SC HUNT

DSCI 551: Final Project Report

Team 56: Deepanshu Jain, Yashika Goyal and Janak Thakkar

Drive Link: Final Project Team 56 | YouTube Link: Project Video

Deepanshu Jain jaind@usc.edu

- Currently enrolled in MS in Applied Data Science at USC, expected graduation in May 2025.
- Completed B.Tech in Computer Science and Engineering from GGSIPU, India in 2023.
- Enthusiastic about Machine Learning & applying skills to solve real-world problems.

Skills: Python, Pandas, Matplotlib, Scikit-Learn, Firebase, Selenium, REST API, SQL, PowerBI, Figma

Yashika Goyal yashikag@usc.edu

- Currently Pursuing MS in Applied Data Science from USC Viterbi School Of Engineering.
- Completed B.Tech in Information Technology from GGSIPU, India in 2023
- Dedicated to AI with a tireless commitment to staying at the forefront of technological advancements.

Skills: Python, Beautiful Soup, Flask, Firebase, Scikit Learn, HTML, CSS, JS

Janak Thakkar janaktar@usc.edu

- Currently enrolled in the USC graduate program in Applied Data Science.
- Graduated in 2021 with a degree in Computer Engineering from the University of Mumbai, India.
- Joined Larsen & Toubro Infotech Ltd. (now LTIMindtree) in 2021 as a Data Engineer.
- Worked on collaborative projects with teams of 10-20 as well as solo projects as an ETL developer.

Skills: Python, Firebase, PowerBI, REST API, Cloud Computing, Snowflake

Table Of Contents

1. Introduction
2. Planned Implementation
3. Architecture Design
4. Implementation
    - 4.1. Technology Stack
    - 4.2. Working
        - 4.2.1. Sample Data Collection & Pre-Processing
        - 4.2.2. Creating Databases & Inserting Data Using Hashing
        - 4.2.3. Front-end Design & Development
        - 4.2.4. Back-end Development & Integration
        - 4.2.5. Website Functionalities
            - 4.2.5.1. Users/Admins Account Management
            - 4.2.5.2. Viewing Job Opportunities
            - 4.2.5.3. Searching & Filtering
            - 4.2.5.4. Tracking Applied Jobs
    - 4.3. Implementation Snapshots
5. Challenges Faced
6. Individual Contribution
7. Conclusion
8. Future Scope

1. Introduction:

According to a recent statistical analysis based on a 2023 survey reveals a striking reality: Despite approximately 80% of job opportunities being sourced online and a plenitude of openings compared to job seekers, the job hunting journey remains arduous and mentally taxing for many. The process, marked by exhaustive searches, exploring countless career websites, filling out tedious application procedures, and subsequently struggling to maintain a comprehensive record of their job pursuits, often takes a toll on individuals, particularly students venturing into the realm of professionalism. This cycle not only consumes significant time and effort but also poses a considerable mental burden, emphasizing the urgent need for a streamlined and user-centric approach to job searching and application management.

This project endeavors to develop an all-encompassing portal for job seekers from diverse domains, sparing them the hassle of navigating through multiple websites in search of employment opportunities. Serving as a centralized hub, the platform will streamline job hunting, enabling users to initiate their job search effortlessly, aligning with their professional aspirations and passions. Distinguished by its comprehensive feature set, the platform will boast an array of functionalities surpassing those of conventional systems. Users will not only have the ability to search and filter job listings based on criteria like job title, type, and company name but also explore additional filtering options tailored to their preferences, such as location, salary range, and required skills. Moreover, we aim to facilitate seamless tracking of job applications, providing users with valuable insights into their application history and status, ensuring a more organized and efficient job search experience.

2. Planned Implementation:

Database Design:

The databases that we will be using in this project are represented through the below ER diagram, Fig[1] and are as follows:

1. Company - This database stores the list of companies that are posting job listings for the users.
2. User - This database will hold the login credentials for all the users.
3. Role - This database will help keep track of the access level a user may have in the web app to maintain security and confidentiality.
4. Jobs - This will be a distributed database based on the type of job whether it is a tech job or a non-tech job, and will contain the links to access the company page to complete the application.
5. Applications - This database will keep track of all the applications being made using the web app along with the status of completion of job applications mainly submitted or incomplete.
6. Archive - This database is a dump of all the jobs posted 2 years before the current date/month.

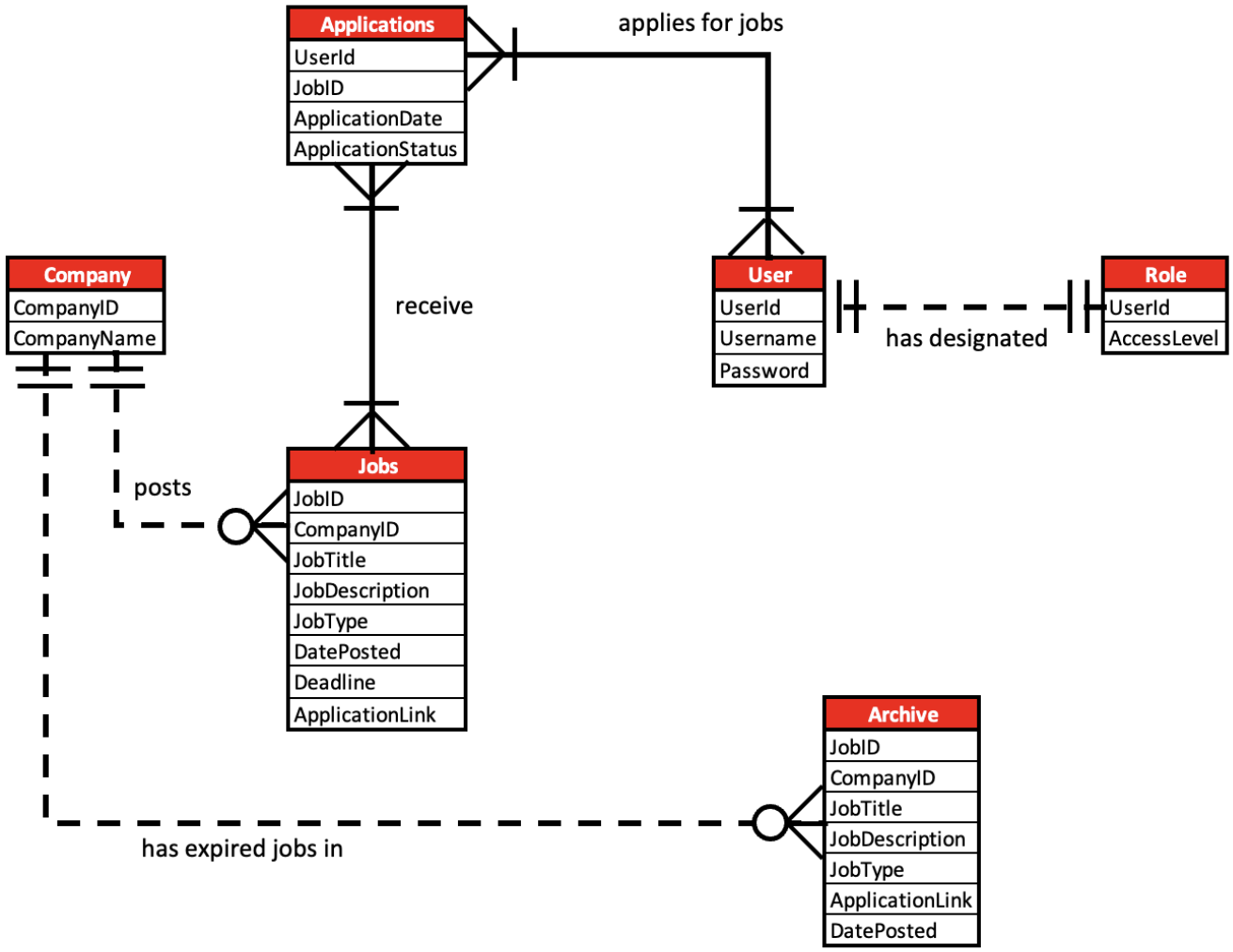

Figure 1: ER Diagram

3. Architecture Design:

Our project's overall working can be seen in the chart below, Fig[2]. We use web scraping to gather additional job postings weekly, ensuring users can access the latest opportunities. Complementing this, in the future, we will allow companies to seamlessly post job openings via a user-friendly form on the website. On the portal, users can view all the available job opportunities. Upon submitting job applications, users can promptly update their application status. Within the portal's dedicated applications tab, users can effortlessly track and manage their submitted applications, fostering organization and efficiency in their job-seeking endeavors. Additionally, our platform will incorporate data visualization tools for users to analyze application trends, job preferences, and success rates, empowering them to optimize their job search process effectively in the future.

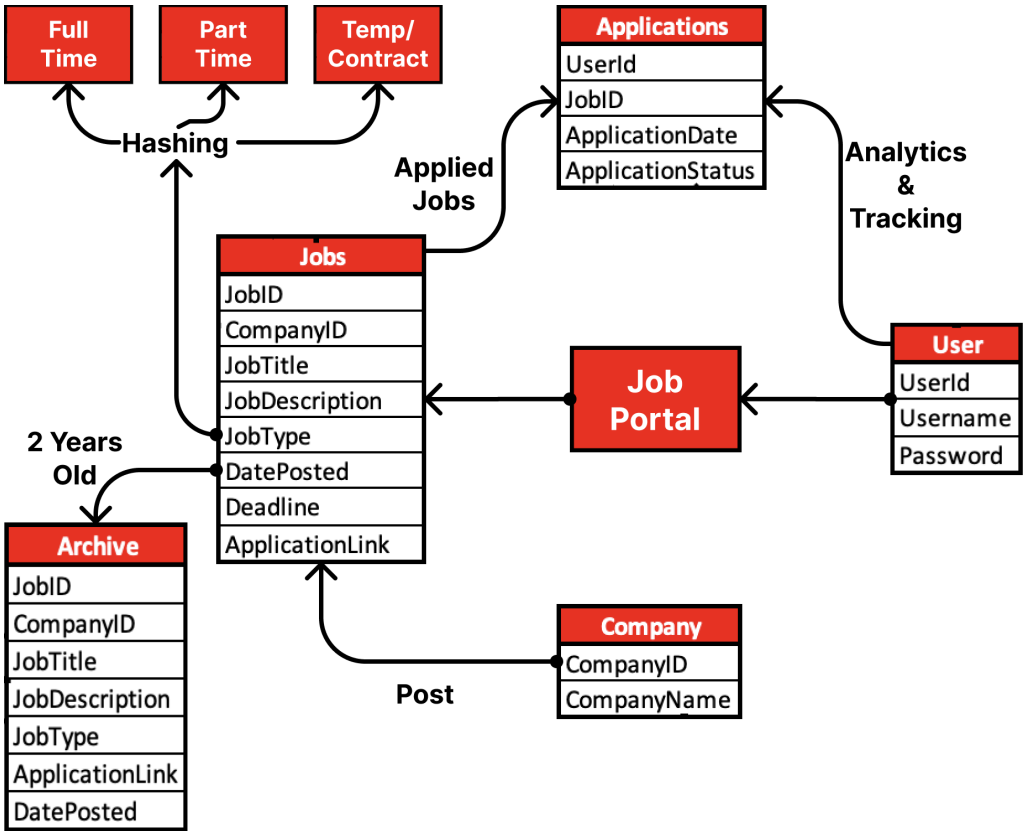

Figure 2: Workflow

4. Implementation:

4.1. Technology Stack:

- ➔ **Programming Language and Libraries:** Python for backend development with Pandas, NumPy for data manipulation, BeautifulSoup4 for web scraping, & Python REST API for Firebase connectivity.
- ➔ **Database Management:** Firebase for scalable, real-time data synchronization, utilizing its NoSQL approach for simplified schema changes.
- ➔ **Web-App Design and Development:** Canva and Figma for collaborative design, leveraging versatile tools to shape a powerful, seamless, user-friendly platform. HTML for structural foundation, CSS for aesthetics, JavaScript for interactivity, and Bootstrap for cross-browser compatibility, ensuring a user-centric, polished web interface.
- ➔ **Framework:** Flask chosen for lightweight, modular design, enabling tailored application development without unnecessary overhead.

4.2. Working:

4.2.1. Sample Data Collection & Pre-Processing

For the initial stage and testing, we acquired a dataset from Kaggle with similar attributes relevant to our project's use case (Available Here). Prior to integrating the dataset into our system, we conducted pre-processing tasks to optimize its usability. This involved removing unnecessary fields and addressing redundant data, which required cleaning and restructuring the dataset. Python libraries such as NumPy and Pandas were instrumental in this process. While the initial dataset served for testing purposes, our ongoing strategy involves real-world data extraction through web scraping to ensure the relevance and timeliness of job listings.

4.2.2. Creating Databases & Inserting Data Using Hashing

For our project, we utilized Firebase to create four specialized databases. Three of these are focused on job data storage, facilitating read-and-query operations. The fourth database is tailored for user management, enabling both read and write operations, with a specific emphasis on tracking users' applied jobs. This structure efficiently segregates data handling based on functionality, optimizing the system's overall performance and organization. Furthermore, employing a hashing technique based on job types (Full Time, Part Time, Temporary/Contract), we partitioned the data across the three job databases. This approach facilitated efficient data storage and retrieval, enhancing system performance.

Hashing based on job types offers several benefits for our project:

- ➔ **Simplified Data Retrieval:** By categorizing jobs into distinct types, users can efficiently filter and access relevant job listings based on their preferences.
- ➔ **Enhanced Scalability**: The partitioning of data allows for seamless scalability as the volume of job listings grows over time.
- ➔ **Improved Database Management:** Distributing data based on job types streamlines database operations, optimizing resource utilization and query performance.

4.2.3. Front-end Design & Development

Leveraging Figma, we crafted the visual identity and user interface design for the SC Hunt platform. This involved designing the logo and conceptualizing the web layout to ensure a visually appealing and intuitive user experience. Subsequently, we translated the design mockups into a functional website using HTML, CSS, and JavaScript. This process involved meticulous attention to detail to ensure consistency and responsiveness across various devices and screen sizes.

4.2.4. Back-end Development & Integration

Flask served as the backbone of our back-end development, enabling the creation of robust API endpoints and routing mechanisms. We implemented various routes to handle user interactions seamlessly. For instance, on the homepage, data is fetched from all three job databases to populate the job listings. Additionally, when a user applies for a job, the system updates the relevant data in the fourth database dedicated to managing user interactions. This ensures that users can conveniently track

their applications and maintain an organized profile within the platform. Integration with Firebase facilitated efficient data management and synchronization, enabling real-time updates and seamless communication between the front and back ends.

4.2.5. Website Functionalities

➔ **Users/Admins Account Management**

For user authentication in SC Hunt, we've implemented features for users to be able to Sign In/ Create an account either using simple Email-Password Combination or Google Authentication. This feature enables users to sign up or sign in to the platform using their Google accounts, providing a seamless and convenient login experience. Leveraging Firebase Authentication, users can authenticate securely, While also having the ability to Reset passwords just in case. Firebase Authentication offers several benefits like Convenience, Security, Reliability, and Scalability. By integrating Google Authentication with Firebase, we've enhanced the usability and security of SC Hunt, providing users with a streamlined and secure authentication mechanism that aligns with industry best practices.

➔ **Viewing Job Opportunities**

Job listings sourced from distributed databases are dynamically fetched and displayed to users, providing comprehensive insights into available opportunities.

➔ **Searching & Filtering**

The platform offers advanced searching and filtering capabilities to enhance the user's job search experience. Filtering based on Job Locations & Job Types: Users can refine their job search by filtering job listings based on locations and job types. This feature allows users to focus on jobs available in specific geographic areas and of particular types such as Full-Time, Part-Time, or Temporary/Contract roles. Initially, the filtering functionality is based on state codes within the United States of America (USA), providing users with localized job search results. Searching based on Job Titles: Users can also utilize the search functionality to search for specific keywords or phrases within job titles. This powerful search feature, combined with filtering options, enables users to conduct a focused job search tailored to their preferences and requirements.

➔ **Tracking Applied Jobs**

Users can conveniently track their applications within the platform upon applying for a job. The applied jobs list, accessible from the user's profile section, ensures streamlined management and organization of job applications, enhancing user productivity and engagement.

4.3. Implementation Snapshots:

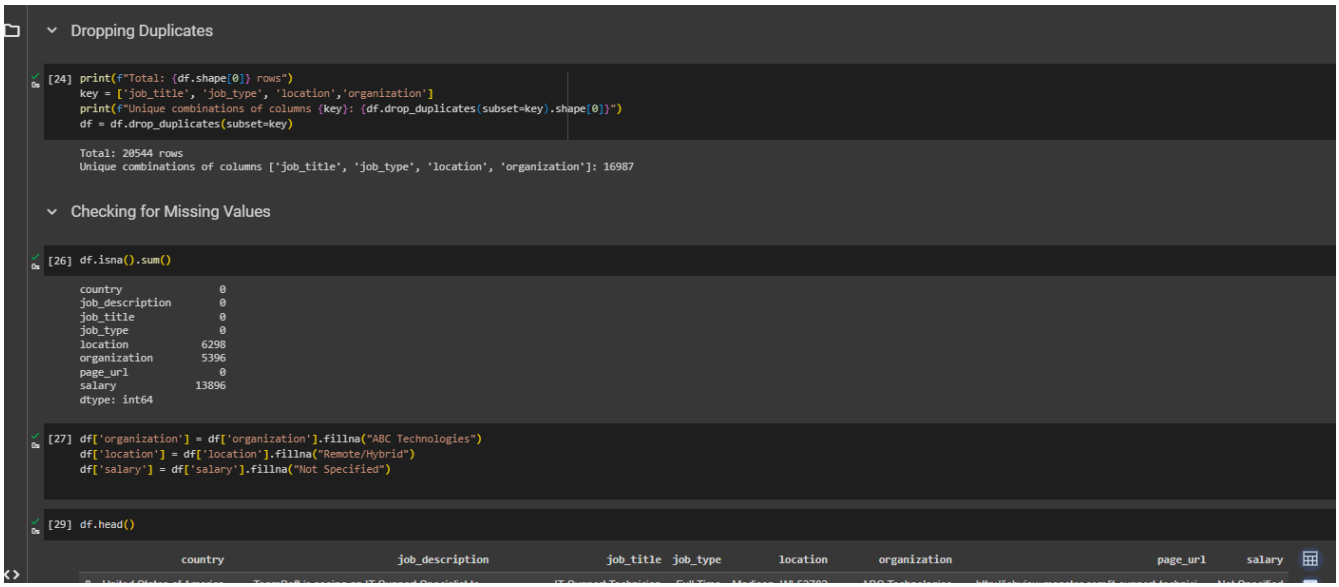

Figure 3: Data Cleaning and Preprocessing

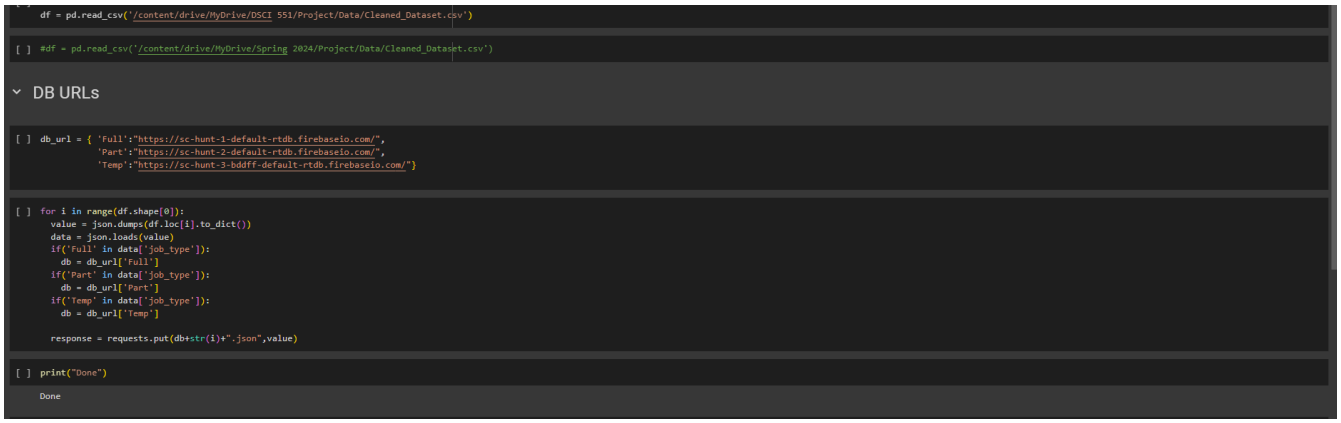

Figure 4: Data Insertion with Hashing

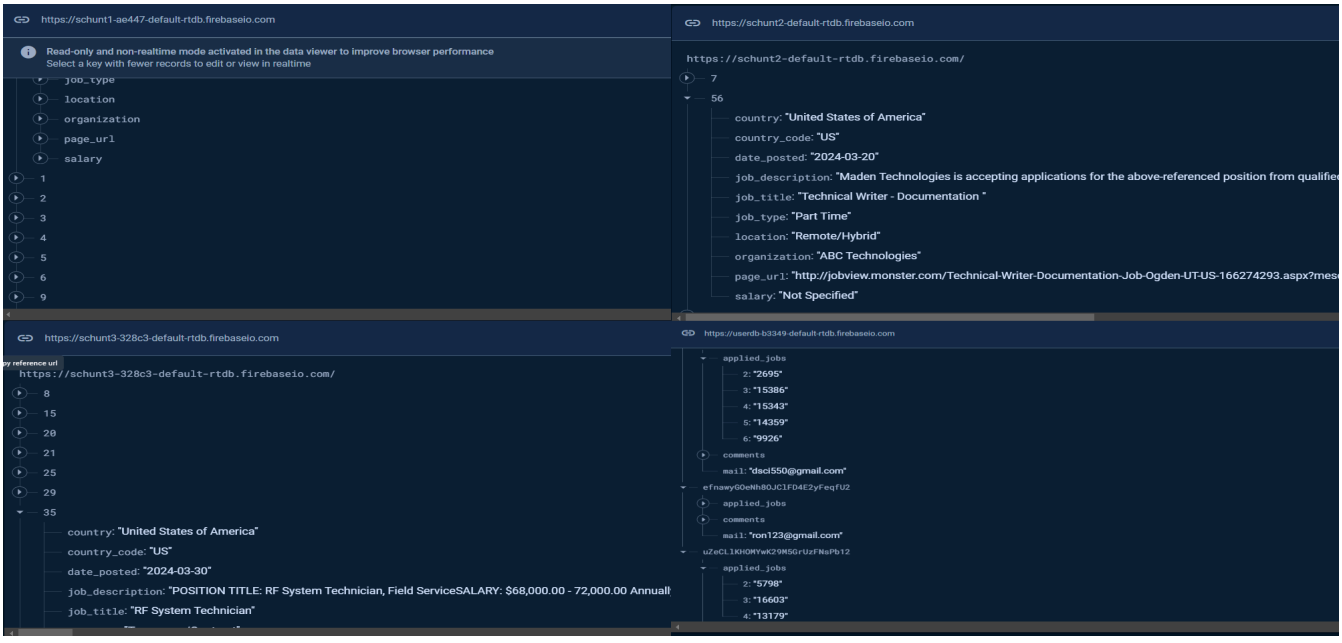

Figure 5: Distributed Databases in Firebase

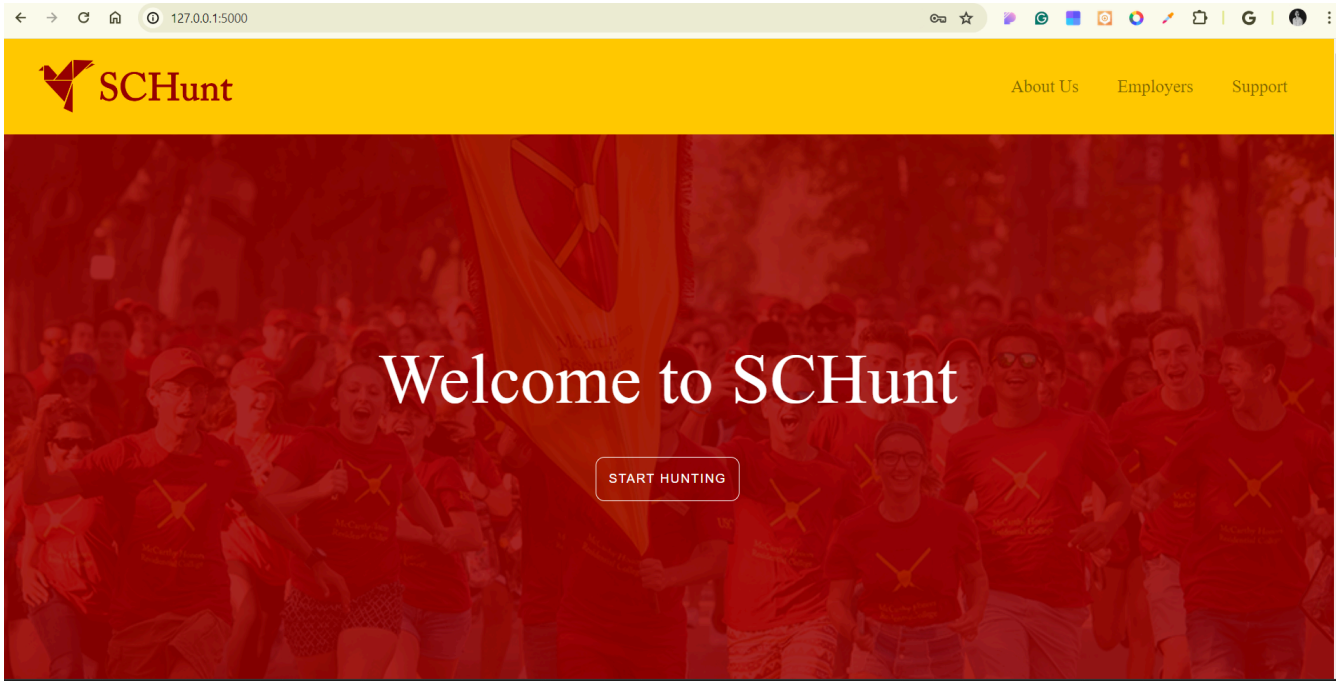

Figure 6: Homepage

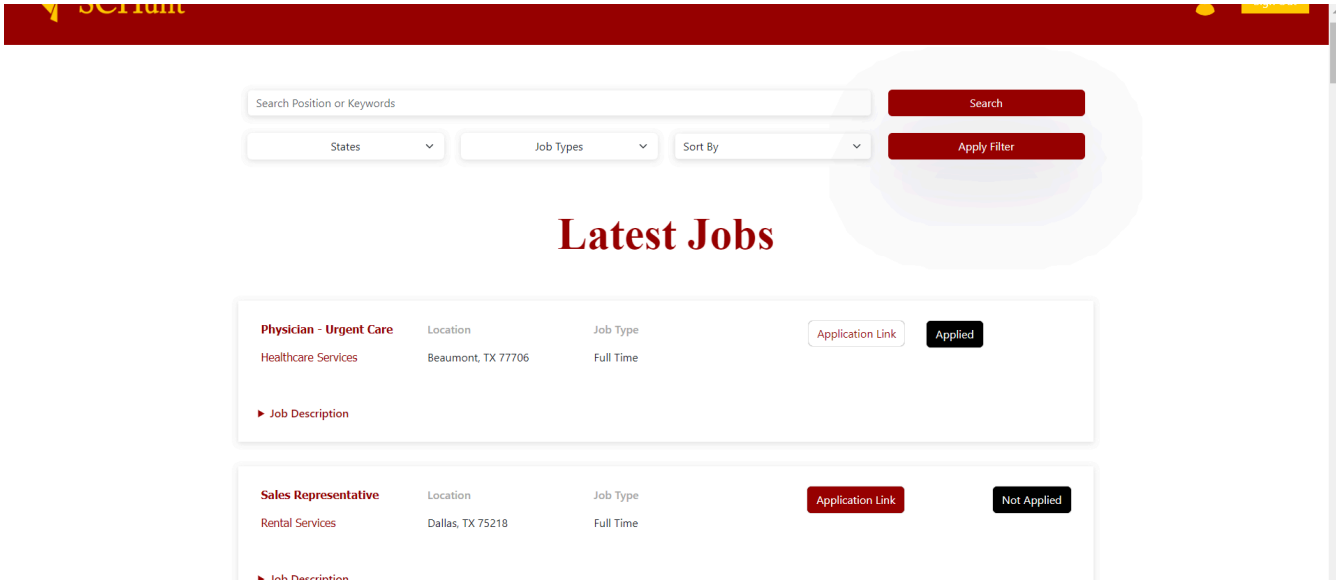

Figure 7: View Available Jobs

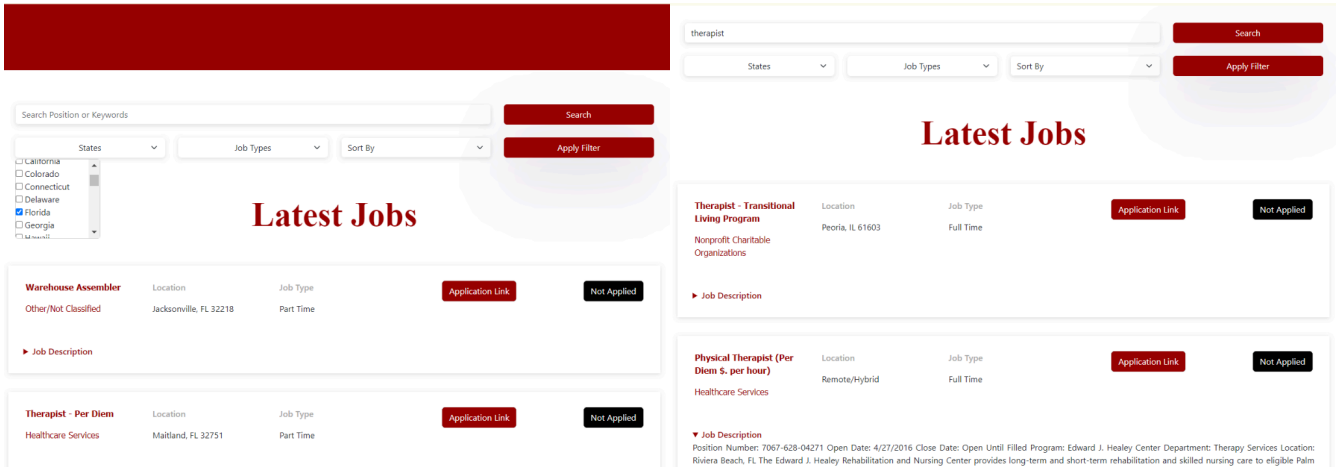

Figure 8: Searching and Filtering Jobs

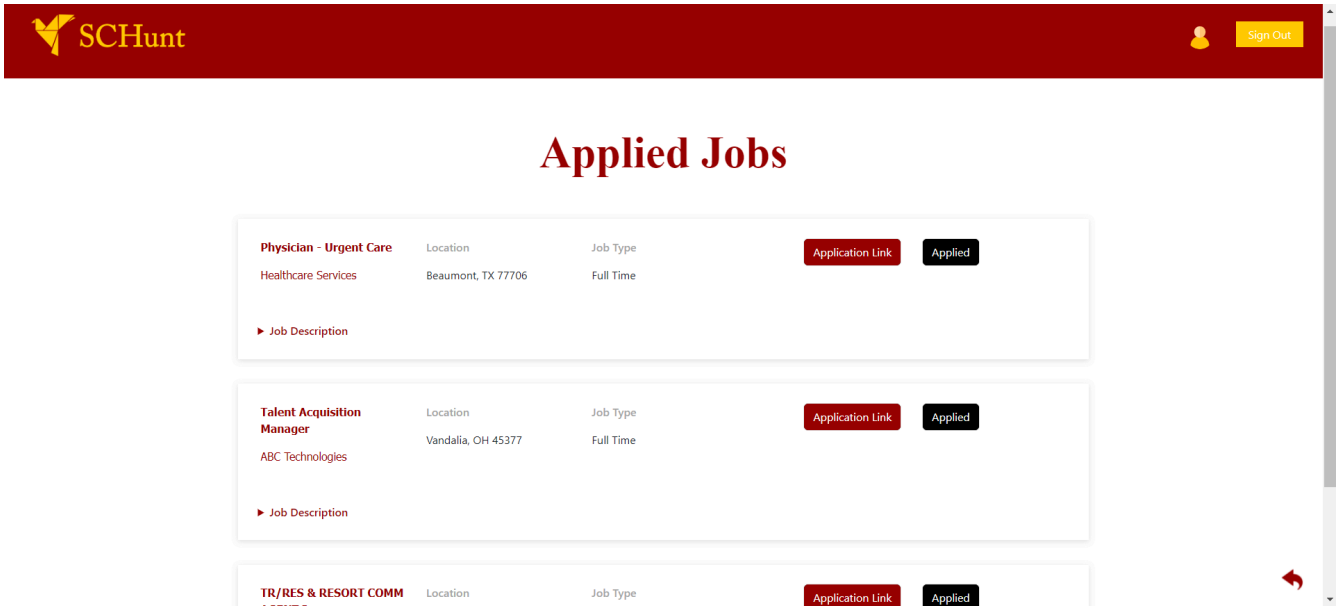

Figure 9: Tracking Applied Jobs

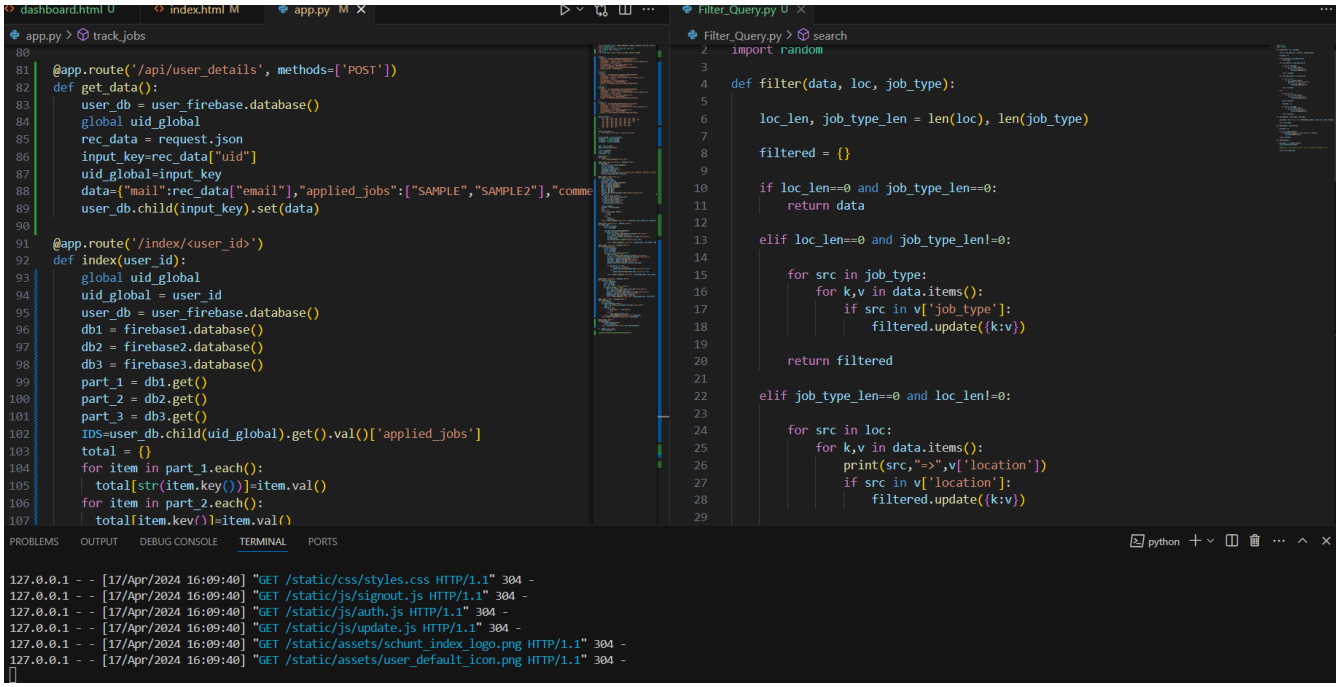

Figure 10: Backend Using Flask

5. Challenges Faced:

One of the significant challenges we faced involved the initial hashing function that was based on job titles. The sheer variety of job titles, which exceeded 20,000, made it difficult to implement an efficient hashing mechanism. To address this issue, we shifted our focus to job types, which offered a more manageable and effective solution for data distribution. Additionally, we encountered many challenges in maintaining the session variables throughout the process. This aspect proved tricky, impacting our ability to consistently track user sessions and preferences effectively.

6. Individual Contributions:

Task Name	Task Responsibility
Data Collection & Preprocessing	Deepanshu Jain
Frontend Development	Yashika Goyal
Data Insertion with Hashing	Janak Thakkar
Backend Development	Deepanshu Jain & Yashika Goyal

Integration of Technologies	Janak Thakkar & Deepanshu Jain
Final Report & Presentation	Deepanshu Jain, Yashika Goyal & Janak Thakkar

7. Conclusion:

In conclusion, SCHunt represents a comprehensive solution designed to revolutionize the job search experience. Through meticulous planning and execution, we've developed a platform that seamlessly integrates web scraping, database management, and user authentication, leveraging cutting-edge technologies like Flask and Firebase. This platform will not only simplify the application process for all the job seekers but also provide invaluable tools for tracking applications and analyzing trends. With a commitment to user-centric design and functionality, it is poised to redefine the landscape of job portals, offering a personalized, intuitive, and efficient platform for job seekers and employers alike.

8. Future Scope:

To ensure our platform stays current and engaging, we have plans to bring real-world companies on-board. This will enable real-time job updates directly from employers, ensuring that our listings remain relevant to job seekers. We also aim to provide personalized dashboards for each user to gain insights into their job application journey. These dashboards will provide valuable feedback on application patterns, preferences, and successes, allowing users to refine their job search strategies. Overall, our platform will continue to evolve to meet the needs of job seekers, keeping them informed, engaged, and successful in their job search.