Department of Computing Science

School of Computing

Instructor: Mandeep Pannu

Students:

Ngan Phan - 300180184
Aaradhya Srivastava - 300187055
Angad Kahlon - 300187930
Sadaf Faizi - 300186394

COMP 455 - AB1: Extreme Computing

University of the Fraser Valley

December 5, 2024

CareerFit: A Job Recommendation System

1 Abstract

CareerFit is a web-based platform designed to simplify the job search process by connecting users with opportunities that align with their skills, preferences, and career goals. The system uses user-provided information, such as resumes, skills, and job preferences, and matches it with job postings fetched from popular job sites like Indeed and Glassdoor. This report outlines the platform's functionality, data collection process, and the backend system that powers the recommendation engine. Built using Python and Flask, with SQLite as the database, CareerFit combines modern algorithms like Min Hashing and Locality-Sensitive Hashing (LSH) for effective job matching.

2 Introduction

Finding the right job can be overwhelming for job seekers, especially when sifting through hundreds of postings. CareerFit aims to solve this problem by offering a personalized job recommendation system. By collecting essential details from users and leveraging machine learning techniques, CareerFit matches users with jobs that best fit their profiles. The platform is designed with two main goals: user-friendliness and effective matching. This report explains how CareerFit is structured, from its frontend interface to its backend system.

3 Purpose

The purpose of CareerFit is to create a reliable and easy-to-use job recommendation system that helps job seekers find suitable positions without spending hours browsing job boards. The platform is meant to streamline the job application process, offering recommendations that align with the user's skills, experience, and preferences. It also benefits employers by ensuring their job postings reach the right audience, improving hiring efficiency.

4 Scope

CareerFit is targeted at individuals actively searching for jobs and companies looking to hire suitable candidates. It primarily focuses on job opportunities in technology-related fields but can be expanded to other industries. The platform is designed for both local and remote job searches. Users can interact with the system through a simple web interface that collects their information and displays personalized job recommendations.

5 Features of CareerFit

5.1 Frontend Functionality

The frontend of CareerFit serves as the primary user interface. It is simple yet functional, focusing on gathering user information and presenting job recommendations in an organized manner. The frontend consists of two main pages:

- **Page 1: Registration Form:** This page collects personal and professional details from the user, such as name, email, work experience, education, and skills. It also allows users to upload their resumes, which are automatically processed by the backend to extract additional details.
- **Page 2: Recommendations Page:** Once the user submits their details, this page displays a curated list of job opportunities tailored to their profile. The recommendations are based on a combination of user preferences and resume data.

The interface is designed to ensure ease of use, with clear instructions and minimal distractions. Users can update their preferences or re-upload their resumes to get updated recommendations.

5.2 Backend System

The backend of CareerFit is the core engine that processes user data and generates job recommendations. It uses Python and Flask to handle user requests and manage data. Here's a breakdown of its main functionalities:

5.2.1 Data Collection and Storage

The backend collects user information from the frontend, such as resume details and job preferences, and stores it in an SQLite database. This database is lightweight yet efficient, ensuring quick access to both user data and job postings. Job postings are fetched from external platforms like Indeed and Glassdoor using APIs. These postings are cleaned and formatted for better processing.

5.2.2 Job Matching Logic

The backend employs advanced algorithms like Min Hashing and Locality-Sensitive Hashing (LSH) to match users with jobs. Min Hashing helps compare user profiles and job descriptions to identify similarities, while LSH speeds up the search process by narrowing down the most relevant matches. The result is a personalized list of jobs that closely align with the user's profile.

6 System Design

6.1 Frontend Design

The frontend of CareerFit is built with simplicity in mind. Users interact with the system through a clean web interface:

- The registration form is designed to capture all essential details in a structured format, ensuring that users don't feel overwhelmed.
- The recommendations page presents job listings in an easy-to-read layout, with options to sort or filter based on factors like location, salary, or job type.

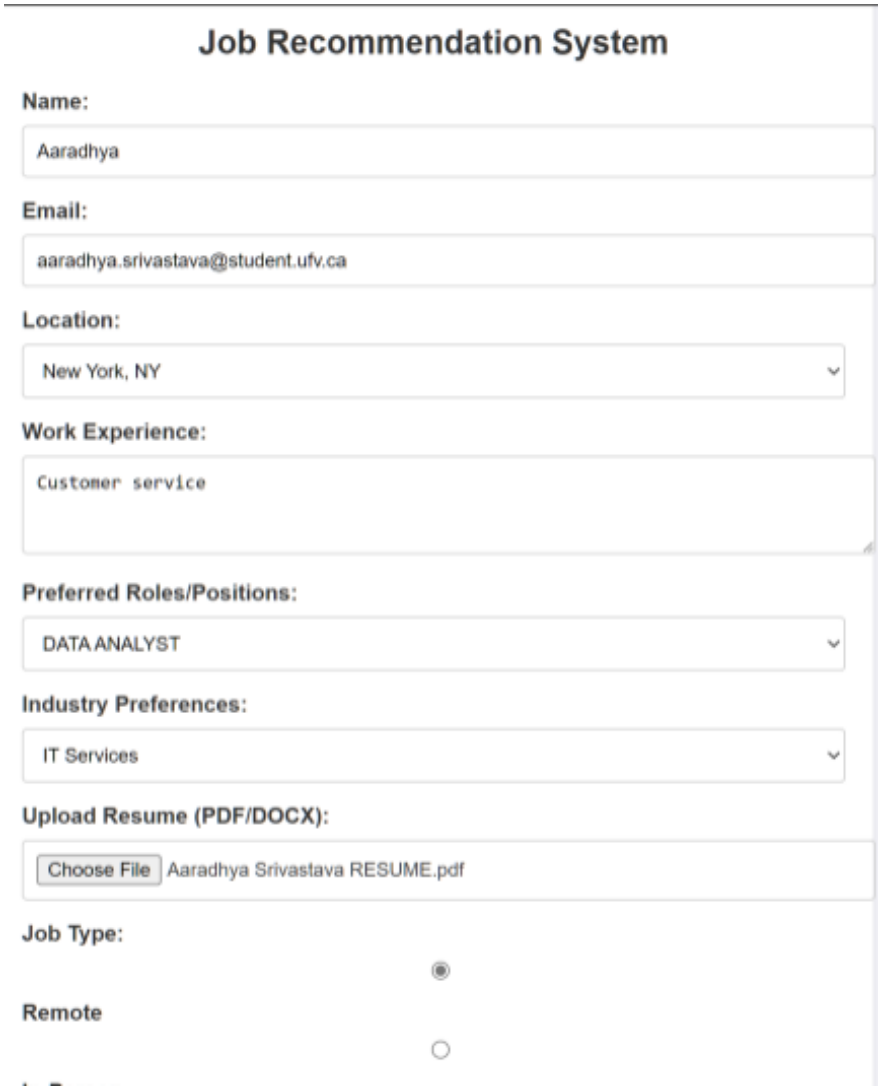

Figure 1: Prototype of the front-end design

6.2 Database Design

The SQLite database is organized into two main tables:

- **User Data Table:** Contains personal information, skills, work experience, education, and job preferences for each user.
- **Job Postings Table:** Stores information about job postings, including job titles, descriptions, company names, locations, and required qualifications.

The database design ensures efficient data retrieval and supports seamless integration with the recommendation algorithms.

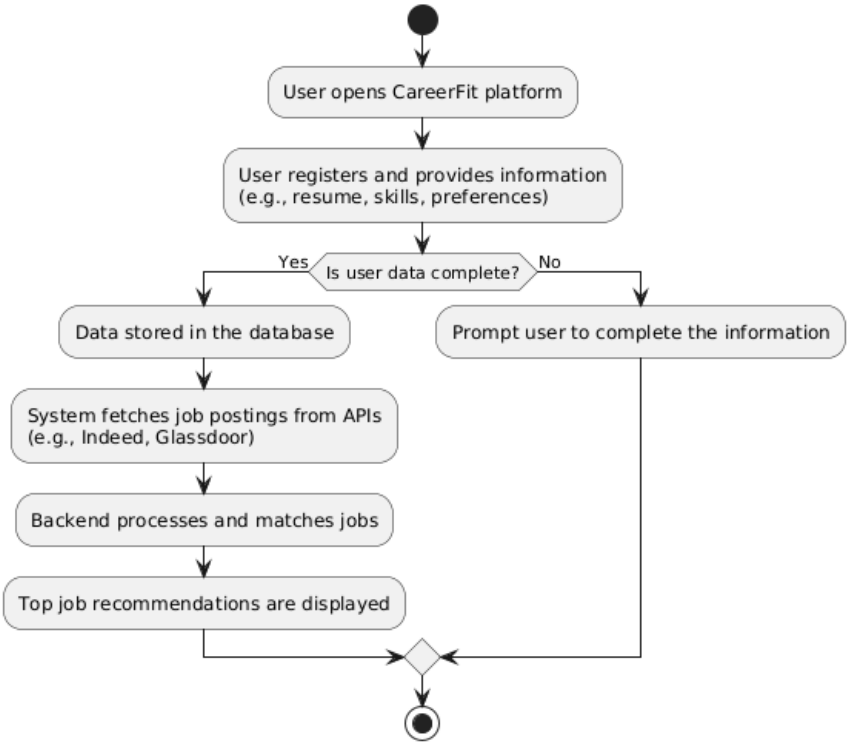

Figure 2: Flowchart representing the system flow of CareerFit

6.3 Backend Implementation

The backend is implemented using Python and Flask, with clear and organized code to handle user requests. Flask routes manage the communication between the frontend and backend, ensuring that data flows smoothly. The backend is also responsible for:

- Fetching new job postings and updating the database regularly.
- Running matching algorithms to identify suitable jobs for each user.
- Sending personalized recommendations back to the frontend for display.

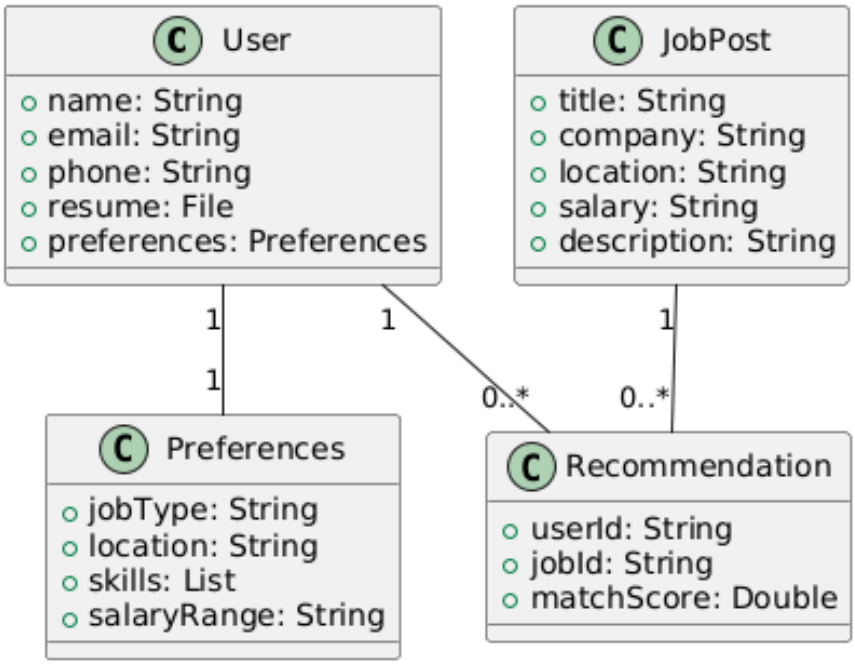

Figure 3: Domain model showing the relationships between system entities

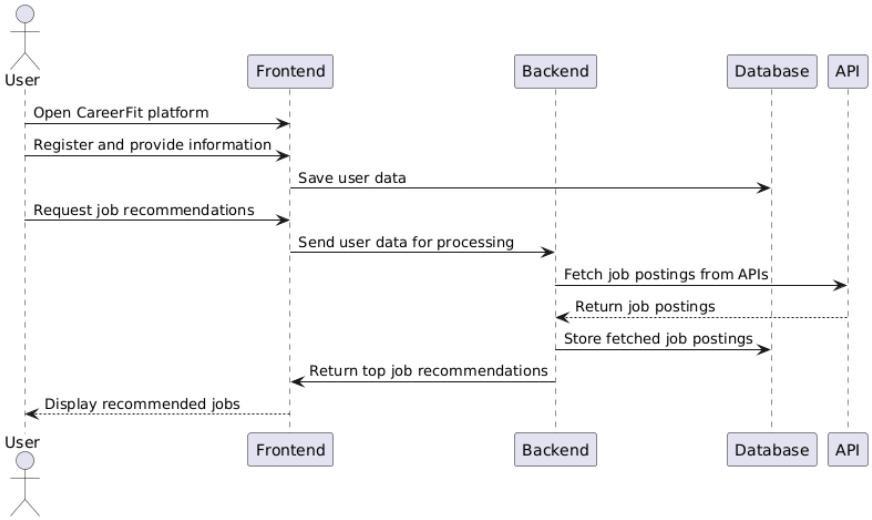

Figure 4: System sequence diagram (SSD) showing the interaction for job recommendations

7 Conclusion

CareerFit is a smart and helpful tool for today's job seekers. It makes finding the right job easier with its simple and easy-to-use design. The platform uses advanced algorithms to match people with jobs that fit their skills and preferences, saving users time and effort. By connecting job seekers with the right opportunities, CareerFit also helps employers find suitable candidates more quickly.This system has the potential to grow and improve in the future. Adding features like notifications, tracking job applications, and supporting jobs from more industries and regions would make it even more useful. It could also benefit from smarter technology to give even better job recommendations. Overall, CareerFit is a practical solution to the challenges of job hunting. It saves time, reduces stress, and helps people find jobs that match their goals and abilities. With more updates and improvements, it can become an even more valuable tool for job seekers everywhere.

8 Future Work

While CareerFit is functional, there is always room for improvement. Future enhancements could include:

- Expanding the job database to cover more industries and geographic regions.
- Incorporating machine learning models to refine job ranking and predictions.
- Adding features like job application tracking and real-time notifications.

These improvements will make CareerFit even more valuable for job seekers and employers.

References

- Locality Sensitive Hashing (LSH): The Illustrated Guide — Pinecone. (2014). Pinecone.io. https://www.pinecone.io/learn/series/faiss/locality-sensitive-hashing/
- McLendon, R. (2019). Building a Recommendation Engine with Locality-Sensitive Hashing (LSH) in Python. Learndatasci.com. https://www.learndatasci.com/tutorials/building-recommendation-engine-locality-sensitive-hashing-lsh-python/
- MinHash LSH — datasketch 1.6.5 documentation. (2024). Ekzhu.com. https://ekzhu.com/datasketch/lsh.html
- Santo, N. D. (2022, October 21). Fast Document Similarity in Python (MinHashLSH). Codemotion Magazine. https://www.codemotion.com/magazine/backend/fast-document-similarity-in-python-minhashlsh/
- How LinkedIn Uses Machine Learning in its Recruiter Recommendation Systems. (n.d.). KDnuggets. https://www.kdnuggets.com/2020/10/linkedin-machine-learning-recruiter-recommendation-systems.html
- Behrain, A. (2023, June 13). Creating an AI-powered Job Recommendation System. Medium. https://medium.com/@abbasbehrain95/creating-an-ai-powered-job-recommendation-system-50ce1cd12d36
- Kenthapadi, K., Le, B., Venkataraman, G. (2017). Personalized Job Recommendation System at LinkedIn. Proceedings of the Eleventh ACM Conference on Recommender Systems. https://doi.org/10.1145/3109859.3109921
- Flask. (2024). Welcome to Flask — Flask Documentation (3.0.x). Palletsprojects.com. https://flask.palletsprojects.com/en/stable/