**Tech Career Development Resources for Students**

INST377 Fall 2018

Irene Su, Justin Kim, Grace Zhang, Rusha Shrestha, Ebrima Jarjue

<https://github.com/jskim97/INST377-Website>

<http://35.237.227.222/Pages/Home.html>

**Information problem**

There are a plethora of career-related resources out there to help prepare students throughout the internship/job hunting process for technical opportunities, whether it be through search engines like Google or in-person services. However, some undergraduate students (especially underclassmen) are oftentimes not aware of these existing resources. Other times, the resources are out there, but may not fit the information need of a typical undergraduate student looking to prepare for the job hunting process for technical opportunities in the tech industry. For example, in-person career resources at the student’s university may help with general resume and behavioral interview preparation, but may not prepare the undergraduate student to tailor their resume towards technical opportunities or prepare them for technical interviews.

With these considerations in mind, our web application aggregates all these curated preparation resources altogether to allow undergraduate students (with an interest in pursuing technical opportunities in the tech industry) to prepare as necessary for the entire job hunting process. At least for this website, we plan to curate resources primarily towards internships and entry-level opportunities.

**Strategies/Solutions**

The solution for addressing the information need/problem of undergraduate students looking for resources to prepare for the job hunting process (for technical opportunities in the tech industry) is aggregating curated resources onto a web application. There’s a ton of different ways to go about with designing the application itself to make navigation throughout the website accessible and user-friendly, but that also requires more time and resources to conduct those usability surveys and interviews!

We personally decided to break the website down into subsections, primarily "careers and internships," "interview resources," and "resume and cover letter resources” to make it broad enough to apply to the entire job hunting process, yet at the same time keep it simple. The originally designed solution was to break the website down based on opportunity types (internships or entry-level positions or UMD-specific resources) however that proved to be a bit more difficult and may lead to information overload if we were to highlight the entire process of job hunting from preparing a resume to the interview process.

**Rationales/Justifications on system design and technology**

Originally when we were determining which tech stack to use, we were a bit unsure of which frameworks and libraries to choose, primarily since we were still ironing out the scope of our project as well as how we would design the website. With this in mind, at that time, one of the first libraries that came to mind was Bootstrap, which is an HTML/CSS/JavaScript open-sourced library that offers various pre-made frameworks and widgets. Additionally, we used JQuery to make it easier to use JavaScript for our website. Since everyone on the team does not have too much experience in web development until this semester, we decide to keep it simple by utilizing current templates that Bootstrap offers. This would make it easier to launch the application from scratch without making it look like basic HTML while utilizing the features that Bootstrap has to offer. However, instead of utilizing a fully developed Bootstrap template, the website was built from scratch while using the widgets and existing code (i.e. CSS styling) that Bootstrap has to offer. Alongside Bootstrap, additional resources were utilized as part of the learning and development process like W3schools.

For the server-side scripting, we stuck with Node.js since in our initial research process we decided it would be easy to implement and was compatible with JavaScript. However, we had many problems with connecting our database with Node.js server. We kept encountering connection problems and while we tried to figure that out, we later switched to php server to make the connection successful. For our database, we used MySQL db and MySQL Workbench to edit our database. In this, we have 5 main tables named company listings, university, students, links, and link resources. These tables consists of resources that would be extracted from the server that resides in it to our website. This would update our website and provide our users the most relevant and updated information for their job searching process. Finally, for all these elements to be accessible, we used Google Cloud and created a virtual machine running Ubuntu. We used Google Cloud because it provides a platform for use to use its assets such as its virtual machine and other services contained in Google cloud data center.

**How the final system solves the problem**

Our final system solves the information problem by removing the middle step of having to search for all the career resources in the technology field. Our system aggregates many tech related career websites that may have job/internship positions open, information on how to practice for technical interviews, resume and cover letter resources, etc. By culminating as many resources as we could on our website, we reduce the chance that students miss out on certain resources due to ignorance, as we have listed it all for them. In addition to creating searchable databases of resources, we also curated career tips for the students to prepare them for technical interviews or job positions. These tips are aimed to give students new information that they may not have heard at their university’s career center, as they are tailored to technical job positions and interviews. Overall, our system is able to improve the user’s chances of finding and being prepared for technical career opportunities, solving the original information problem.

**Challenges faced/impact on final design**

As mentioned in the "Strategies/Solutions" section, there's a ton of different ways to go about with designing this application. And a ton of potential issues with design, primarily since we would be reflecting our own opinions and biases as we designed the front-end. To some, a specific design may be better, but to others, that specific design may not be as great. With this in mind, we hoped to keep the design as simple and broad as possible without throwing extra sparkles and fluff (and information) onto the website. The original design was to break down the website based on opportunity types (internship/entry-level/UMD-specific resources) but that proved to be a bit more difficult and may cause information overload if we aggregated all those resources within three pages only.

Instead, we decided to keep it simple and make the sections broader (for the general resume and cover letter and interview preparation). However, the difference in internship and entry-level opportunities are still distinct as highlighted in the “job type” section of the database. Another issue we ran into was just structuring the information in an efficient manner. Originally, the "Interview Resources" and "Resume and Cover Letter Resources" section had both sections about various types of tips as well as the external resource databases. However, that proved to be a bit too much on one page since the website is not set up in a way to allow users to scroll down and consume the information efficiently. Instead, these sections were broken down into respective tips and database pages. However, looking back at this process, the final front-end (and system design) can definitely be improved with feedback from undergraduate testers as well as mentorship from software engineers (and our professors)!

Also for the final design of the system design, originally we planned to utilize Bootstrap, Node.js, JQuery, and MySQL. As of now, there is a working front-end but we struggled connecting the database to the front-end on the server; due to this, we had to change plans from utilizing Node.js as our server-side scripting to utilizing PHP as well as potential PHP libraries. Although utilizing different technologies should not have an impact on our final web design, at the same time it could potentially impact the capabilities and functions of the website. General challenges for system design mainly was choosing what technologies to utilize (tech stack) since often times some frameworks or libraries may not compliment one another as much.

**Future work**

Moving forward as we complete a working demo of the project, potential additions of future works include refining the database so that it will update the system automatically based resources so the database administrator does not have to do it manually every time the database needs updates. We can also add better features like better filtering features to ease the searching experience. As of now, for any of the databases, a user can type in any word that will match up to any of the values in the shown database. However, we could add more filtering features to the database tables. For example, we could limit each page to about fifteen to twenty search results and then put the next fifteen to twenty results on the next page. We could also create filter arrows to sort values in general. For example, filtering “Date Posted” from most recent to least recent.

Along with improving the general system design and implementing better features, we could also reiterate the design based on user feedback. The current design that we have is based on the feedback shared by team members primarily, but that also reflects our own opinions and biases and thus is not reflective of the general undergraduate population. With this in mind, we could do usability testing and ask undergraduate students around the University of Maryland as well as other universities to test whether the current design is efficient, accessible, and usable. Based on that feedback, we could iteratively update the design and implement new features and formats based on user feedback.