Job Recommender Project Documentation

A) Overview of the project

The topic of our project is 'Job Recommendation System'. Being students ourselves, it's very difficult to find the right jobs based on our resumes. Currently, we end up going through most of the job descriptions and start manually checking if the job description has the skills mentioned that match with the skills that we have. Therefore, we are trying to solve this problem by providing job recommendations based on the resume that is uploaded by the user. This way, the manual process for keyword matching based on skills is not needed anymore.

B) Team Member Contributions:

Stage 1: We will look for a dataset that perfectly fits the scope of our project and then pre-process it. - Smit Trailokya - srt6@illinois.edu

Stage 2: Use a text extraction function to extract keywords from the resume and our dataset. - Maahi Patel - maahidp2@illinois.edu

Stage 3: Create a similarity function that matches the keywords extracted in the previous step and returns a similarity score.We will use the cosine similarity for this and return the top 5 jobs which will get matched by ranking them in descending order of their cosine similarity score. - Ojasvi Agarwal - ojasvia2@illinois.edu

Stage 4: Create an interactive user interface for our project by designing the front end with React and Flask. - Dhyey Pandya - dhyeyhp2@illinois.edu

C) Related work and used libraries/models/previous projects

1. https://nycdatascience.com/blog/student-works/r-shiny/match-skill-job-simple-job-recommendation-system/ → Related Project
2. https://ceur-ws.org/Vol-2077/paper6.pdf → Related Research Paper
3. https://pypi.org/project/linkedin-jobs-scraper/ → Library used to web scrape linkedIn data
4. https://towardsdatascience.com/overview-of-text-similarity-metrics-3397c4601f50 → Jaccard vs Cosine Similarity

D) How to use the software ? (Instructions to Setup and Run the Code)

1. **Environment Setup:**

    - Go to https://nodejs.org/en/ and download version 18.12.1 LTS
    - python (3.0 +)
    - Install virtualenv (optional)

    The project is python (3.0 +) based and would require installing a certain set of python libraries in an isolated environment.

a. Install virtualenv where virtualenv is a tool used to isolated python projects. This helps in ensuring the libraries and version of python used in this particular isolated environment doesn't get affected by the other versions of python and other libraries present on the system locally :

    - sudo pip install virtualenv

    If you get an error : sudo -H pip install virtualenv

b. Next, Navigate to where you want to store your code For this, create a new directory "
    - mkdir my_project && cd my_project

    Then inside the "my_project folder" create a new virtualenv and activate it :

    - virtualenv env

c. Next, activate the virtualenv:
    - source env/bin/activate

    Do a git clone and get course project on your local machine:

    - git clone <insert the final repo name>

B) How to Start the web application?

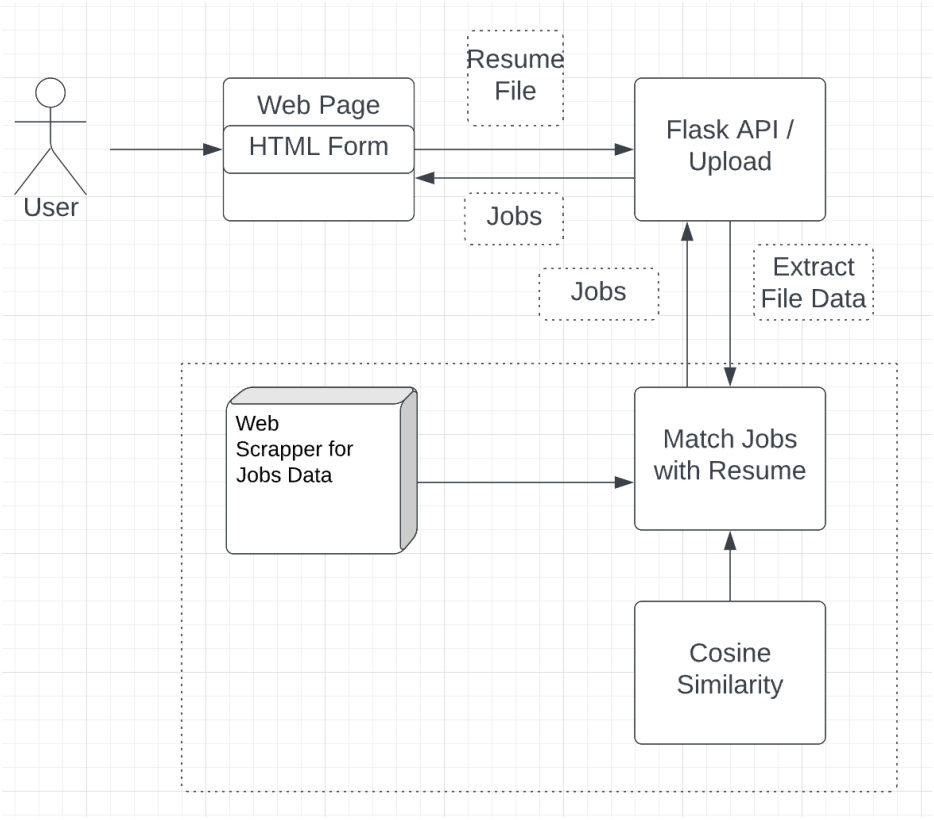
Frontend (In a dedicated Terminal)

cd client
npm install
npm start
Open in browser http://localhost:3001/

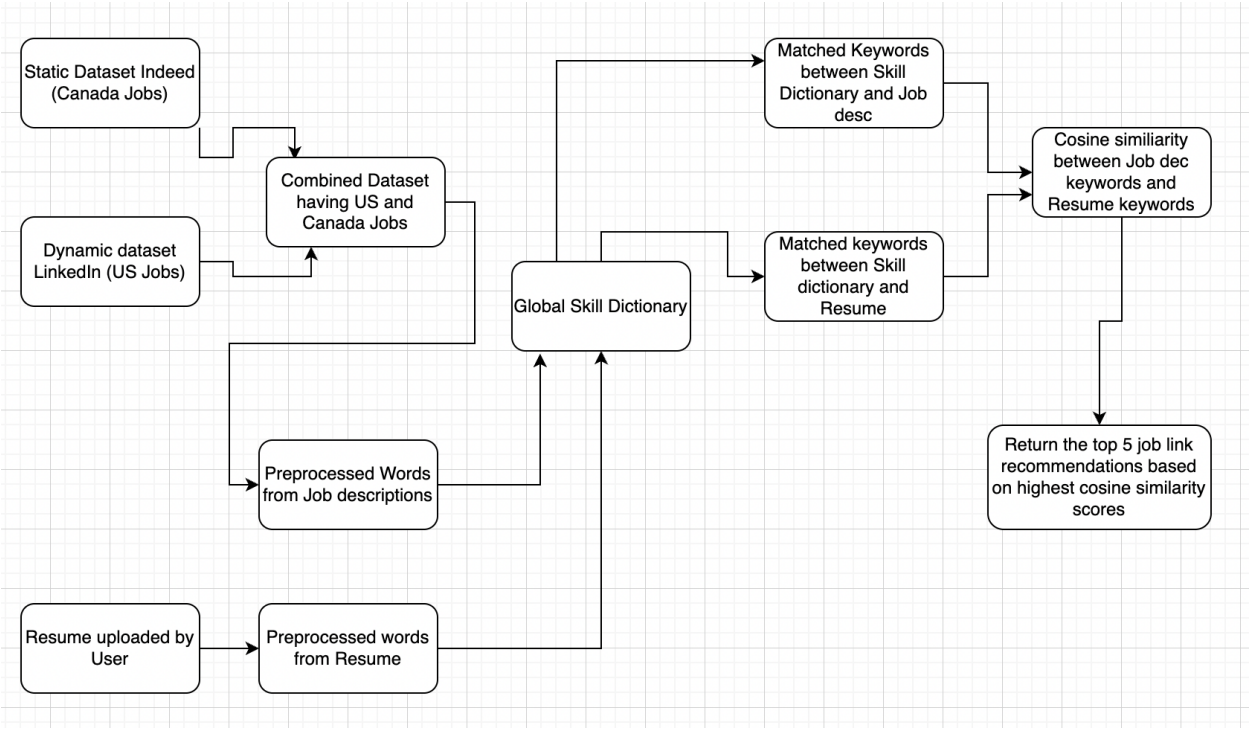
C) Backend (In a separate dedicated terminal)

cd backend
pip install -r requirements.txt
Install these packages separately to avoid issues while running the project if needed:
pip install nltk
pip install pyPDF2
python server.py

E) How is the software implemented ? (Code Structure and Workflow Diagram)



Overview of User Workflow



Detailed Workflow

The topic of our project is 'Job Recommendation System'. We are providing job recommendations based on the resume that is uploaded by the user. This way, the manual process for keyword matching based on skills is not needed anymore.

In order to provide job recommendations, we need to have a dataset. We have used a combination of 2 datasets - Indeed and LinkedIn. Indeed is our static dataset which consists of information regarding technology jobs present in the canada region. We have taken the Indeed dataset and integrated it with a LinkedIn one where we are fetching real-time jobs data through web scraping regarding software engineering jobs in the United States. We create a python dataframe called "jobs_info_df" which contains the combined data from Indeed and LinkedIn.

We have also gone through some of the common job openings across different web sites and came up with a global dictionary which will be used as the baseline for extracting keywords from job descriptions and resume. The dictionary is a mix of education level qualifications as well as skills coming from different domains like data analysis, Machine Learning, Different software methodologies followed, databases etc.

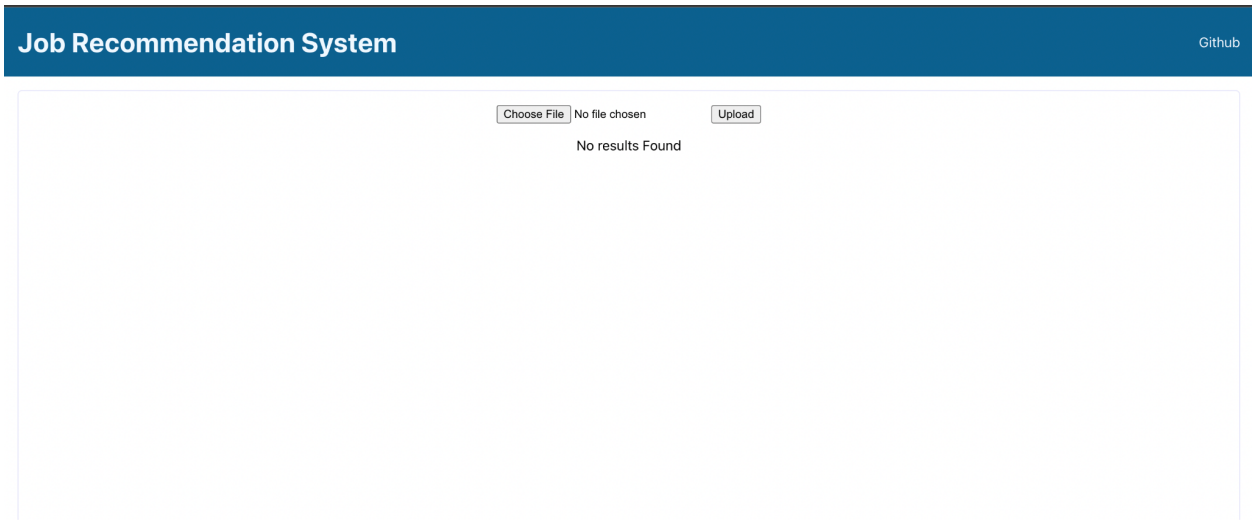
Next, we will go through the job descriptions one by one and extract the skill keywords from them. In order to extract the skill keywords, we will first clean the dataset, make them lowercase, remove the stopwords and get the unique words from it. Once we got the unique words from the job description, we will match them with our globally defined skillset dictionary and return the matched keywords. These keywords will be added as a column in the job description (jobs_info_df) dataframe.

A similar process will be followed for resume as well where we will go through every page of the resume and get a unique set of words after processing, these set of words will be matched with our globally defined skillset dictionary and will return the matched keywords.

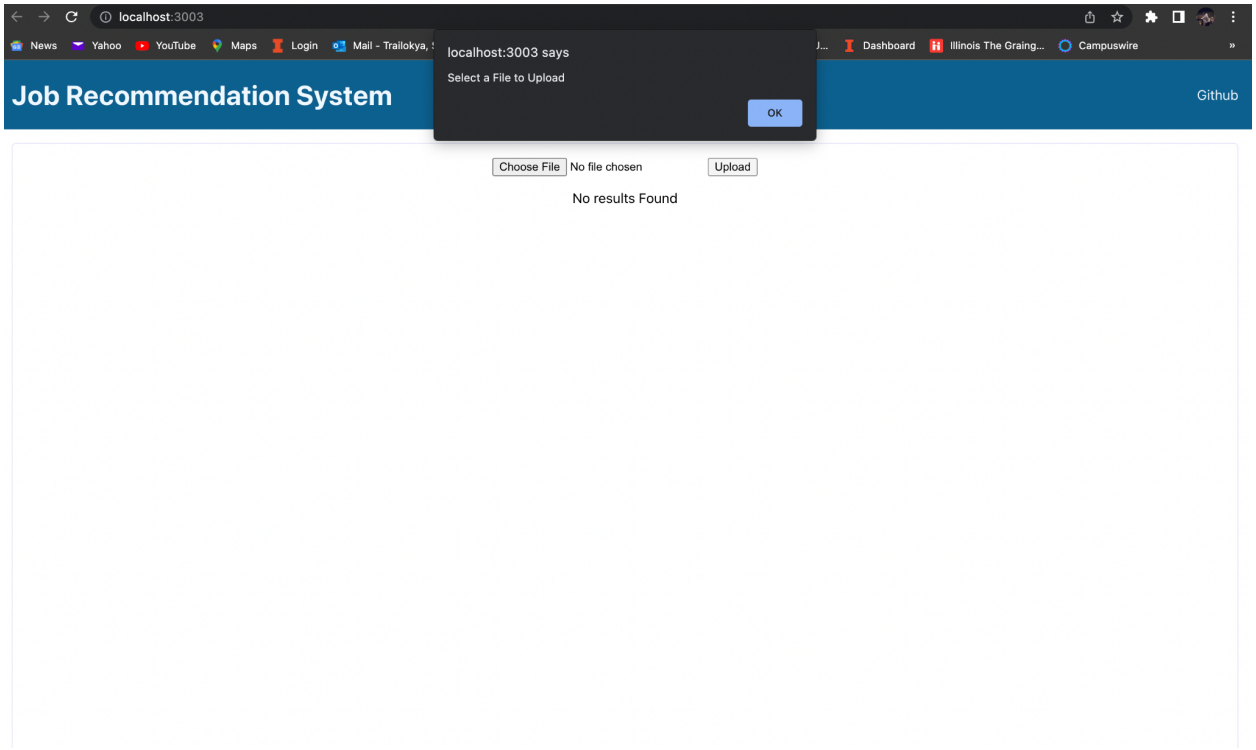
Now we have 2 sets of words, job-description specific keywords and resume specific keywords and our next goal would be to calculate the similarity between them. We will use the cosine similarity for this and return the top 5 jobs which will get matched by ranking them is descending order of their cosine similarity score.The reason for choosing cosine similarity was because we want to give preference for duplication since we might see instances of the same skill at multiple occurrences in either the resume or the job_description and we wanted to use that as a criteria while ranking jobs.

We have tried our best not to use static values in our project for variables like input path for resume, output path for web scraped data as well as job recommendations etc. and we are passing them dynamically through a configuration file.

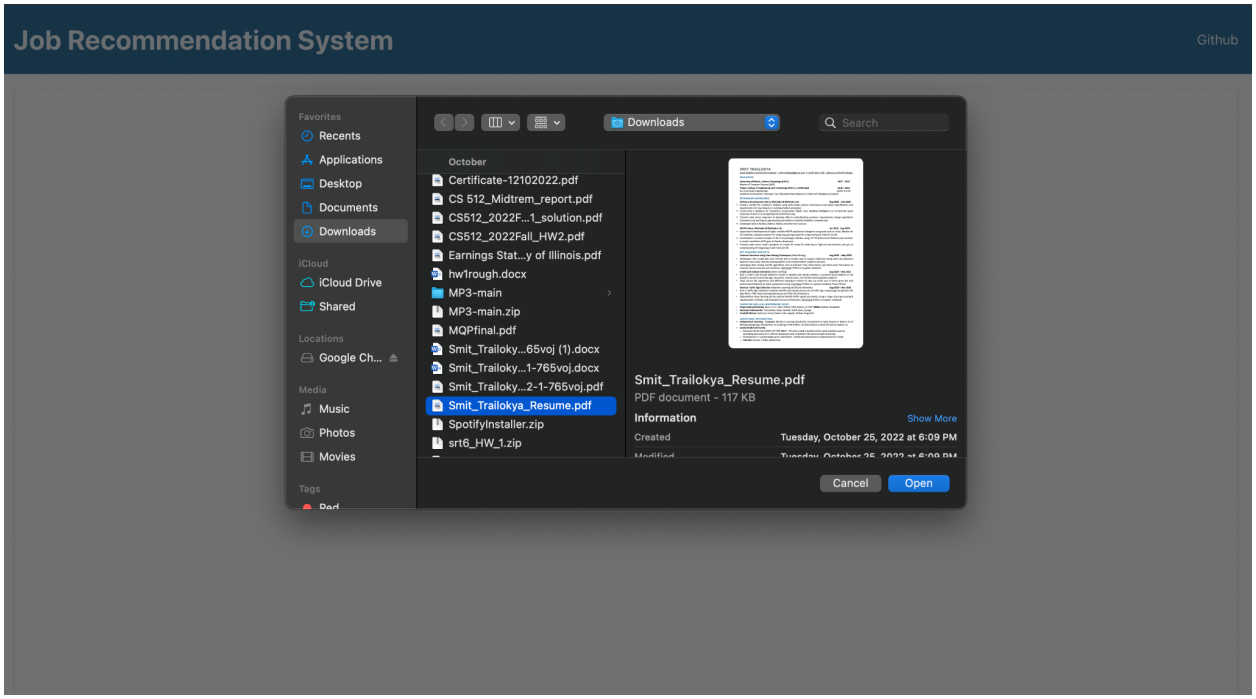
F) Main Results



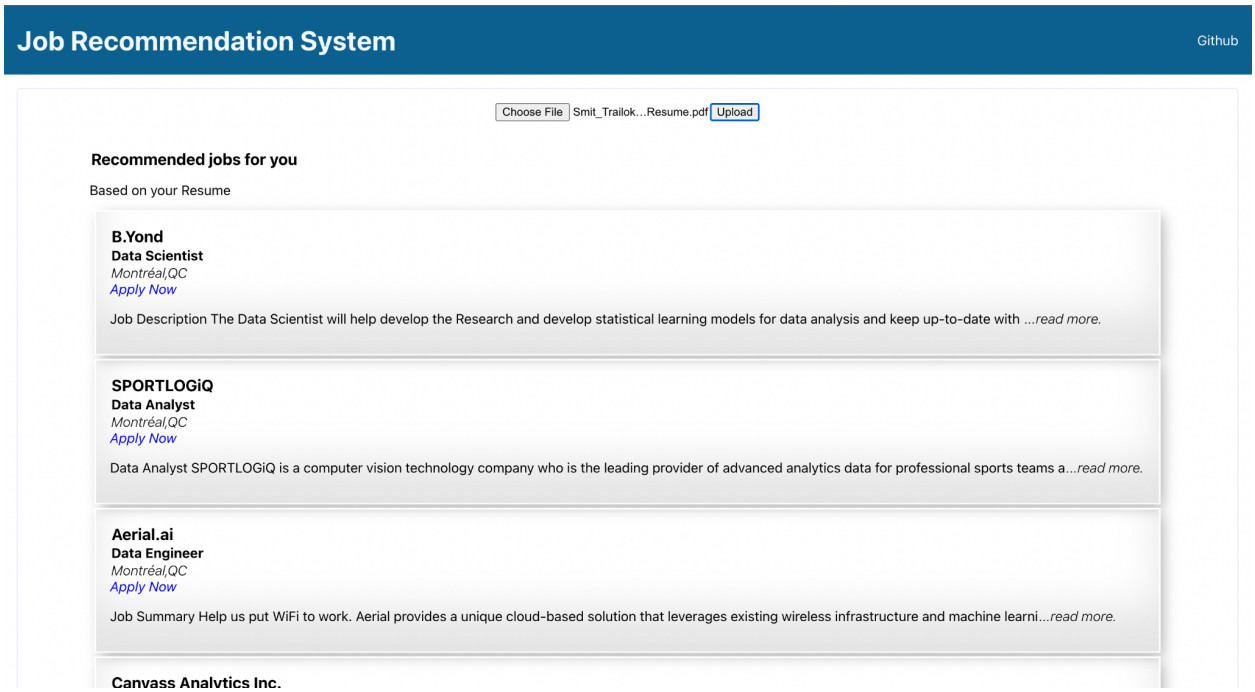
Home Page



Alert displayed if nothing is uploaded



Uploading Resume



Displaying recommended Jobs

G) Self-Evaluation

We were successful in completing all 4 stages of the project as described below and successful is getting the expected outcome where we were getting the best job recommendations having jobs coming from both the datasets.

Stage 1: We will look for a dataset that perfectly fits the scope of our project and then pre-process it. - Completed

Stage 2: Use a text extraction function to extract keywords from the resume and our dataset. - Completed

Stage 3: Create a similarity function that matches the keywords extracted in the previous step and returns a similarity score.We will use the cosine similarity for this and return the top 5 jobs which will get matched by ranking them in descending order of their cosine similarity score. - Completed

Stage 4: Create an interactive user interface for our project by designing the front end with React and Flask. - Completed

It was one of our stretch goals to provide job recommendations based on inputting skill keywords directly from the UI as well as provide country specific recommendations (United

States and Canada). We weren’t unable to complete these stretch goals due to bandwidth issues.