Jobby: Job Search Made Easy

Ashok Kumar Selvam
aselvam@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Rithik Jain
rjain25@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Sri Athithya Kruth Babu
sbabu@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Subramanian Venkataraman
svenka25@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Zunaid Hasan Sorathiya
zhsorath@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

ABSTRACT

For the next and final iteration of projects, we have chosen to extend and work further on the SRIJAS Project. SRIJAS is a resume builder and job search assistant which can help job seekers find positions of their choice on popular sites with ease and convenience. The previous team had the following features: LinkedIn job finder, Glassdoor job finder, deployment on the cloud. We have added the following features to the project: Indeed Job finder, GoingGlobal job finder, SimplyHired job finder, Monster.com job finder, Right to Be Forgotten, Resume Update, UI revamp, test case improvement. We believe these features make the project even more useful to its targeted users and we strongly believe that further development will be simpler and easier with the improved repository.

KEYWORDS

Job Finder, Job Search, LinkedIn, Good Repository, Readability and Reusability

1 INTRODUCTION

Job searching, in today's world of online job boards is a multi-faceted task, owing to the number of locations for job postings on the internet. There are multiple websites such as LinkedIn, SimplyHired, Indeed, Monster.com and GoingGlobal, to name a few. However, there is a dearth of quality aggregators in this area, and job seekers have to peruse all of these sites on daily basis to be aware of new job postings. Or they must sign up for job alerts on each site.

The number of tasks comprising this endeavour makes it cumbersome and tedious, not to mention difficult to manage effectively. SRIJAS was a step to alleviate this problem. Seeing its potential, we chose to extend and work on it further for the current iteration of projects.We noticed that the previous teams had incorporated several key features to the project, and we opted to continue taking the project further as we saw a great potential for improvement in this project.

We identified multiple facets of the project which would be served well with improvements: the previous teams had implemented up to two job boards, LinkedIn and Glassdoor, and we felt that target audience for the project would greatly benefit from the addition of several other job board websites. We identified several targetable websites: GoingGlobal, Indeed, Monster and SimplyHired from which we scrape data in addition to a revamp of the existing LinkedIn scraper to improve its working and clarity.

We also noticed that location-based filtering could be improved and used on all future chosen job board sites, which would be a valuable addition to the project's capabilities. This would allow users to filter jobs by location on all of the sites they try to use to find jobs through Jobby.

We also noticed a scope for improvement in the Database schema structure of the project. Updating resumes was another key requirement we identified.We also decided to add a delete data feature so the Right To Be Forgotten would be implemented. Both of these changes were added to the project as well.

We also decided to add an option for users to update their resumes. This would allow them to update their resume which would improve the accuracy of their future job searches, which we note to be a feature of vital importance for the perceived user base. We also redesigned the User Interface of the web application to improve User Experience.

We also identified that the administrators needed to run the job scrapers manually for the emails to be sent out to users, and deciding to remove the scope for human error, we have implemented a scheduler to do this automatically from the system so that job searches are run at the same time on a daily basis and users can get daily alerts.

Another one of our goals was increase the number of tests in the codebase to improve code coverage and ensure reliability of the system. We achieved this through unit testing.

2 MULTIPLE JOB BOARDS ADDED

In the previous version of SRIJAS, the job search and scraping had been implemented from two main sources: LinkedIn and Glassdoor. Both are common sources for jobs in the user space and would contribute an adequate number of roles for a user's perusal. However, we noticed that there was potential for expansion in the job boards used here.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Based on our research, we identified four other job boards for which job search and scraping have been implemented into the current iteration of the project. These are SimplyHired, Indeed, Monster and GoingGlobal. All of them are well-known sites with plenty of opportunities for users to look into. We believe they will be a valuable addition to the project in terms of the sheer variety of job positions they will contribute to the database.

We also proceed to reformat and improve the working of the LinkedIn scraper to improve the code quality, fixing bugs and improving the overall working of it. Since it is important to allow users to search for jobs only on the websites they would prefer to use, we also have implemented a feature to allow users to choose a selection of job board sites from where the scraper will draw and provide positions for perusal.

We believe that this will make for a much better user experience since users might have preferences on the sources of the postings they are looking for, and this is an important step to maintain their agency in the job search process. Additionally, we have also added a feature that displays the percentage of skills matching with the job posting description. This will allow the user to understand how closely the requirements of the job posting pertain to the skills in their own resume.

3 RESUME UPDATE FEATURE

Users often make changes to their resumes, adding on new skills and experiences, removing irrelevant parts, and editing the content and formatting to impress prospective employers. This understanding was our motivation to design and implement a feature to allow users to update their resume.

Using this feature, users are able to update their resume in Jobby's database. Each time it is updated, it is parsed and skills are extracted and updated in the database for the user in question. This will in turn contribute to job search results further down the line. An additional point to note here is that this allows for application to understand and work with user's ever-changing skillsets.

4 RIGHT TO BE FORGOTTEN

In the modern world, the right to have your data removed has become of paramount importance. Users should have the ability to remove their data from applications completely when they would like to do so. This is known as the Right to be Forgotten.

In line with this ethic, we pursued an implementation of a delete data or remove user functionality in our application. A user who longer wishes to be known by Jobby can at their whim, at any time, through the click of a button completely erase all their data from Jobby's database.

Future sign-ups are of course, welcome and encouraged, but the principle here is that Jobby gives users the right to be Forgotten by Jobby.We believe this is key to the entire user experience of the application and makes up a vital feature in the current iteration.

5 SCHEDULER FOR JOB SEARCH

In the previous version of Jobby, the flow of the application for daily users involved the intervention of human administrators to run the job searching scripts to scrape fresh jobs from the various job board sites in the application and then these would be sent to the users.

This introduced an element of human intervention requirement which we felt could be automated to make the system less error-prone, more reliable and stable. To this end, we have implemented a scheduler for the job scraping scripts to be run with. The scheduler will run the job scrapers at a fixed time on a daily basis, for the sites selected by the users, and proceed to generate email alerts for the registered users on Jobby.

We believe this feature improves the overall stability and reliability of the system and also adds an additional element which can be customized in future iterations.

6 TESTING AND CODE COVERAGE

We have ensured that all the new features we have implemented in Jobby have sufficient unit tests written. This is both to maintain the stability and reliability of the system, as well to ensure that in the process of adding our new functionality, we do now interfere with the existing features.

To this end, we have added test cases for all of the scrapers in Jobby and ensured that we meet a standard of code coverage. We believe this strongly adds to the overall robustness of the project, and makes it very easy for further extension and further work to happen on the same codebase.

7 UI/UX IMPROVEMENT

We have performed a redesign of the logo to fit the new name(Jobby) and also changed the overall structure of the web application. The web application previously comprised a single page. It now contains separate pages for login, new user registration, and user signup for new job boards, and a listing of currently registered job boards as well. The UX has been enhanced through these steps to create a more comprehensive and cohesive experience. We have also added improvements to the email formats being sent to end users.

8 REDESIGN

There has been a redesigning of the database structure underlying the application. We have added new tables to store the job boards each user has registered for. This is to ensure consistent user experience across devices. Constraints on the database have also been addded - a cascade deletion constraint has been added, and we have also added foreign key constraints.

In terms of the project file structure, we have changed the directory and subdirectory naming convention to use all lowercase to avoid issues when moving between systems. All test cases have been written or moved to the appropriate subdirectory.