

# Sprint 3 Deliverable

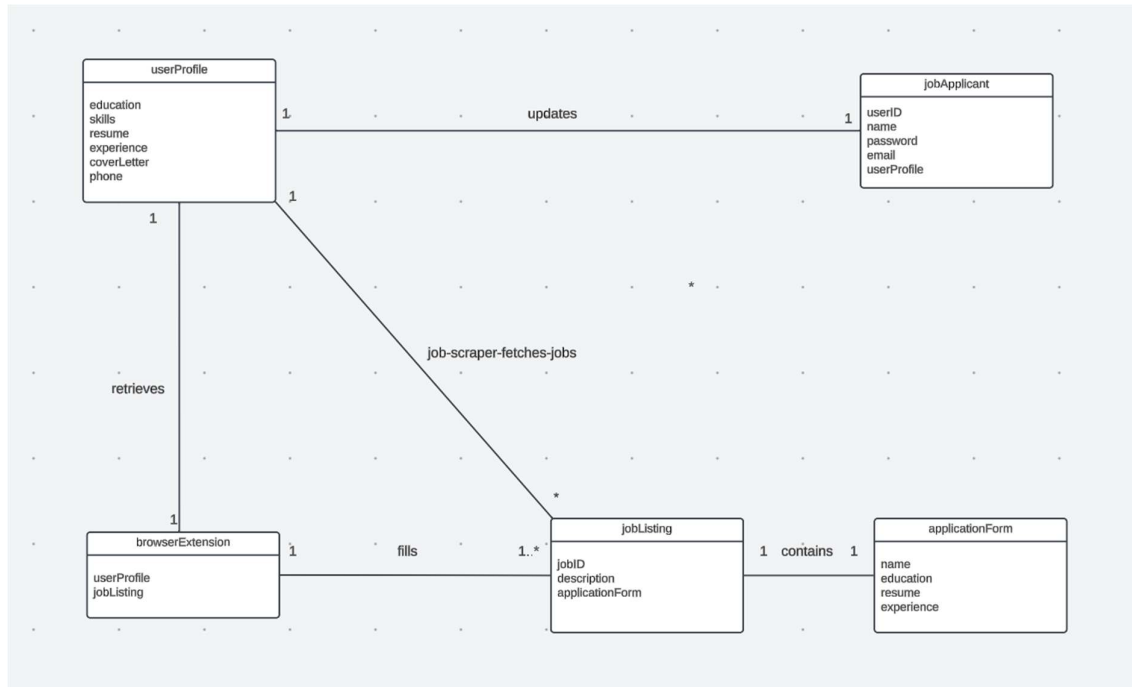
## Quantify

### Project members:

Khavin Krishnan Kalpana  
Vineela Yerrabelli  
Aditya Reddy Sabbella  
Uma Sruthy Gajula  
Krishna Vamsi Dhulipalla

## 1. Domain Model

Our Domain model remains the same.

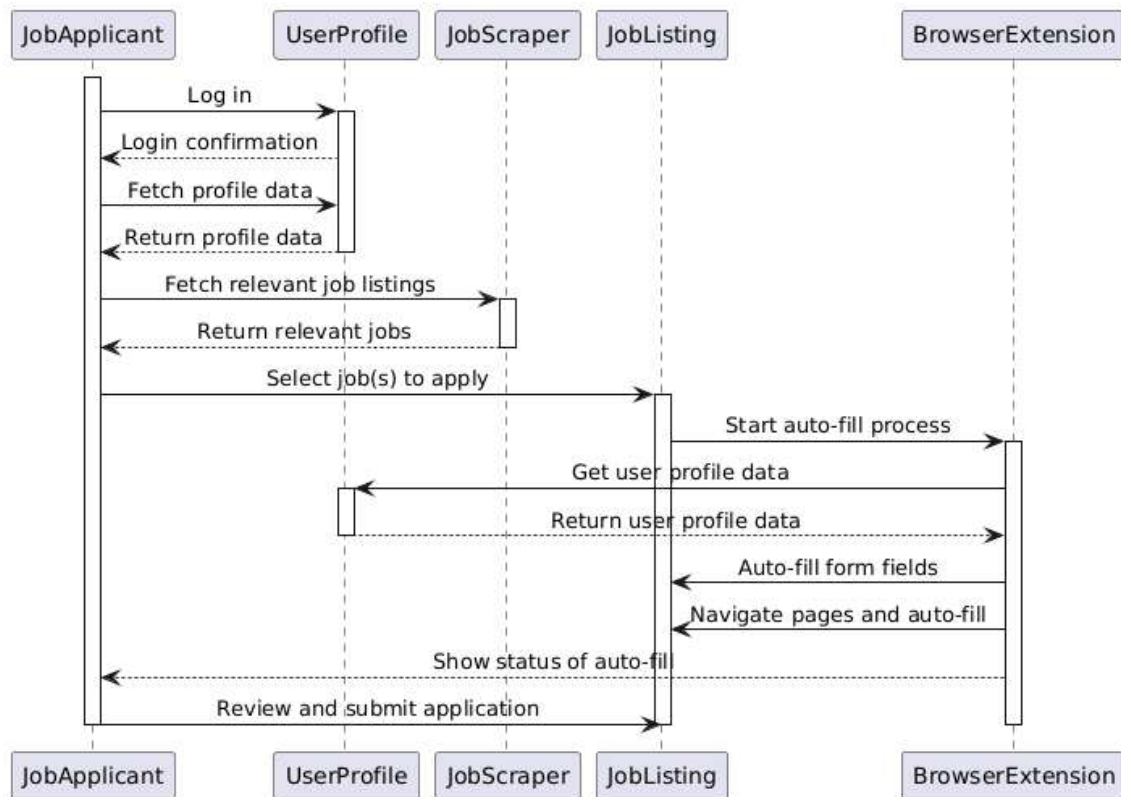


## 2. Interaction Diagrams

### User Stories Covered:

1. As a bot, I want to automatically fill out job application forms
2. As a bot, I want to fetch user information to auto-fill job applications

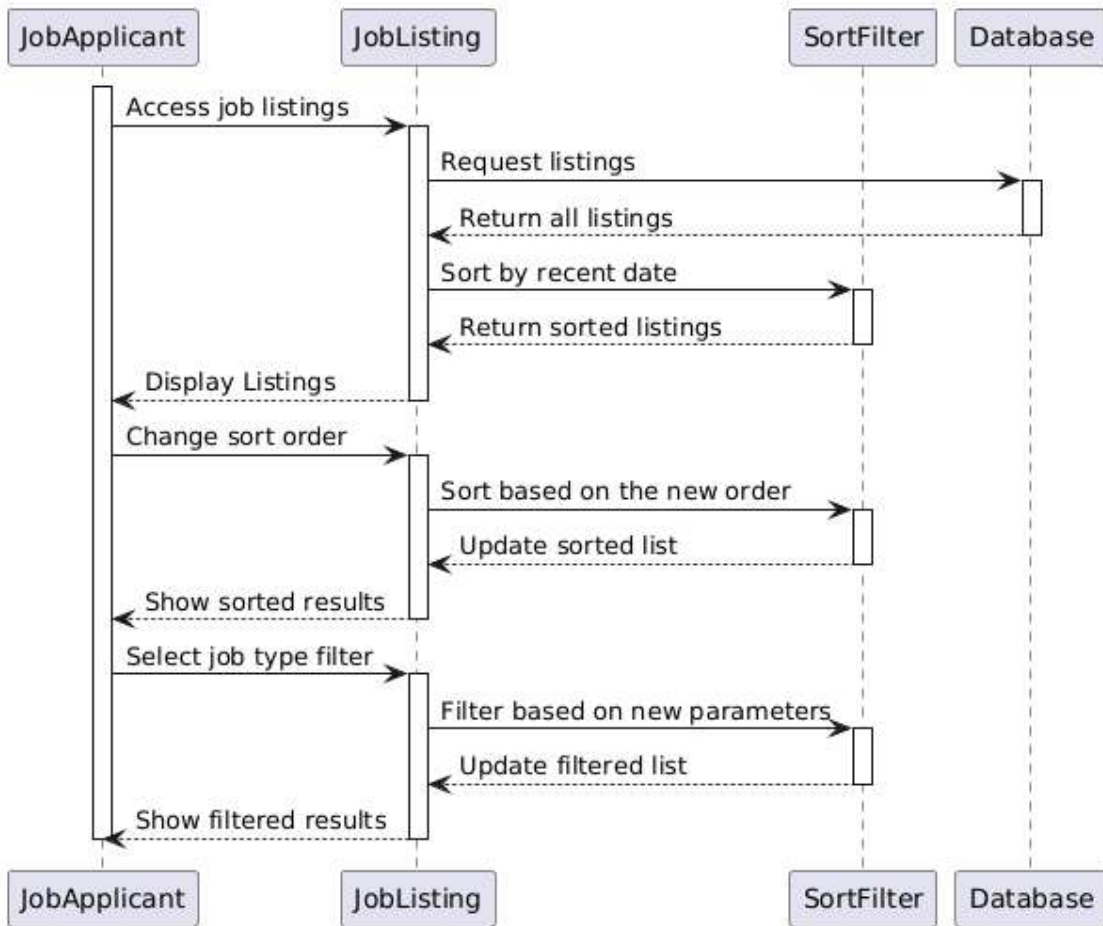
The following sequence diagram covers the above-mentioned user stories. The diagram also shows the time using rectangular blocks to denote that the objects were active during that time period.



### User Stories covered:

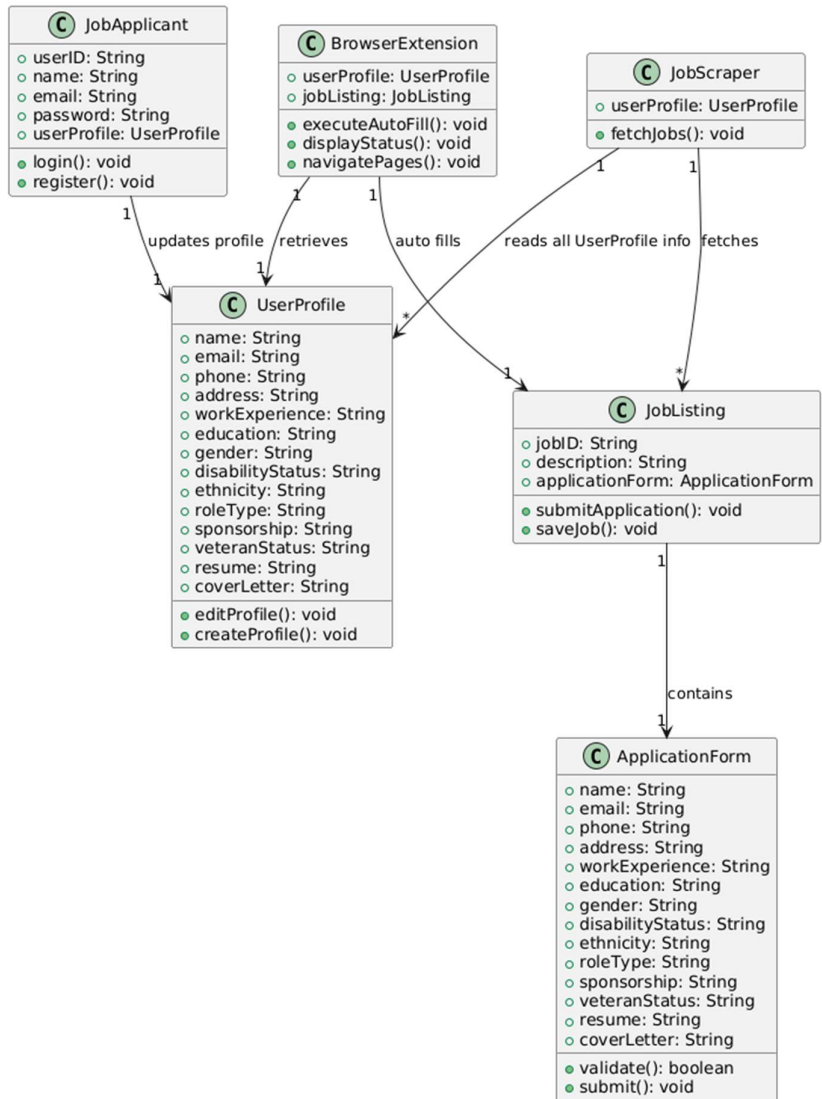
1. As a job applicant, I want to sort job listings by the date they were posted so that I can see the most recent opportunities.
2. As a job applicant, I want to filter jobs by type (e.g., full-time, part-time, contract) so that I can find jobs that fit my preference.

The following sequence diagram covers the above-mentioned user stories. Here we have added two new domain objects: “SortFilter” and “Database”. The main function of “SortFilter” entity is to sort and filter the job listings based on the sort or filter requirements. The main function of the “Database” entity is to store the job listings.



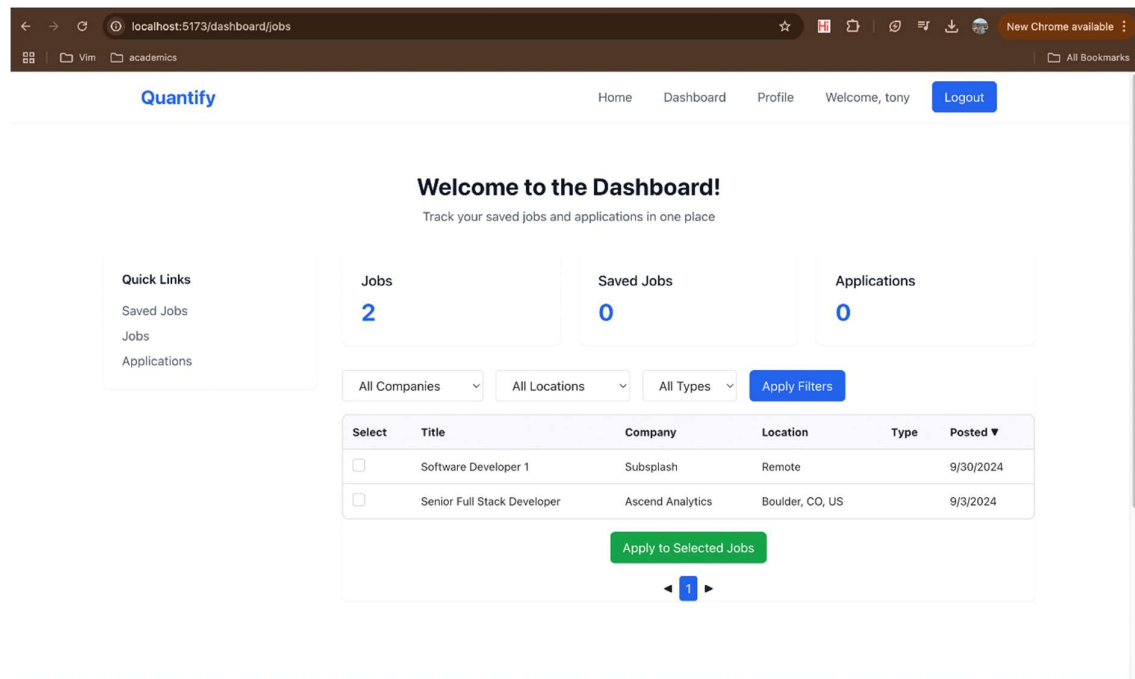
### 3. Design Class Diagram

Our Design Class Diagram stays the same. However, we have added two new fields “address” and “veteranStatus” to classes “UserProfile” and “ApplicationForm”.



## 4. Demo working product system

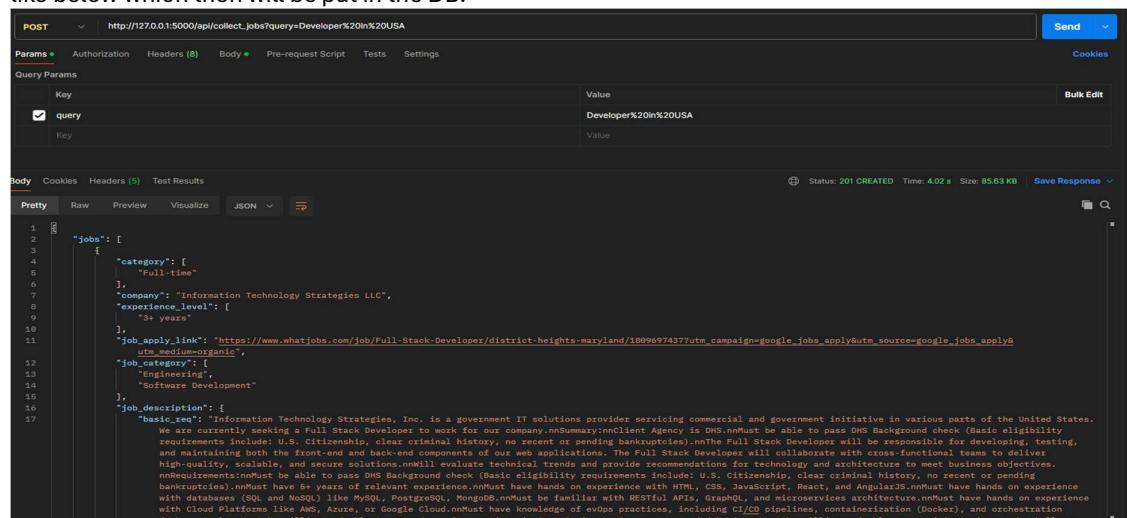
Updated Dashboard showing filter and sort options:



In the job listings page, we have added the ability to filter by Company name, Location and Job type. The user can also sort the job listings by clicking on the “Posted” table header. The “sort” option is indicated using an arrow near the header.

## Job Scraping:

We designed the scraper as an integral component of our project to manage job data efficiently. It enables us to fetch job postings from an external API based on specific queries, transform the data into a structured format, and store it in our MongoDB database. The scraper collects key details such as job titles, companies, locations, salary ranges, and application links. It also allows us to search the stored data using queries, making it easy to retrieve relevant job listings. This scraper automates the job collection process and plays a crucial role in streamlining our job management system. It gives us the response something like below which then will be put in the DB.



## Cloud hosting:

We've hosted our project on Azure, setting up resources to manage both our website and its backend. Our website runs on an Azure App Service called quantify, while the backend API, which our Chrome extension connects to, is hosted on another App Service called quantify-api. Both of these services are powered by an App Service Plan (ASP-Quantify-9ce5), which provides the necessary compute resources.

For deployment, we're using an Azure Container Registry (quantify) to store Docker images of our application, which are then deployed to the App Services. Most of our resources are located in the West US region, though the container registry is in East US 2, likely for performance or cost reasons. This setup allows our website, Chrome extension, and backend API to work seamlessly together.

<input type="checkbox"/> Name ↑↓	Type ↑↓	Location ↑↓
<input type="checkbox"/> ASP-Quantify-9ce5	App Service plan	West US
<input type="checkbox"/> quantify	Container registry	East US 2
<input type="checkbox"/> quantify	App Service	West US
<input type="checkbox"/> quantify-api	App Service	West US
<input type="checkbox"/> quantify-app_key	SSH key	West US

quantify-gjfscafuf4d4cfag-westus-01.azurewebsites.net

Quantify Home Login Sign Up

## Land the Right Job Simple, Strategic

From discovering tailored job opportunities to crafting standout applications and cover letters, we make every step towards your next job simpler and more effective.

Get Started → Learn More

## Chrome Extension:

boards.greenhouse.io/subsplash/jobs/7653784002

#LI-Remote #BI-Remote

### Apply for this Job

Apply With LinkedIn

First Name \* Tony

Last Name \* Stark

Email \* tony@gmail.com

Phone \*

Location (City) \* New York [Locate](#)

Resume/CV \* resume.pdf (File types: pdf, doc, docx, txt, rtf)

Cover Letter [Attach, Dropbox, Google Drive, or enter manually](#) (File types: pdf, doc, docx, txt, rtf)

LinkedIn Profile  
https://www.linkedin.com/in/khavin-kk/

How did you hear about this job?

Previously the Chrome extension “Quantify” was only capable of auto filling text input and dropdown fields. Now the extension is capable of attaching resumes to the job listing as well. The extension fetches both the user profile information and the resume from the backend built using Python Flask. We have also made the extension execute automatically without any need for the user to perform manual click to start the auto filling process.

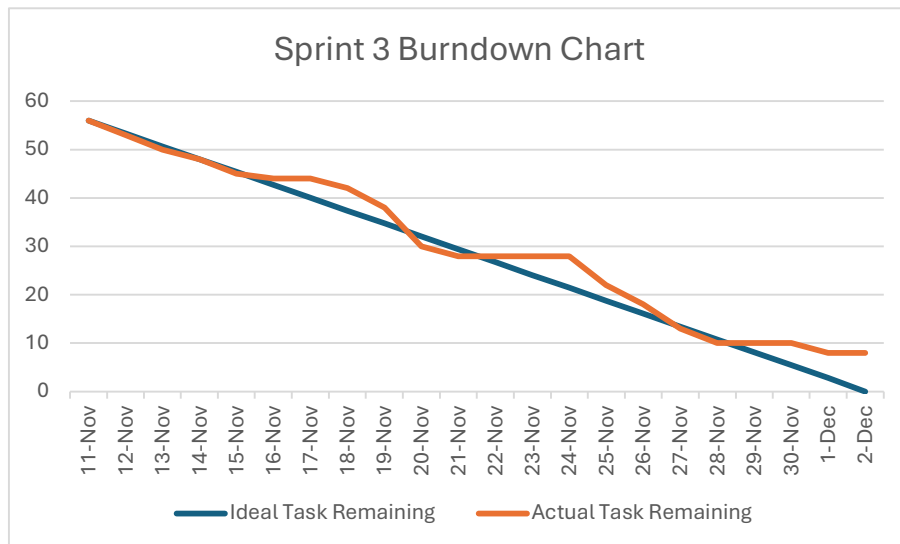
Note: The extension will only work for domains that start with the url pattern: “<https://boards.greenhouse.io/>”

## 4. Scrum

### Sprint 3 Backlog:

	JAB-7	As a job applicant, I want the bot to help generate a custo...	IN PROGRESS		Krishna vamsi Dhuli...	
	JAB-6	As a job applicant, I want to filter jobs by type (e.g., full-ti...	DONE	JAB Sprint 3	Aditya Reddy Sabb...	
	JAB-5	As a job applicant, I want to sort job listings by the date th...	DONE	JAB Sprint 3	Uma Sruthy Gajula	
	JAB-1	As a bot, I want to fetch user information to auto-fill job ap...	DONE	JAB Sprint 3	Vineela	
		JAB-4	As a bot, I want to automatically fill out job application for...	DONE	JAB Sprint 3	khavin
	JAB-36	Retrieve user profile info and resume from content script	DONE	JAB Sprint 3	khavin	
	JAB-35	Store the user profile info and resume in indexed db	DONE	JAB Sprint 3	khavin	
	JAB-34	Load user resume from service worker	DONE	JAB Sprint 3	khavin	
	JAB-33	Load user profile information from service worker	DONE	JAB Sprint 3	khavin	

## Sprint 3 Burndown Chart:



## Sprint retrospective:

We were able to complete 4 out of 5 user stories mentioned in Sprint 3.

### *Completed User Stories:*

**JAB-1:** As a bot, I want to fetch user information to auto-fill job applications

**JAB-4:** As a bot, I want to automatically fill out job application forms

**JAB-5:** As a job applicant, I want to sort job listings by the date they were posted so that I can see the most recent opportunities.

**JAB-6:** As a job applicant, I want to filter jobs by type (e.g., full-time, part-time, contract) so that I can find jobs that fit my preference.

### *Incomplete tasks:*

**JAB-7:** As a job applicant, I want the bot to help generate a custom cover letter for each job so that I can apply more effectively.

### *Challenges faced:*

We realized that the generalizability of the browser extension to different job portals is going to be a problem. We are currently able to fill the input text fields in most of the job portals. But the DOM structure of the dropdown fields is different for different job portals. For the scope of the project, we are planning to make the browser extension fully compatible with boards.greenhouse.io job portals.

### *Things that went well:*















































Like Sprint 2, we met twice a week using Zoom to discuss the progress and to see if there any blockers. We created the necessary sub tasks at the start of the sprint and each task was assigned a resource.



### Things that could be improved:

We were not able to integrate the user story JAB-7 with the extension. JAB-7 user story focuses on creating a custom cover letter for the user based on their resume. As of now, we are currently able to create a custom cover letter by leveraging LLMs. But we did not have enough time to integrate this part with the rest of the dashboard. Most of our resources were focused on finishing the extension and integrating the extension with frontend and backend.

### Product Backlog (Updated):

	JAB-7	As a job applicant, I want the bot to help generate a custo...	IN PROGRESS		 Krishna vamsi Dhuli...	
	JAB-6	As a job applicant, I want to filter jobs by type (e.g., full-ti...	DONE	 JAB Sprint 3	 Aditya Reddy Sabb...	
	JAB-5	As a job applicant, I want to sort job listings by the date th...	DONE	 JAB Sprint 3	 Uma Sruthy Gajula	
	JAB-1	As a bot, I want to fetch user information to auto-fill job ap...	DONE	 JAB Sprint 3	 Vineela	
		JAB-4	As a bot, I want to automatically fill out job application for...	DONE	 JAB Sprint 3	 khavin
	JAB-36	Retrieve user profile info and resume from content script	DONE	 JAB Sprint 3	 khavin	
	JAB-35	Store the user profile info and resume in indexed db	DONE	 JAB Sprint 3	 khavin	
	JAB-34	Load user resume from service worker	DONE	 JAB Sprint 3	 khavin	
	JAB-33	Load user profile information from service worker	DONE	 JAB Sprint 3	 khavin	
	JAB-3	As a bot, I want to handle job application errors gracefully	DONE	 JAB Sprint 2	 khavin	
		JAB-8	As a job applicant, I want to save jobs that interest me so t...	DONE	 JAB Sprint 2	 Uma Sruthy Gajula
		JAB-10	As a job applicant, I should see jobs which matches my pr...	DONE	 JAB Sprint 2	 Aditya Reddy Sabb...
		JAB-9	As a job applicant, I want to edit and update my profile, so...	DONE	 JAB Sprint 1	 Vineela
		JAB-2	As a job applicant, I want to be able to Register/Login to th...	DONE	 JAB Sprint 1	 Uma Sruthy Gajula

We were unable to fully implement JAB-7 user story “As a job applicant, I want the bot to help generate a custom cover letter for each job so that I can apply more effectively” during this sprint. This feature creates custom cover letters using LLMs, we were able to do this as a standalone activity but since most of our effort went into completing the extension and connecting it to the frontend and backend, we were unable to add it to the dashboard.

## Issue Tracking:

<input type="checkbox"/>	0 Open ✓ 10 Closed	Author ▾	Label ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<input type="checkbox"/>	<b>Jab 26</b> ✓ #10 by adiboy6 was merged 5 days ago						
<input type="checkbox"/>	<b>Jab 4 chromium extension</b> ✓ #9 by khavin was merged 2 weeks ago						
<input type="checkbox"/>	<b>Jab 26</b> ✗ #8 by adiboy6 was merged last week						1
<input type="checkbox"/>	<b>Jab 39 scraper to crawl through job boards</b> ✓ #7 by adiboy6 was merged 3 weeks ago				2		1
<input type="checkbox"/>	<b>Merge backend with UI and extension</b> ✗ #6 by khavin was merged on Nov 4 • Review required						1
<input type="checkbox"/>	<b>Merge dashboard home page UI</b> ✓ #5 by khavin was merged on Nov 4 • Review required						
<input type="checkbox"/>	<b>Scraper to crawl through job boards</b> #4 by adiboy6 was closed 3 weeks ago				1		
<input type="checkbox"/>	<b>Web extension to fetch user profile data</b> #3 by adiboy6 was closed 2 minutes ago				1		
<input type="checkbox"/>	<b>Create a User Profile page with all the relevant fields</b> #2 by adiboy6 was closed on Nov 3						
<input type="checkbox"/>	<b>Create a Login/Registration page</b> #1 by adiboy6 was closed on Nov 3						