favorites page. The link to the user Favorites page should only be available in the navbar if the user is logged in. When they click on the user Favorites page, the website should show all the “Favorites” a user has. The results should shown in the same way as the search results (i.e., only show job title, location and company name.) If a user clicks on one of the favorites, they should be redirected to the jobs detail page.

Working Github and Server Link

For this assignment, I recommend you use Heroku to host your code, but you are welcome to use any web hosting service you are comfortable with. Please follow the instructions from the lectures or contact any of the teaching staff to get this set up if you need help. Please be sure to add the TA’s as collaborators to your Github repos.

Correct Pages and Good Styling

You are responsible for building out a website with all of the pages above pages (this is a good summary or checklist for the above):

- Home page
- Search result page (this should use query params in the URL)
- Job Detail Page
- Sign in/Registration page (this can be one or two pages)
- Favorites Page

- Create and edit page

Additionally, you are required to have a navbar with the specification listed above.

As always, you should have a unique and consistent style across the different pages. There are no specific styling requirements, but make sure that your website looks good on mobile as well as desktop. You are welcome to use any 3rd party styling libraries, such as Tailwind, React Bootstrap, Material UI, etc. Ensure that this is something you would be proud to show an employer.

Finally, if these views are on different pages, consider sensical and good URL design.

Well Written JavaScript

Now that we're writing logic, you must start considering the quality of the code you're writing. Functions should be simple, easy to read and avoid repetition. Make sure to make helper functions to simplify code in the backend and ensure that your React components are as simple as possible. We are not expecting you to use any "advanced" JavaScript functionality, but you should be writing code that you would be happy to show to a potential employer.
You are welcome to use or not use any library of your choice (this means that you are not expected to use Redux, for instance.)

RESTful APIs

Since this is a full stack app, you are expected to write backend Express APIs using *each* of the proper RESTful verbs that we learn in this course: POST, PUT, DELETE and GET. It is important that your code respects the promise made with these verbs so that there are no unexpected side effects.

MongoDB, Mongoose and Security Implementation

You should be correctly connecting to MongoDB and Mongoose with your app. You should have at least 2 collections (i.e., MongoDB tables).
Additionally, your data should be secure in the ways we show in class. Passwords should be encrypted and requests from invalid users should be rejected (for instance, if an API request comes to delete another user's comment, that should not succeed.)

Extra Credit

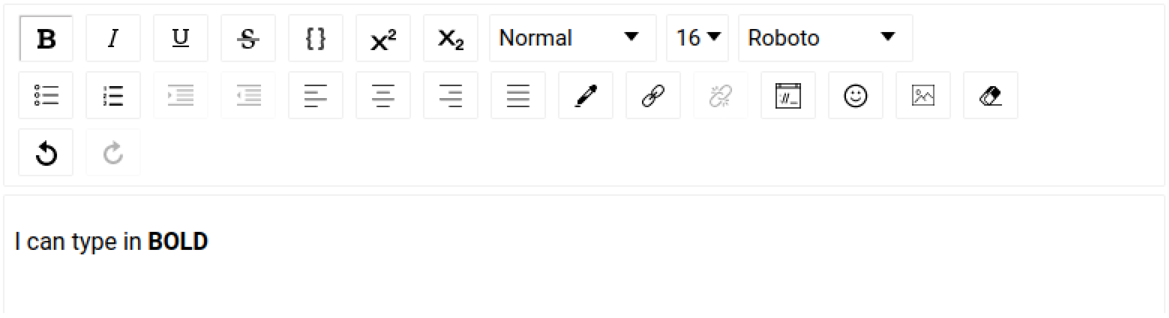
These tasks are OPTIONAL but will be good experiences if you're interested in exploring further into some harder ideas in web development.

Submit Early - 5pts

Submit this assignment by December 13th at Midnight to receive extra points.

Rich Text Editor - 3pts

If you go to a website like Piazza, Google Docs, or other, you will often see that in input fields there are options for more dynamic text input, including bold, italics, fonts, etc. (an example is below.) These are known as WYSIWYG editors or Rich Text editors (you can also consider using a Markdown editor). Implement this on the job description part of your form and properly render this text when a user looks at the job detail page.



I can type in **BOLD**

Correct Post-Login Redirect - 3pts

When a user logs in or registers, they are redirected to the home page. This is a *bad* customer experience as they may have been looking at a particular search page or a detail page. Instead figure out a way to redirect the user back to the page that they were just looking at.

Company Icon Photo Upload - 5pts

Sometimes companies like to display company icons or logos with their job postings to really grab attention. On the create/edit delete page for the job posting, add new fields to upload an image and then display that image when a user looks at the job detail page or sees the job on the job result page.

Job Submission Tracker - 5pts

For users that have many favorite pages, it can be hard to remember which ones they've applied to or not. On the job detail page, below the option to "Favorite" a page there should be a drop to indicate the user's application status: "Not started", "Applied", "Interview Scheduled", "Accepted", "Rejected". The value should ONLY be visible to the currently logged in user (i.e., one user may mark one application as "rejected" and another may mark it as "applied").

Then, on the user;s Favorites page, the job applications should be grouped and sorted in the order described above. You can choose to style this however you please.

Writeup

With your submission, you must include a writeup that touches on the following points. You may discuss any other ideas that you deem salient to this work:

- What were some challenges you faced while making this app?
- Given more time, what additional features, functional or design changes would you make
- What assumptions did you make while working on this assignment?
- How long did this assignment take to complete?

Deliverables

Include all the following in your submission on Canvas:

1. A link to your Github repo. If you are working with a partner, you may submit a link to the same repo (for grading purposes, the TA's will likely only look at a single repo so make sure they are identical.) Please note that your Github repo should be named: {firstname}-{lastname}-project3, and if you're working with a second person both names should appear on the repo.
2. A link to your Heroku app. As above, if you are working with a collaborator, please submit the same link.
3. Your writeup. If you are working with a partner, you may each write this together or individually, but please indicate this with your submission.
4. The name of your collaborator(s), if any

Academic Integrity

As always, all assignments are expected to uphold and follow NEU policies regarding academic integrity. Any students that violate these policies will be reported to OSCCR immediately.