

Job Tracker Capstone Project Final Report

Project Archive

CS467/Spring/2025

Tony Chan, Jacky Kuang, Katherine Jaeger

6/4/25

Table of Contents

Introduction.....	3
Functions.....	3
Development Efforts.....	3
Tools Used.....	4
Backend.....	4
Languages & Frameworks.....	4
Database.....	4
API & Testing.....	4
Deployment.....	5
Schema Design.....	5
Frontend.....	5
UI & Assets.....	5
Web Development.....	5
Project Management.....	5
GitHub Projects.....	5
Development Tools.....	5
Editors.....	5
Version Control.....	5
Conclusion & Future Directions.....	6

Introduction

Landing a job in today's competitive market requires more than just submitting applications—it demands organization, self-awareness, and strategic skill-building. Many job seekers, particularly students and early-career professionals, find it difficult to manage multiple applications, keep track of follow-ups, and identify which skills they need to develop. To address this gap, our team set out to create a tool that would bring clarity and structure to the job search process. *Enter Jobscript.*

Jobscript is a web-based application designed to streamline job hunting by helping users organize and manage their career pursuits in one centralized platform. Whether they're actively applying for positions or preparing for future opportunities, *Jobscript* provides tools to track applications, maintain networking contacts, and monitor growth in key technical skills. Its intuitive layout and targeted features make it ideal for job seekers who want a clearer view of their progress and greater control over their career planning.

Software Functions

As a *Jobscript* user, the entire job search process becomes more structured and less stressful. Upon logging in, users are immediately greeted by a personalized dashboard where they can easily add new job opportunities, update application statuses, and track every step—from submitting a résumé to completing interviews and receiving offers. Each job entry includes relevant details like company name, role title, contact information, and progress stage, so users never lose track of where they stand.

One of *Jobscript's* main features is its skill-matching system. When a user logs a new job, *Jobscript* pulls the required technical and soft skills from the posting, allowing them to compare the job's demands to their current self-assessed skill profile. This helps users see where they're a strong fit and where they might need to grow, skill by skill. The result is a clear, actionable path forward—helping users decide what to learn or improve before their next interview.

Jobscript also helps users maintain a personal library of contacts—recruiters, mentors, hiring managers, and others—so their professional network grows alongside their job search. With visual summaries and organized data at their fingertips, *Jobscript* empowers users to stay focused, feel prepared, and apply with purpose. Whether they're actively hunting for a job or just exploring what roles align with their strengths, *Jobscript* is designed to provide clarity, confidence, and control throughout the career journey.

Development Efforts

Our initial plan was to create a Job Tracker web application with the listed features on the EECS Project Portal. For our technology, we planned to use React on the Frontend, Express.js on backend, and Supabase for authentication which also provided a PostgreSQL database. The goal here was to create a backend API that interacts with the Supabase so that the frontend

React is unable to interact with the database directly. This would have made the app secure while providing a more dynamic and interactive experience.

We worked on the backend API to implement the listed features and were able to accomplish account creation, job and skill tracking, and a contact system for networking with other users. However, we weren't able to implement the use of any API's other than the one used to interact with Supabase. Our skill tracking is also somewhat different than what's listed as we had to create a job and user skill to keep track of the skills that are part of a job requirement and the user's own skills. We were also unable to complete the skill matching and improvement features due to a lack of time.

Plan Deviations

During the second half of the semester we decided to change our stack from having separate frontend and backend to just having server side rendering on the backend. It was after our second progress report where we realized that we couldn't complete and integrate a separate frontend in time. This was due to poor initial planning and underestimating the amount of work involved which made it difficult to complete both frontend and backend. While we didn't have the React files ready, we did have the HTML and CSS files which allowed us to quickly switch to using EJS. We also had to refactor the backend API into server side rendering with EJS but it wasn't too difficult since their structure was fairly similar.

Tools Used

Backend

Languages & Frameworks

JavaScript using Node.js and Express.js for server-side logic

Embedded JavaScript (EJS) for dynamic HTML templating

Database

Supabase (PostgreSQL) with custom SQL schema

Tables include companies, jobs, users, job_skills, tags, contacts, and auth tokens
SQL_create_db_tables

API & Testing

RESTful API routes with tests using REST Client extension in VSCode

Deployment

Hosted backend and database on Render

Schema Design

Entity Relationship Diagrams (ERDs) created with Lucidchart

Frontend

UI & Assets

Figma and Procreate for designing layouts and custom images

Web Development

HTML/CSS with dynamic updates using JavaScript and EJS templates

Project Management

GitHub Projects

Used GitHub Project Boards to organize tasks in Agile and Kanban format
Columns included: Backlog, In Progress, Review, and Done
Linked issues and pull requests to project board cards for better traceability

Tony found GitHub Projects to be an effective and user-friendly tool for managing tasks using Kanban and Agile project workflows.
Tasks were linked directly to pull requests and commits, making progress tracking and project visibility straightforward within the GitHub interface.

Development Tools

Editors

Visual Studio Code (VSCode) for frontend/backend development
Visual Studio 2022 for broader testing and code editing support

Version Control

Git and GitHub for managing branches, feature commits, and team collaboration

Conclusion & Future Directions

The development of *Jobscrip*t allowed our team to take an abstract idea—organizing and empowering the job search process—and turn it into a functioning, user-focused web application. While we encountered challenges, particularly with project scope and frontend/backend integration, we were able to adapt by shifting to server-side rendering and refining our goals based on time and resources. The result is a solid foundation that meets core user needs: tracking job applications, managing contacts, and mapping job skills to personal skills. Through this project, we gained valuable experience with full-stack development, project management, and collaborative problem-solving.

Looking forward, *Jobscrip*t offers many opportunities for further development. Integrating external APIs from platforms like LinkedIn, Indeed, or Glassdoor could allow users to automatically import job postings and access richer job data. Expanding the skill-matching system with personalized improvement recommendations or progress analytics could enhance user engagement. A built-in messaging feature would also support networking and reference requests, helping users make the most of their stored contacts. With these additions, *Jobscrip*t could evolve into a comprehensive career companion—equipping users not only to stay organized, but to move strategically and confidently through each phase of their job journey.