DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Major Project Phase-II Report

Job Recommendation System

By

Atharva Patil [ENG19CS0052]
Aryan Kumar [ENG19CS0044]
Divyansh Suwalka [ENG19CS0089]
Gaurav Rai [ENG19CS0097]

Under the supervision of

Dr. Jayita Saha
Assistant Professor,
Dept. Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,
BANGALORE

(2022-2023)

DAYANANDA SAGAR UNIVERSITY

School of Engineering

Department of Computer Science & Engineering

Kudlu Gate, Bangalore – 560068

Karnataka, India

CERTIFICATE

This is to certify that the Major Project Stage-II work titled **"JOB RECOMMENDATION SYSTEM"** is carried out by **Atharva Patil (ENG19CS0052), Aryan Kumar (ENG19CS0044), Divyansh Suwalka (ENG19CS0089) and Gaurav Rai (ENG19CS0097),** a bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2022-2023**.

Dr Jayita Saha	**Dr Girisha G S**	**Dr. Udaya Kumar Reddy K**
Assistant Professor Dept. of CS&E, School of Engineering Dayananda Sagar University	Chairman CSE School of Engineering Dayananda Sagar University	Dean School of Engineering Dayananda Sagar University
Date:	Date:	Date:

Name of the Examiner **Signature of Examiner**

1.

2.

DECLARATION

We, **Atharva Patil (ENG19CS0052), Aryan Kumar (ENG19CS0044), Divyansh Suwalka (ENG19CS0089) and Gaurav Rai (ENG19CS0097),** are students of eighth semester B.Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the Major Project Stage-II titled **"JOB RECOMMENDATION SYSTEM"** has been carried out by us and submitted in partial fulfillment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2022-2023**.

Student	**Signature**

Name 1: Atharva Patil
USN : ENG19CS0052

Name 2: Aryan Kumar
USN : ENG19CS0044

Name 3: Divyansh Suwalka
USN : ENG19CS0089

Name 4: Gaurav Rai
USN : ENG19CS0097

Place : Bangalore
Date :

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to the School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our bachelor's degree in this institution.

We would like to thank ***Dr. Udaya Kumar Reddy K R, Dean, School of Engineering & Technology, Dayananda Sagar University*** *for his constant encouragement and expert advice.*

It is a matter of immense pleasure to express our sincere thanks to ***Dr. Girisha G S, Department Chairman****,* ***Computer Science and Engineering****,* ***Dayananda Sagar University,*** *for providing the right academic guidance that made our task possible.*

We would like to thank our guide ***Dr Jayita Saha****,* ***Assistant Professor****,* ***Dept. of Computer Science and Engineering****,* ***Dayananda Sagar University****, for sparing his/her valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.*

We would like to thank our ***Project Coordinators Dr. Meenakshi Malhotra and Dr. Pramod Naik, Associate Professors, Department of Computer Science and Engineering*** *and all the staff members of Computer Science and Engineering for their support.*

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

	Page
LIST OF ABBREVIATIONS	vi
LIST OF FIGURES	vii
LIST OF TABLES	viii
ABSTRACT	ix
CHAPTER 1 INTRODUCTION	1
1.1. INTRODUCTION	2
1.2. WHY DO WE USE RECOMMENDER SYSTEM	6
1.3. OBJECTIVE	6
1.4. SCOPE	7
CHAPTER 2 PROBLEM DEFINITION	8
CHAPTER 3 LITERATURE SURVEY	11
CHAPTER 4 PROJECT DESCRIPTION	19
CHAPTER 5 REQUIREMENTS	20
5.1. FUNCTIONAL REQUIREMENTS	21
5.2. NON-FUNCTIONAL REQUIREMENTS	21
5.3. HARDWARE AND SOFTWARE REQUIREMENTS	22
CHAPTER 6 METHODOLOGY	23
6.1. CONTENT BASED FILTERING	24
6.2. DATASED USED	25
CHAPTER 7 EXPERIMENTATION	27
CHAPTER 8 TESTING AND RESULTS	31
8.1 TESTING	32
8.2 TEST CASES	37
8.3 RESULTS	39
8.4 DISCUSSION OF RESULTS	41
CHAPTER 9 CONCLUSION	42
9.1 CONCLUSION	43
9.2 FUTURE SCOPE	44
CHAPTER 10 REFERENCES	45
Funding and Published Paper details	48

LIST OF ABBREVIATIONS

RS	Recommender System
CF	Collaborative Filtering
CBF	Content Based Filtering
KB	Knowledge Based
SVD	Singular Value Decomposition

LIST OF FIGURES

Fig. No.	Description of the figure	Page No.
1	Collaborative Based Filtering	3
2	Content Based Filtering	4
3	System Design	19
4	Methodology Flow Chart	24
5	Code Snippets 1	32
6	Code Snippets 2	33
7	Cosine Similarity	33
8	Filtering Keywords	34
9	Code Snippet 4	34
10	Code Snippet 5	35
11	Output Snippet	35
12	Data Set Used	36
13	Testing For Network Engineer Job	37
14	Domain Not Mentioned or Specified	37
15	Skillset is not mentioned	38
16	Input given as Cloud Computing	38
17	Input given as Mobile Developer	39
18	UI Snippet 1	39
19	UI Snippet 2	40
20	UI Snippet 3	40

LIST OF TABLES

Table No.	Description of the Table	Page No.
1	Literature Review	17

ABSTRACT

Job recommendation systems are AI-based platforms that use techniques such as deep learning, reinforcement learning, and knowledge graphs to provide personalized job recommendations to job seekers based on their skills, experience, and preferences. These systems face several challenges, including sparse data, data privacy and security, bias and discrimination, and transparency and interpretability. The techniques used in job recommendation systems help to overcome these challenges and improve the job search experience for job seekers by reducing the time taken to find suitable job opportunities and increasing job satisfaction.

The study presents a comprehensive survey of various filtering, machine learning, and deep learning techniques used in job recommendation systems. It also discusses the applications and challenges associated with existing job recommendation systems. By addressing the challenges such as sparse data, improving the accuracy of recommendations, and ensuring transparency and interpretability, the job recommender architecture serves as a mediator to provide a more efficient and satisfactory job search experience for job seekers.

CHAPTER 1

INTRODUCTION

CHAPTER 1 INTRODUCTION

1.1. INTRODUCTION

Joblessness is a significant issue that affects individuals, families, and entire communities. It can lead to financial instability and poverty, as well as mental health problems and a sense of hopelessness. The current economic climate has made the problem worse, with many businesses shutting down and job opportunities becoming scarce. Governments around the world are grappling with how to address this issue and provide support for those who are out of work. It's crucial that a long-term solution is found to combat joblessness and help people get back on their feet.

Irrelevant job postings can be a major problem for job seekers, as they can waste a lot of time and energy searching for positions that they are not qualified for or do not meet their needs. By filtering out these irrelevant job postings and focusing on positions that are a good fit for the individual's qualifications and preferences, job seekers can save time and increase their chances of finding a suitable job. Additionally, this can also help to reduce the number of non-applicable job postings and make the job searching process more efficient for everyone. By offering tailored job recommendations to job seekers based on their abilities, geography, and other preferences, a recommender system can be an effective tool in the recruiting process. The user's qualifications and preferred jobs are gathered through this process, the recommender system can match them with job openings that are most relevant to them. This can save job seekers time and effort in their job search, as they will only be presented with positions that are a good fit for them. Additionally, by using a resume as input, the system can quickly and easily assess the user's qualifications and make recommendations without the need for additional forms or applications.

1.1.1. Types Of Recommender System

These are the systems that help us to select out similar things whenever we select something online. The concept of understanding a user's preference by their online behavior, previous purchases, or history in the system is called a recommender system. The need for a recommender system has grown from time to time. At First, Entertainment industries exploited the benefits of these systems. Then recommender systems were implemented in e-shopping businesses, online news, but very few companies have tried implementing it in the hiring process.

A. Collaborative Based Filtering:

Collaborative filtering is a technique used in recommendation systems to predict the interests of a user by collecting preferences or taste information from many users. The idea behind this approach is that people tend to have similar preferences and behaviors to others with similar characteristics, so by analyzing the behavior of similar users, we can predict the preferences of a particular user.

Collaborative filtering algorithms use a variety of techniques to find similar users or items, such as cosine similarity or Euclidean distance, and then make recommendations based on the preferences of those similar users or items. This technique is widely used in many industries, including e-commerce, social media, and music and movie recommendation systems.

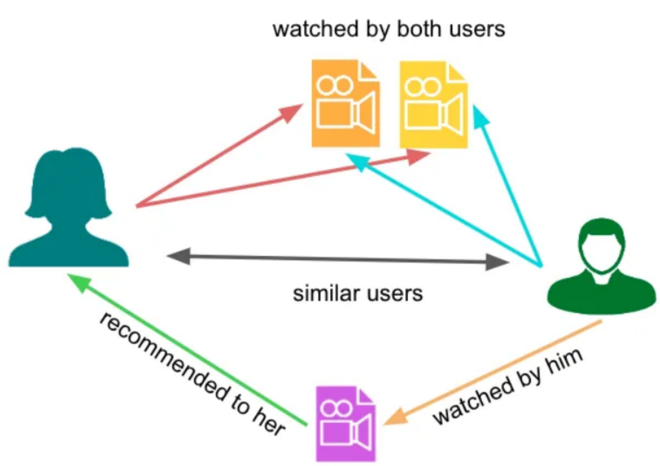

Fig 1: Collaborative Based Filtering

B. Content-Based Recommendation:

Content-based recommendation is a technique used in recommendation systems to make personalized recommendations to users based on their previous interactions and interests. This approach relies on analyzing the features or characteristics of the items being recommended and matching them to the user's past behavior and preferences.

The main advantage of content-based recommendation systems is that they do not require large amounts of data on other users' preferences or behaviors, making them more suitable for recommending niche or specialized products. Moreover, content-based recommendation can also provide an explanation for why a particular item was recommended, which can increase the user's trust in the system.

The content-based approach is typically used for recommending products or items that have clear and distinct features, such as movies, books, or music. The features of the items are usually described using metadata, such as genre, director, actors, author, or keywords. The system then creates a profile of the user's preferences based on their interactions with items and uses that profile to match the user with new items that have similar features.

However, the content-based approach also has some limitations, such as the difficulty of identifying the most relevant features of the items, and the tendency to recommend similar items over and over again, which can lead to a lack of diversity in the recommendations.

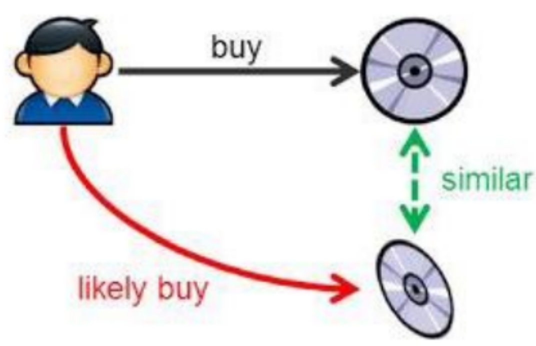

Fig 2: Content Based Filtering

C. Knowledge based recommendation system:

A knowledge-based recommendation system is a type of recommendation system that uses knowledge representation and reasoning techniques to make recommendations. This approach relies on a knowledge base that contains information about the items being recommended, as well as the user's preferences, requirements, and constraints. The knowledge base is created manually or using automatic techniques such as web scraping, text mining, or machine learning.

In a knowledge-based recommendation system, the system typically asks the user to provide input about their preferences, requirements, or constraints. Based on this input, the system searches the knowledge base to identify items that meet the user's criteria and then recommends those items to the user.

The main advantage of knowledge-based recommendation systems is their ability to make highly personalized and transparent recommendations that take into account the user's preferences, requirements, and constraints. They can also provide explanations for why a particular item was recommended, which can increase the user's trust in the system.

However, the main limitation of knowledge-based recommendation systems is their reliance on a high-quality and up-to-date knowledge base, which can be difficult and expensive to create and maintain. They also suffer from the cold start problem, which occurs when there is not enough information about the user's preferences or when a new item is added to the system.

1.2 WHY DO WE USE RECOMMENDER SYSTEM

Industries try finding ways to increase their revenue. In a classic business model, the up-selling is the term used when a sales advisor tries to sell an item to a customer based on things that he is planning to purchase. As businesses started to integrate the technology to increase the user interaction with business, customers gave out their preferences by interacting or purchasing the product the business is selling to them. The business has utilized the collected big data to make a better-personalized recommendation to its customer base. Netflix is a streaming service that generates its revenue from customer subscriptions to its content. According to reports from business insider Australia, Netflix estimates 20% of their video watches are derived from the search. Whereas 80% comes from its recommendation system.

1.3 OBJECTIVE

- To develop a content-based job recommendation system using machine learning algorithms that can analyze job descriptions and user profiles to suggest relevant job openings based on the users' skills and preferences.
- To address the challenges faced by job seekers in the current job market by providing personalized job recommendations that increase their chances of finding suitable employment.
- To evaluate the system's performance and effectiveness by conducting experiments and analyzing the results.
- To identify the strengths and limitations of the approach and provide insights for future research and development in this area.
- To ensure that the job recommendation system is user-friendly and easy to navigate, with a clear and intuitive interface.
- To explore the potential of incorporating additional data sources, such as social media profiles or online job application history, to further improve the accuracy and relevance of the job recommendations.

1.4 SCOPE

The scope of this project is quite broad, as it encompasses various aspects of the job market and recruitment process. The primary objective of the project is to develop a system that can provide job seekers with personalized job recommendations based on their skills, qualifications, and preferences. The scope of the job recommendation system project is not limited to just these objectives. Depending on the project's goals and objectives, the scope can be expanded to include additional features, such as integrating with social media platforms, providing real-time job recommendations, and incorporating feedback from job seekers and employers.

CHAPTER 2

PROBLEM DEFINITION

CHAPTER 2 PROBLEM DEFINITION

The job market is becoming increasingly competitive, and job seekers often struggle to find suitable job opportunities that match their skills, experience, and preferences. Job recommendation systems have emerged as a solution to this problem by providing personalized job recommendations to job seekers. However, these systems face several challenges that need to be addressed to ensure that the recommendations are accurate, unbiased, and understandable to job seekers.

One of the primary challenges facing job recommendation systems is sparse data. This occurs when there is not enough data available on a job seeker to make accurate and relevant job recommendations. This can be due to various reasons such as lack of job history, limited educational background, or limited personal information. To address this challenge, job recommendation systems need to use techniques such as collaborative filtering, content-based filtering, and hybrid filtering to analyze the available data and make accurate and relevant job recommendations.

Another major challenge facing job recommendation systems is bias and discrimination. The recommendations made by the system can be influenced by factors such as gender, race, or ethnicity, resulting in unfair or discriminatory recommendations. To address this challenge, job recommendation systems need to use techniques such as fairness-aware machine learning and explainable AI to ensure that the recommendations are fair, unbiased, and transparent.

Data privacy and security is another important challenge facing job recommendation systems. Job seekers are often hesitant to share their personal information with job recommendation systems due to concerns about privacy and security. To address this challenge, job recommendation systems need to use techniques such as differential privacy and secure multi-party computation to ensure that the data is secure and the privacy of job seekers is protected.

In summary, job recommendation systems have the potential to revolutionize the job search experience for job seekers. However, they face several challenges such as sparse data, bias and discrimination, and data privacy and security. By addressing these challenges through the use of various techniques such as collaborative filtering, fairness-aware machine learning, and differential privacy, job recommendation systems can provide accurate, unbiased, and understandable job recommendations that benefit job seekers.

CHAPTER 3

LITERATURE REVIEW

CHAPTER 3 LITERATURE REVIEW

In this work [11], by H. Jain and M. Kakkar provided a comprehensive overview of existing job recommendation systems and the techniques used in these systems. The authors highlighted the importance of job recommendation systems in reducing the time and effort required for job seekers to find suitable job opportunities.

The review covered several job recommendation systems such as LinkedIn, Indeed, and Glassdoor, and discussed the features and techniques used in these systems. The authors noted that these systems are effective in providing job recommendations but have limitations such as lack of transparency, bias, and limited personalization. This highlights the need for better job recommendation systems that can provide more accurate and personalized job recommendations.

The authors also reviewed several studies that have addressed the challenges facing job recommendation systems. They highlighted the use of deep learning techniques such as convolutional neural networks (CNN) and recurrent neural networks (RNN) to improve the accuracy of job recommendations. The authors noted that these techniques have shown promising results and have the potential to improve the accuracy of job recommendations.

The review also highlighted the importance of addressing issues related to bias and discrimination in job recommendation systems. The authors noted that job recommendation systems can be influenced by factors such as gender, race, or ethnicity, resulting in unfair or discriminatory recommendations. They highlighted the need for fairness-aware machine learning and explainable AI to ensure that the recommendations are fair, unbiased, and transparent.

Overall, the literature review section of the paper provided valuable insights into existing job recommendation systems and the challenges faced by these systems. The authors identified the limitations of existing systems and highlighted the need for more accurate, personalized, and fair job recommendation systems. The review also highlighted the potential of deep learning techniques such as CNN and RNN to improve the accuracy of job recommendations.

The paper [12] titled "A Machine Learning Approach for Automation of Resume Recommendation System" by Roy et al. presented a resume recommendation system that used machine learning techniques to match job seekers with relevant job openings based on their skills and experience. The paper included a literature review section that provided an overview of existing resume recommendation systems and the techniques used in these systems.

The authors reviewed several resume recommendation systems, including LinkedIn, Indeed, and Monster. They highlighted the features and techniques used in these systems, such as natural language processing, machine learning algorithms, and user feedback. The authors noted that these systems are effective in providing relevant job recommendations but have limitations such as lack of transparency and interpretability.

The authors also reviewed several studies that have addressed the challenges facing resume recommendation systems. They highlighted the importance of addressing the issue of bias and discrimination in these systems. The authors noted that resume recommendation systems can be influenced by factors such as gender, race, or ethnicity, resulting in unfair or discriminatory recommendations. They suggested that fairness-aware machine learning techniques can be used to address this issue.

The literature review section also discussed the use of machine learning techniques such as deep learning, decision trees, and random forests in resume recommendation systems. The authors noted that these techniques can improve the accuracy of recommendations and reduce the time and effort required to match job seekers with relevant job openings.

The authors highlighted the importance of data privacy and security in resume recommendation systems. They noted that the system should ensure that the personal data Job seekers are protected and not misused. The authors also suggested that blockchain technology can be used to ensure data privacy and security.

Overall, the literature review section of the paper provided a comprehensive overview of existing resume recommendation systems and the challenges faced by these systems. The authors identified the limitations of existing systems and highlighted the need for more accurate, transparent, and fair resume recommendation systems. They also discussed the potential of machine learning techniques such as deep learning, decision trees, and random forests to improve the accuracy of recommendations. The authors suggested that the use of blockchain technology can ensure data privacy and security in these systems.

The paper [13] titled "Combining content-based and collaborative filtering for job recommendation system" by Shou Yang et al. presented a job recommendation system that combined content-based and collaborative filtering techniques. The paper included a literature review section that provided an overview of existing job recommendation systems and the techniques used in these systems.

The authors reviewed several job recommendation systems, including LinkedIn, Indeed, and Glassdoor. They highlighted the features and techniques used in these systems, such as collaborative filtering, content-based filtering, and hybrid filtering. The authors noted that these systems are effective in providing relevant job recommendations but have limitations such as the cold start problem, sparsity of data, and the scalability of the system.

The literature review section also discussed the use of content-based filtering and collaborative filtering techniques in job recommendation systems. The authors noted that content-based filtering techniques use the attributes of the job seeker and job descriptions to recommend relevant job openings, while collaborative filtering techniques use the ratings or preferences of other users to recommend job openings. The authors suggested that combining these techniques can improve the accuracy of job recommendations.

The authors highlighted the importance of overcoming the cold start problem in job recommendation systems. The cold start problem refers to the difficulty in providing job recommendations for new users or job openings with limited data. The authors suggested that content-based filtering can be used to address this issue by recommending jobs based on the attributes of the job seeker or job description.

The authors also discussed the importance of addressing the issue of sparsity of data in job recommendation systems. The sparsity of data refers to the fact that job seekers may not have a complete profile or may not have rated many jobs. The authors suggested that collaborative filtering can be used to address this issue by using the preferences of other users with similar profiles to recommend jobs.

Overall, the literature review section of the paper provided a comprehensive overview of existing job recommendation systems and the challenges faced by these systems. The authors identified the limitations of existing systems and highlighted the need for more accurate, scalable, and efficient job recommendation systems. They also discussed the potential of combining content-based and collaborative filtering techniques to improve the accuracy of job recommendations and to address the cold start problem and sparsity of data issues.

The paper[14] titled "Job Recommendation System, Machine Learning, Regression, Classification, Natural Language Processing" by K. Appadoo et al. presented a job recommendation system that utilized machine learning, regression, classification, and natural language processing techniques. The paper included a literature review section that provided an overview of existing job recommendation systems and the techniques used in these systems.

The authors reviewed several job recommendation systems, including LinkedIn, Indeed, and Glassdoor. They highlighted the features and techniques used in these systems, such as collaborative filtering, content-based filtering, hybrid filtering, and deep learning. The authors noted that these systems are effective in providing relevant job recommendations but have limitations such as the cold start problem, sparsity of data, and the scalability of the system.

The literature review section also discussed the use of natural language processing techniques in job recommendation systems. The authors noted that natural language processing can be used to analyze job descriptions and match them with the skills and experience of job seekers. The authors highlighted the importance of extracting relevant information from job descriptions, such as job title, location, industry, and required skills, to improve the accuracy of job recommendations.

The authors also discussed the importance of addressing the issue of bias and discrimination in job recommendation systems. The authors noted that machine learning algorithms can be biased if the training data is biased. They highlighted the need to ensure that the training data is diverse and representative of the population to avoid bias and discrimination.

The authors proposed a job recommendation system that utilized machine learning techniques such as regression and classification to match job seekers with relevant job openings based on their skills, experience, and preferences. They also used natural language processing techniques to extract relevant information from job descriptions and match them with the skills and experience of job seekers. The authors noted that their system addressed the cold start problem and sparsity of data issues by utilizing regression and classification techniques.

Overall, the literature review section of the paper provided a comprehensive overview of existing job recommendation systems and the challenges faced by these systems. The authors identified the limitations of existing systems and highlighted the need for more accurate, scalable, and efficient job recommendation systems. They also discussed the potential of using natural language processing techniques to improve the accuracy of job recommendations and to address the issue of bias and discrimination in job recommendation systems. The authors proposed a job recommendation system that utilized machine learning, regression, classification, and natural language processing techniques to address the challenges faced by existing job recommendation systems.

Table 1: Literature Review

Reference and Year	Methodology Considered	Remarks
[11], 2019	RestFul API, Android IDE	High Database Latency
[12], 2020	Machine Learning,Automation	Less Accuracy at 78.53%
[13], 2017	Content Based Filtering, Collaborative Filtering	Cold Start Problem
[14], 2020	Natural Language Processing, Regression, Classification	Combination of NLP and CFR

CHAPTER 4

PROJECT DESCRIPTION

CHAPTER 4 PROJECT DESCRIPTION

With the appearance of joblessness expanded in the present situation, there must be an appropriate framework for job aspiring candidates. People are getting too many options which make them difficult to differentiate among various jobs. This leads to information overload. To lighten this issue, a job recommendation system is presented which fundamentally filters the jobs using the candidate's profile. Recommender system is nothing but a decision-making tool that recommends products based on user's preferences or interests. It acts as an information retrieval tool that helps to filter out and prioritize the data. The framework at that point suggests the job seekers with proper jobs that are appropriate for them and matches their profile as needs be. Job recommender framework subsequently goes about as a middle person between the job aspiring candidates and recruiters.

As a result, the developed recommendation system will take in users' resumes as an input then run the NLP scripts on it, extract important information from it such as the name, phone number and email and skillset.

This then would generate a feature vector which will be given to the recommendation engine which will suggest the jobs suited to that candidate.

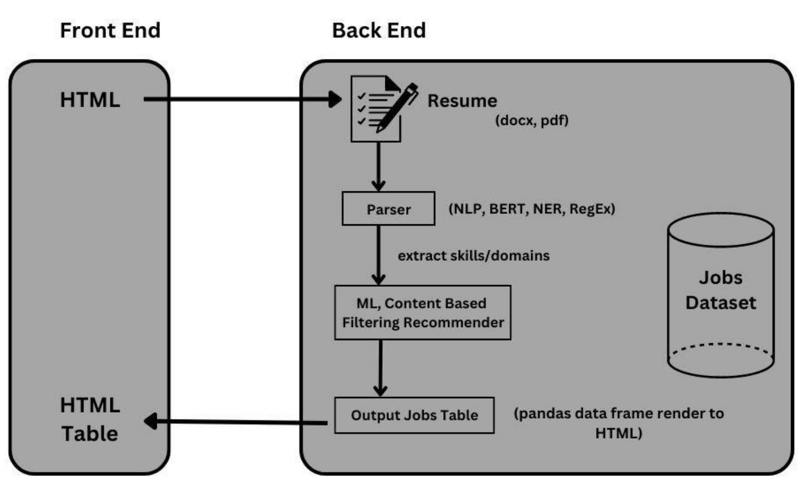

Fig 3: System Design

CHAPTER 5

REQUIREMENTS

CHAPTER 5 REQUIREMENTS

5.1 Functional Requirements:

- We calculate the matching of the applicant's profile with the job listing.
- We use NLTK and Spacy for doing the work of Information Extraction.
- We extract all required information of the candidate automatically from their respective resume.
- **spaCy** is an open-source software library for advanced natural language processing, written in the programming language Python.
- **NLTK,** is a suite of libraries stical natural language processing for English written in the Python programming language.

5.2 Non-Functional Requirements:

- **Usability Requirements:** The application shall be user friendly and doesn't require any guidance. In other words, the application will be easy to learn and use.
- **Reliability Requirements:** The application should not have any unexpected failure. To avoid any failure occurrence, the specifications have been respected and followed correctly. Any component can be modified to correct faults, improve performance or other attributes, or adapt to a changing environment, because of the development methodology followed.
- **Performance and Accuracy:** The accuracy achieved in this application would be after testing out multiple models and accessing the services of our DL models would be through significantly faster API requests from the browser to the server.
- **Correctness of Output:** The output of the system will match the expectations outlined by the user, as all the models would be trained on huge datasets and validated manually before deployment.

5.3 Hardware And Software Requirements

5.3.1. Hardware Requirements

5.3.1.1. Minimum 16 GB RAM

5.3.1.2. At Least 8 GB of available disk space.

5.3.2. Software Requirements

5.3.2.1. 64-bit Windows 8-11 / macOS 10.14 or higher / 64-bit Linux

5.3.2.2. Visual Studio Code or similar code editor.

5.3.2.3. Python3.7 and above

5.3.3. Bandwidth Requirements

5.3.3.1. Bandwidth: 2-5 Mbps

CHAPTER 6

METHODOLOGY

CHAPTER 6 METHODOLOGY

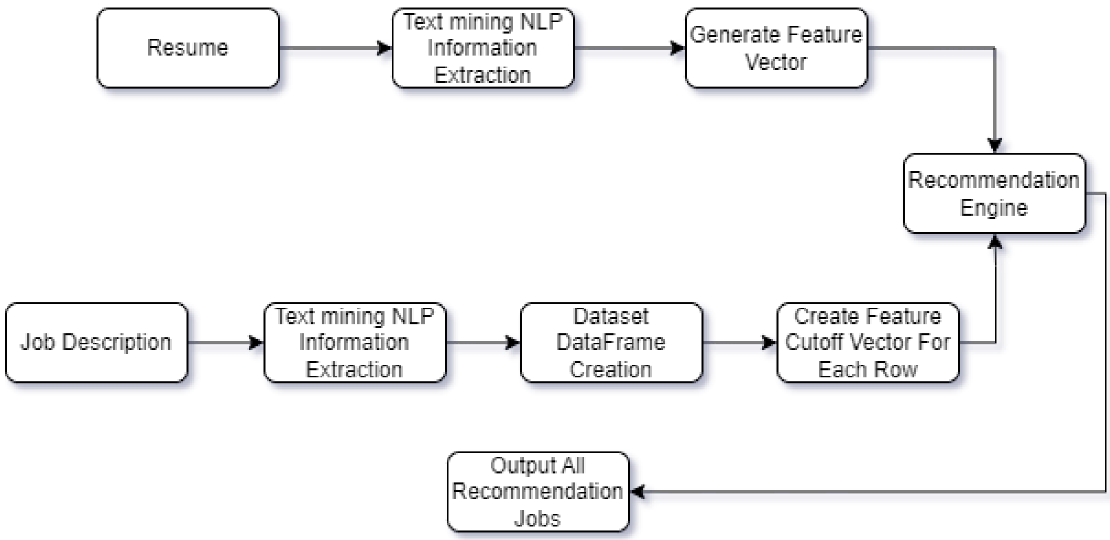

Fig 4: Methodology Flow Chart

6.1 Content Based Filtering

The collaborative filtering method uses the history of the user's interaction with an item as the input for building the recommenders. In contrast, Content or attribute-based recommenders use specific properties of an item or the user along with the previous user interaction with the item to recommend the user with the relevant item. As we have collected two data sets, where one is a user data set and another is a dataset of jobs extracted from the web, there is no previous interaction between the user and item. So during the implementation, we will be generating a user profile and the item profile based on the dataset considering the explicit attributes of user and item. Content-based recommenders operate with the assumption that items having related attributes have similar interest at the user level. Content-based recommendation engines keep suggesting an item to a user similar to the items rated highly by the same user. The user will most likely get recommendations about jobs with the same skill or Job domain as the user preferred in the past. In most scenarios, this behavior of the recommending system is considered as it is saturated. However, in the job domain, user preference remains the same in the system. As the user rarely changes his preference to change his job domain or work on different skill sets and also the previously recommended job would not be in the system as job listings will expire when the role is fulfilled.

6.2 Dataset Used

The pre-crawled dataset available on Kaggle is a collection of job postings in the IT sector from the US-based job board, Dice.com. The dataset contains information about various IT jobs available in the US, including the job title, company, work location, job description, and type of employment.

The dataset has been created by PromptCloud's in-house web-crawling service, which extracted data from Dice.com. It is a valuable resource for job seekers, researchers, and companies looking to understand the job market trends in the IT industry.

The dataset contains over 22,000 job listings, spanning across different IT roles such as software engineer, data scientist, network administrator, and many more. The job titles in the dataset range from entry-level to senior positions, offering a comprehensive view of the IT job market.

In addition to job titles, the dataset also provides information on the companies offering the positions. The company information includes the company name, location, and industry sector. This can be useful for job seekers who are looking to target specific companies or industries.

The dataset also includes a detailed job description for each listing, which provides insights into the specific responsibilities, qualifications, and skills required for the job. This can be useful for job seekers to understand the requirements for different IT roles and prepare themselves accordingly.

Finally, the dataset also includes information on the type of employment offered, such as full-time, part-time, or contract. This can be useful for job seekers who are looking for specific types of employment.

Overall, the pre-crawled dataset available on Kaggle is a valuable resource for anyone interested in the IT job market in the US. It provides comprehensive information on different job roles, companies, and job requirements, making it a valuable tool for job seekers, researchers, and companies.

CHAPTER 7

EXPERIMENTATION

CHAPTER 7 EXPERIMENTATION

Data collection: The first step is to collect job-related data from various sources, including job postings, resumes, user profiles, and job-related websites. This data will be used to train the machine learning algorithms that power the recommendation system.

Data preprocessing: Once the data is collected, it needs to be preprocessed to extract useful features that can be used to make job recommendations. This step involves text normalization, feature selection, and feature extraction.

Model development: The next step is to develop machine learning models that can learn from the preprocessed data and make job recommendations based on the user's preferences and qualifications. This step involves the selection of appropriate algorithms and fine-tuning the model parameters.

Evaluation: The developed model needs to be evaluated to assess its performance and effectiveness in making job recommendations. This step involves testing the model on a large dataset, and measuring its accuracy, precision, recall, and F1 score.

Optimization: Based on the evaluation results, the model needs to be optimized to improve its performance. This step involves tweaking the model parameters and exploring alternative algorithms to improve the recommendation system's accuracy and effectiveness.

Deployment: Finally, the optimized model is deployed in a web-based application, which can be used by job seekers to find suitable job openings based on their qualifications and preferences.

A job recommendation system is a useful tool that provides job seekers with personalized job recommendations based on their interests, skills, and previous work experience. The process of building an effective recommendation system involves several steps, including data collection, preprocessing, and experimentation.

Data collection is a crucial step in building an effective recommendation system. In this case, a large dataset of job opportunities and user profiles is collected from various sources, including job listing websites and user profiles. The data collected may include information in PDF and DOCX format, which must be preprocessed to extract useful information.

Preprocessing involves converting the text in PDF and DOCX files into a string format, which is then analyzed using regular expressions. Regular expressions are powerful tools for extracting specific information from unstructured text. In this case, they help retrieve information such as email and phone numbers from the text. Additionally, POS tagging is used to analyze the preprocessed text to retrieve user names.

Named entity recognition (NER) is another useful technique used to identify entities in text, including job titles, companies, and locations. BERT, a powerful Transformer-based NLP model, is used for this purpose. BERT has been shown to achieve state-of-the-art results on several NER benchmarks, making it a useful tool for various NER tasks.

Experimentation is a crucial step in building an effective job recommendation system. Various machine learning algorithms, including collaborative filtering and content-based filtering, can be used to generate recommendations. Collaborative filtering is a technique that works by finding similarities between users' interests and recommending jobs based on those similarities. On the other hand, content-based filtering works by analyzing the job descriptions and recommending jobs that match the user's skills and preferences.

Apart from the technical aspects, the job recommendation system can also help improve diversity and inclusion in the workplace. By recommending jobs to users based on their skills and interests, the system can help promote diversity in the workplace by attracting a more diverse pool of candidates.

Additionally, the system can help bridge the gap between employers and job seekers. Employers can use the system to find candidates with the necessary skills and qualifications for specific job positions, while job seekers can use the system to find jobs that match their skills and preferences.

In conclusion, the process of building an effective job recommendation system involves several steps, including data collection, preprocessing, and experimentation. Preprocessing involves converting data into a suitable format, using techniques such as regular expressions, POS tagging, and NER to retrieve relevant information. The job recommendation system has the potential to promote diversity and inclusion in the workplace and help bridge the gap between employers and job seekers. As the job market becomes more competitive, the job recommendation system can provide a valuable tool for job seekers and employers alike.

CHAPTER 8

TESTING AND RESULTS

CHAPTER 8 TESTING AND RESULTS

8.1 TESTING

```
def categorize_jobs(self):
    # #Predefined categories
    #Compare similarities of word embeddings
    nlp=spacy.load('en_core_web_lg')
    job_id=self.df2.loc[:,'uniq_id'].tolist()[:self.training_range]
    job_titles=self.df2.loc[:,'jobtitle'].tolist()[:self.training_range]
    job_descriptions=self.df2.loc[:,'jobdescription'].tolist()[:self.training_range]
    final_cat=pd.DataFrame(index=job_id)
    #categories=['Network Engineer','Application Development','Big Data','Data Analyst','Software Developer','DevOps','Software Testi
    categories=['Network Engineer','Full stack','QA/Test Developer','Enterprise application','DevOps','Mobile Developer','Back End','I
    for category in categories:
        final_cat[category]=np.nan
    for job_t_d in list(zip(job_id,job_titles,job_descriptions)):
        id_job=job_t_d[0]
        job_i=job_t_d[1]
        job_d=job_t_d[2]
        job_title=nlp(job_i.lower())
        job_description=nlp(job_d.lower())
        match_cat_title=dict()
        match_cat_description=dict()
        for category in categories:
            word=nlp(category.lower())
            match_cat_title[category]=job_title.similarity(word)
            match_cat_description[category]=job_description.similarity(word)
        match_cat_title=sorted(match_cat_title.items(),key=lambda x:x[1],reverse=True)
        match_cat_description=sorted(match_cat_description.items(),key=lambda x:x[1],reverse=True)


        #a represents max
        #if(match_cat_title[0][1]>0.5 or match_cat_description[0][1]>0.5):
        a=match_cat_title[0]
        #print(a)
        match_cat_description=list(filter(lambda x: self.check_threshold(match_cat_title,x),match_cat_description))
        if(len(match_cat_description)!=0):
            print(match_cat_description)
            print(id_job)
            #b=match_cat_description[0]
            final_cat.loc[id_job,a[0]]=1
            match_cat_description.extend([(match_cat_title[0][0],1)])
            sum_proportion=sum([x[1] for x in match_cat_description])
            for ele in match_cat_description:
                final_cat.loc[id_job,ele[0]]=ele[1]/sum_proportion
        else:
            print(id_job)
            final_cat.loc[id_job,a[0]]=1
    return final_cat
```

Fig 5: Code Snippet 1

```python
def extract_skills(input_text):
    stop_words = set(nltk.corpus.stopwords.words('english'))
    word_tokens = nltk.tokenize.word_tokenize(input_text)

    # remove the stop words
    filtered_tokens = [w for w in word_tokens if w not in stop_words]

    # remove the punctuation
    filtered_tokens = [w for w in word_tokens if w.isalpha()]

    # generate bigrams and trigrams (such as artificial intelligence)
    bigrams_trigrams = list(map(' '.join, nltk.everygrams(filtered_tokens, 2, 3)))

    # we create a set to keep the results in.
    found_skills = set()

    # we search for each token in our skills database
    for token in filtered_tokens:
        if token.lower() in SKILLS_DB:
            found_skills.add(token)

    print(bigrams_trigrams)  #comment this before production

    for x in DOMAIN_DB:
        if x in bigrams_trigrams:
            domain_des=x

    # we search for each bigram and trigram in our skills database
    for ngram in bigrams_trigrams:
        if ngram.lower() in SKILLS_DB:
            found_skills.add(ngram)

    return found_skills,domain_des
```

Fig 6: Code Snippet 2

```python
def cosine_similarity(arr1,arr2):
    ans=1- spatial.distance.cosine(arr1,arr2)
    if(np.isnan(ans)):
        return 0
    else:
        return ans
```

Fig 7: Cosine Similarity

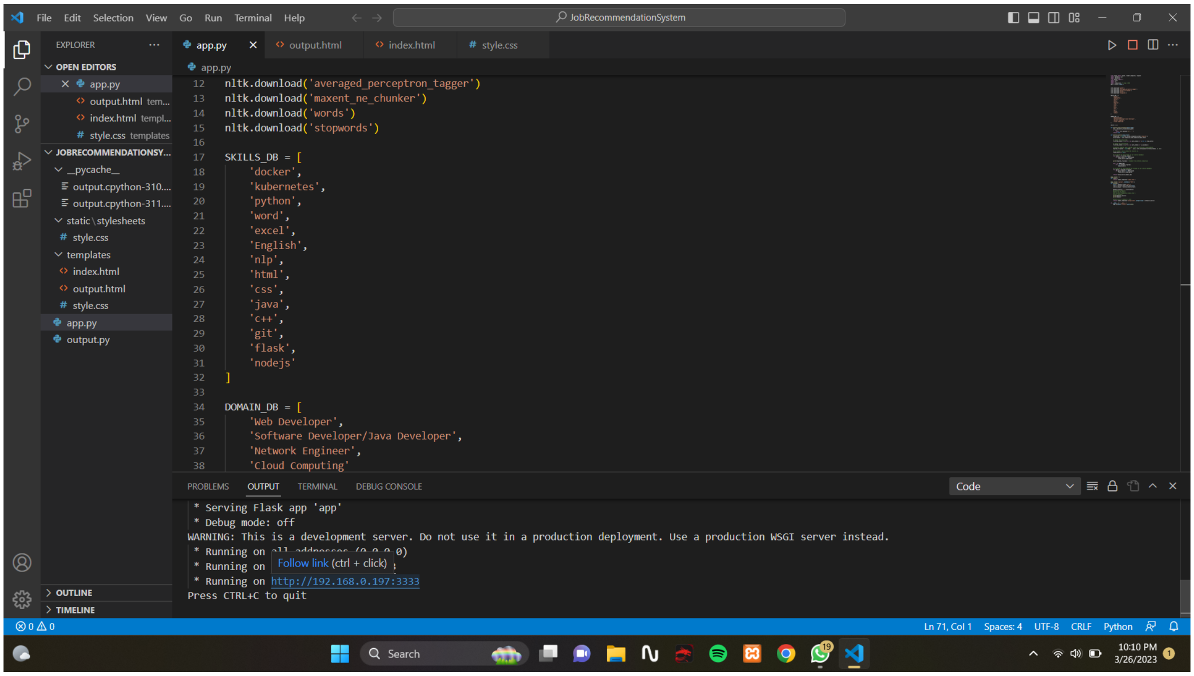

Fig 8: Filtering Keywords

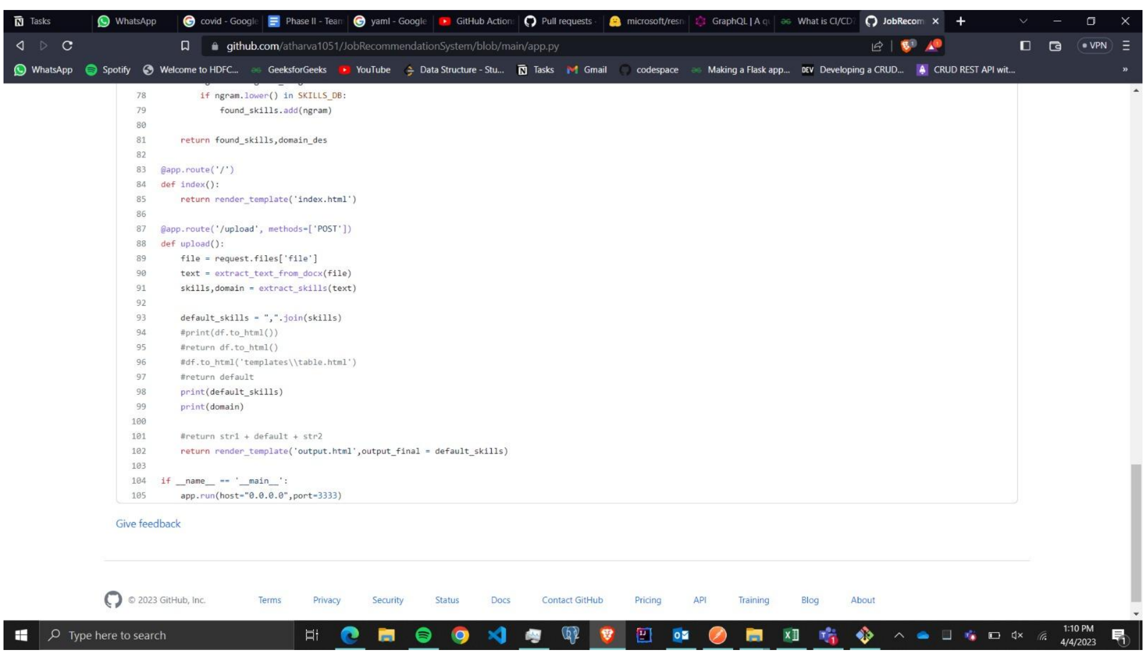

Fig 9: Code Snippet 4

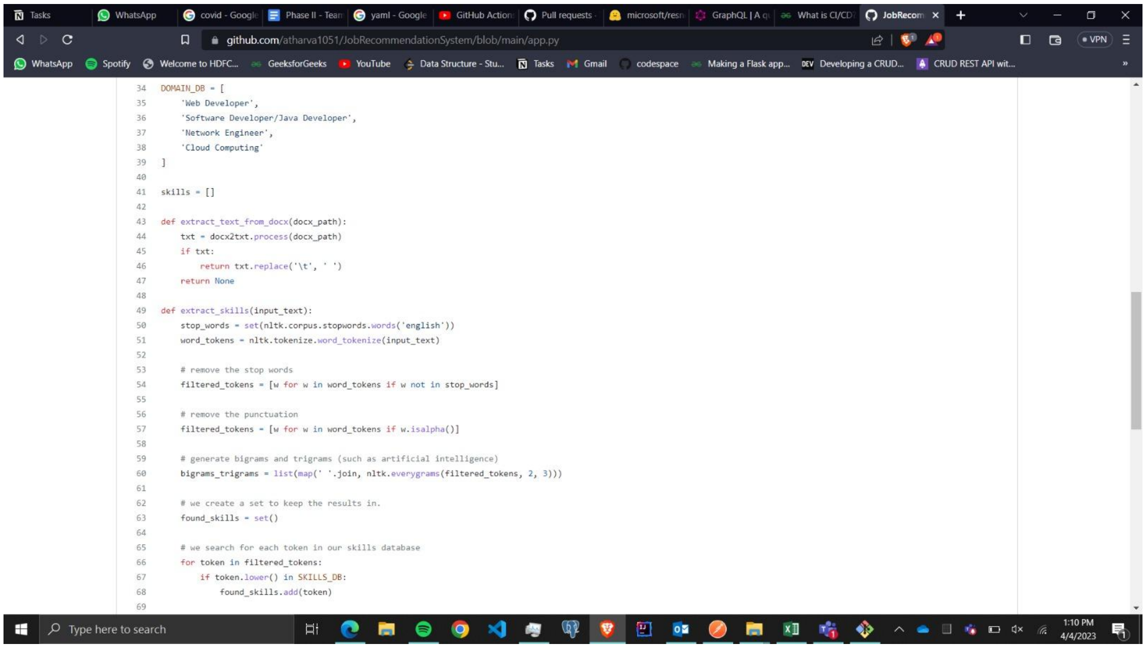

Fig 10: Code Snippet 5

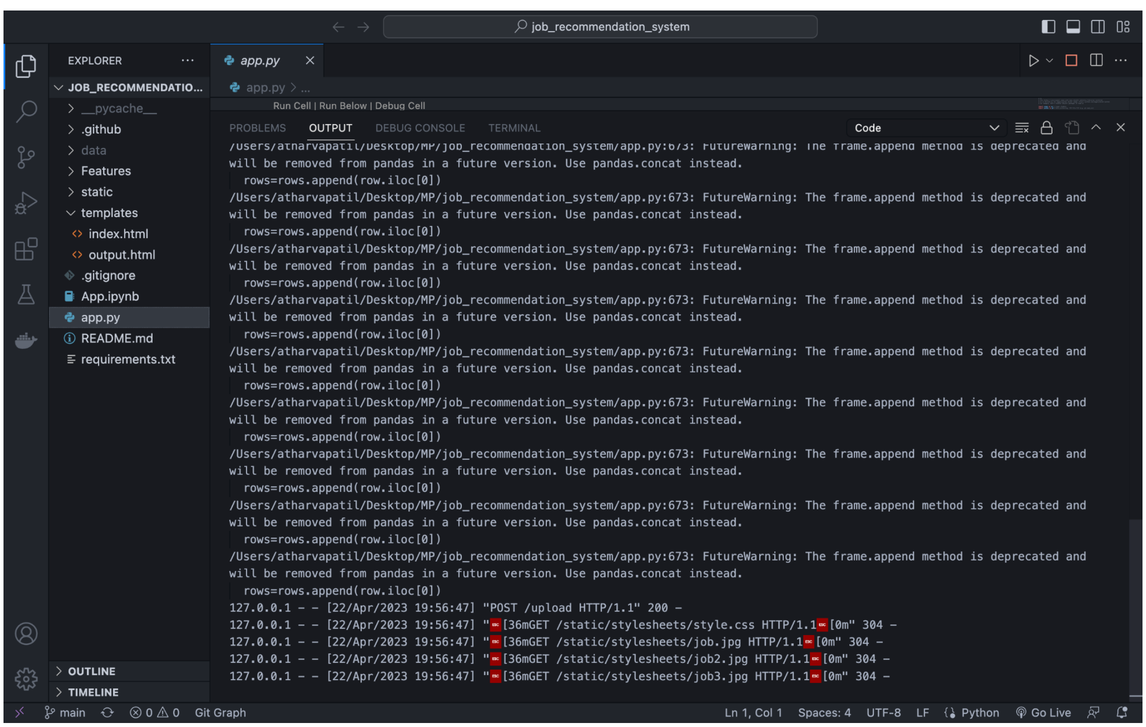

Fig 11: Output Snippet

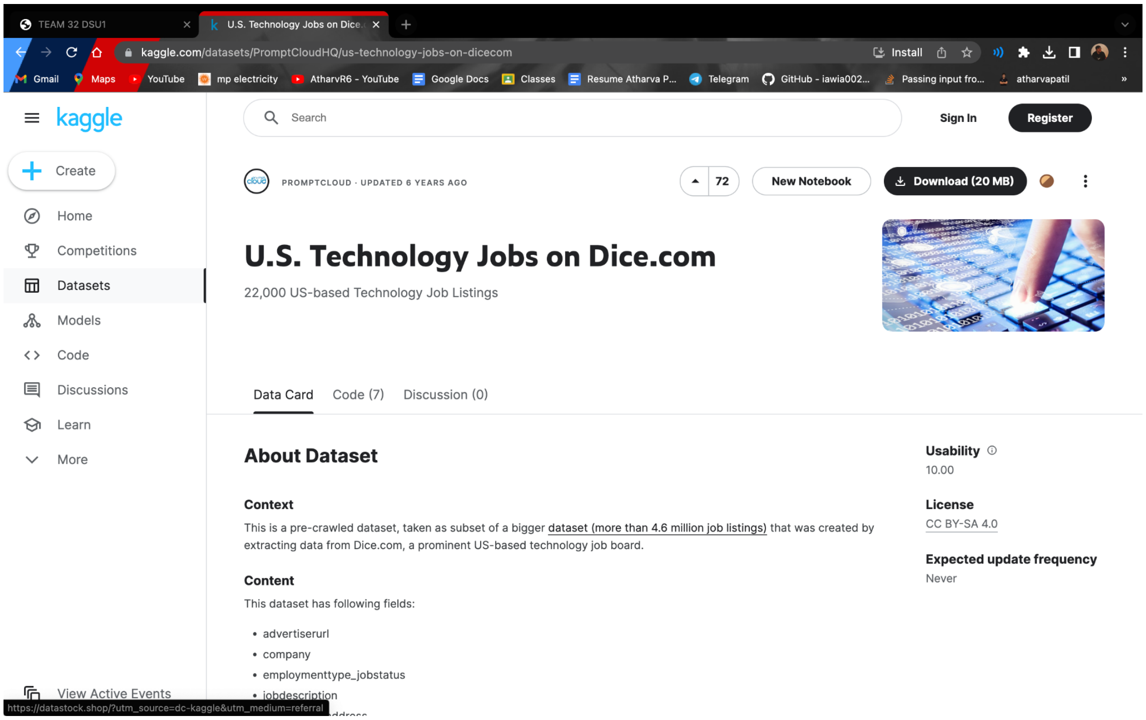

Fig 12: Data Set Used

8.2 TEST CASES

Fig 13: Testing For Network Engineer Job

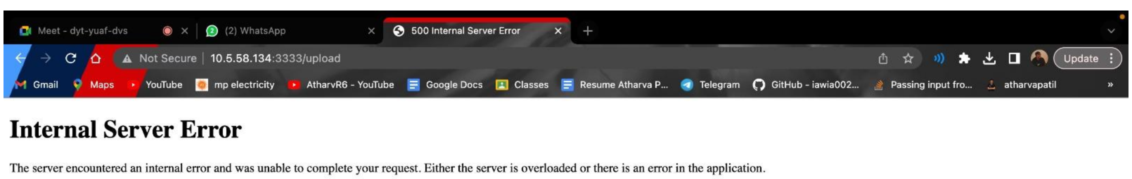

Fig 14: Domain Not Mentioned or Specified

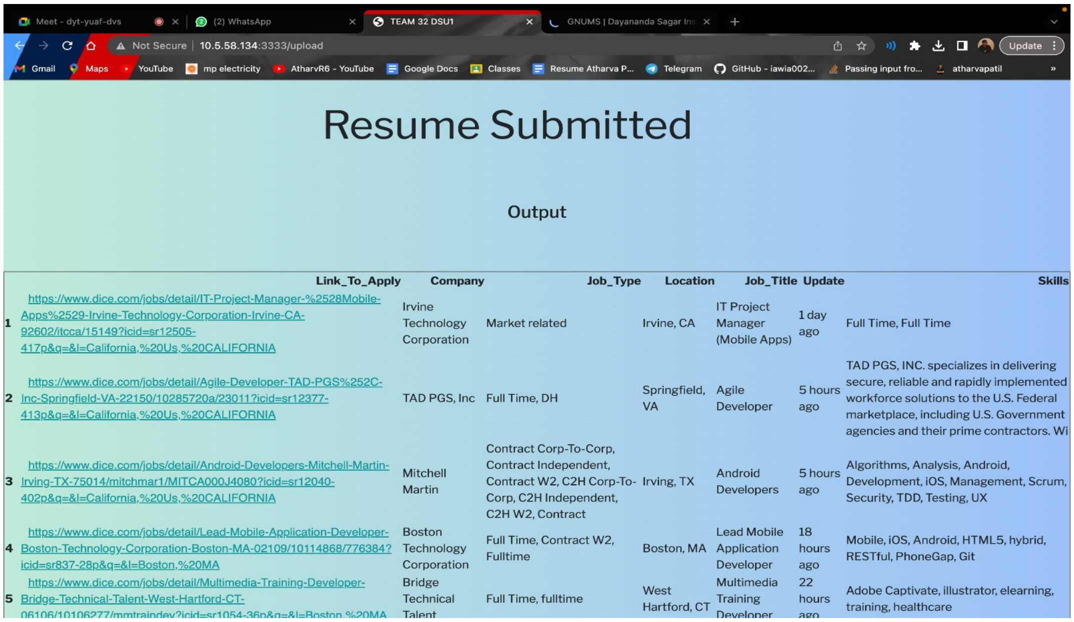

Fig 15: Skillset is not mentioned

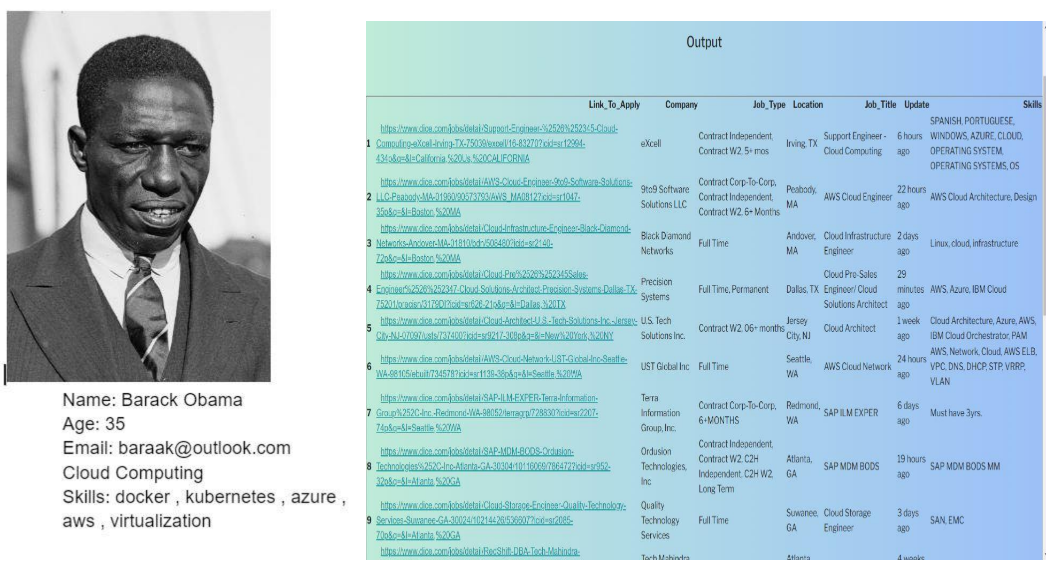

Fig 16: Input given as Cloud Computing

	Link_To_Apply	Company	Job_Type	Location	Job_Title	Update	Skills
1	https://www.dice.com/jobs/detail/IT-Project-Manager-%2528Mobile-Apps%2529-Irvine-Technology-Corporation-Irvine-CA-92602/itcca/15149?icid=sr12505-417p&q=&l=California,%20Us,%20CALIFORNIA	Irvine Technology Corporation	Market related	Irvine, CA	IT Project Manager (Mobile Apps)	1 day ago	Full Time, Full Time
2	https://www.dice.com/jobs/detail/Android-Developers-Mitchell-Martin-Irving-TX-75014/mitchmar1/MITCA000J4080?icid=sr12040-402p&q=&l=California,%20Us,%20CALIFORNIA	Mitchell Martin	Contract Corp-To-Corp, Contract Independent, Contract W2, C2H Corp-To-Corp, C2H Independent, C2H W2, Contract	Irving, TX	Android Developers	5 hours ago	Algorithms, Analysis, Android, Development, iOS, Management, Scrum, Security, TDD, Testing, UX
3	https://www.dice.com/jobs/detail/Lead-Mobile-Application-Developer-Boston-Technology-Corporation-Boston-MA-02109/10114868/776384?icid=sr837-28p&q=&l=Boston,%20MA	Boston Technology Corporation	Full Time, Contract W2, Fulltime	Boston, MA	Lead Mobile Application Developer	18 hours ago	Mobile, iOS, Android, HTML5, hybrid, RESTful, PhoneGap, Git
4	https://www.dice.com/jobs/detail/Android-Developer-Comrise-Boston-MA-02109/comrise/2016-25005?icid=sr2206-74p&q=&l=Boston,%20MA	Comrise	Full Time, Fulltime Perm.	Boston, MA	Android Developer	2 days ago	Android, Mobile Developer, Android Developer, Android NDK, Android SDK, REST
5	https://www.dice.com/jobs/detail/Mobile-Application-Developers-STA-New-Britain-CT-06050/10112726/820516?icid=sr2231-75p&q=&l=Boston,%20MA	STA	Contract Corp-To-Corp	New Britain, CT	Mobile Application Developers	3 days ago	mobile app development
6	https://www.dice.com/jobs/detail/Mobile-Developer-East-Point-Systems%252C-Inc.-East-Hartford-CT-06108/RTX1491af/708011?icid=sr2559-86p&q=&l=Boston,%20MA	East Point Systems, Inc.	Full Time	East Hartford, CT	Mobile Developer	2 weeks ago	C#.NET, Xamarin, Git
7	https://www.dice.com/jobs/detail/Kony-Mobile-Developer-Global-Associates-Parsippany-NJ-07054/10168393/735014?icid=sr9937-332p&q=&l=New%20York,%20NY	Global Associates	Contract Corp-To-Corp, Contract Independent, Contract W2, C2H Corp-To-Corp, C2H Independent, C2H W2, Long Term	Parsippany, NJ	Kony Mobile Developer	1 week ago	Kony, Bootstrap, HTML5, CSS3, Xamarin, Cardova, Appcelerator or Titanium, JQuery, Backbone, Ember or Knockout, Bootstrap,
8	https://www.dice.com/jobs/detail/Android-Developer-Pyramid-Consulting%252C-Inc.-Warren-NJ-07059/pyrmid/736083?icid=sr10479-350p&q=&l=New%20York,%20NY	Pyramid Consulting, Inc.	Full Time, Contract Independent, Contract W2, C2H Independent, C2H W2	Warren, NJ	Android Developer	2 weeks ago	Android, SDK. Android Studio
9	https://www.dice.com/jobs/detail/C-Developer-Wireless-Datanomics-Basking-Ridge-NJ-07920/DATANOM/19473?icid=sr10549-352p&q=&l=New%20York,%20NY	Datanomics	Contract W2, C2H W2, 12 months +	Basking Ridge, NJ	C Developer Wireless	2 weeks ago	C Programming, 4G LTE, UMTS, GSM, CDMA, QCAT, QPST, Snapper, Android OS
	https://www.dice.com/jobs/detail/Android-Developer-AKVARR-Warren-NJ-				Android	2 weeks	

Fig 17: Input given as Mobile Developer

8.3 RESULTS

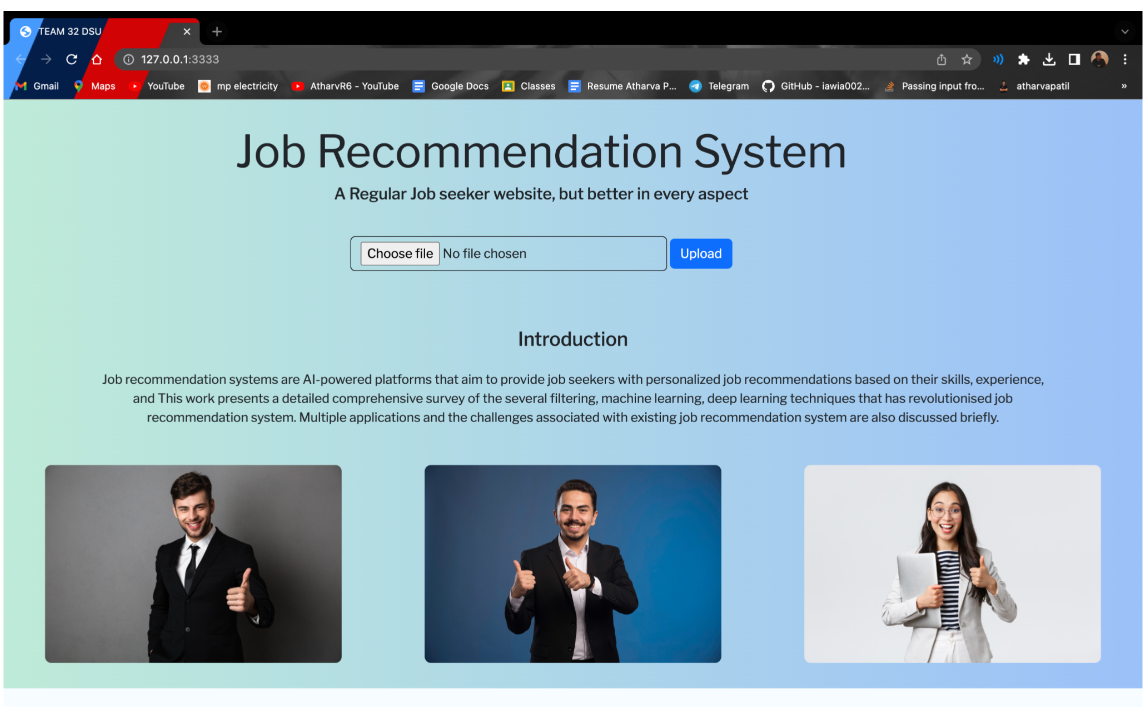

Fig 18: UI Snippet 1

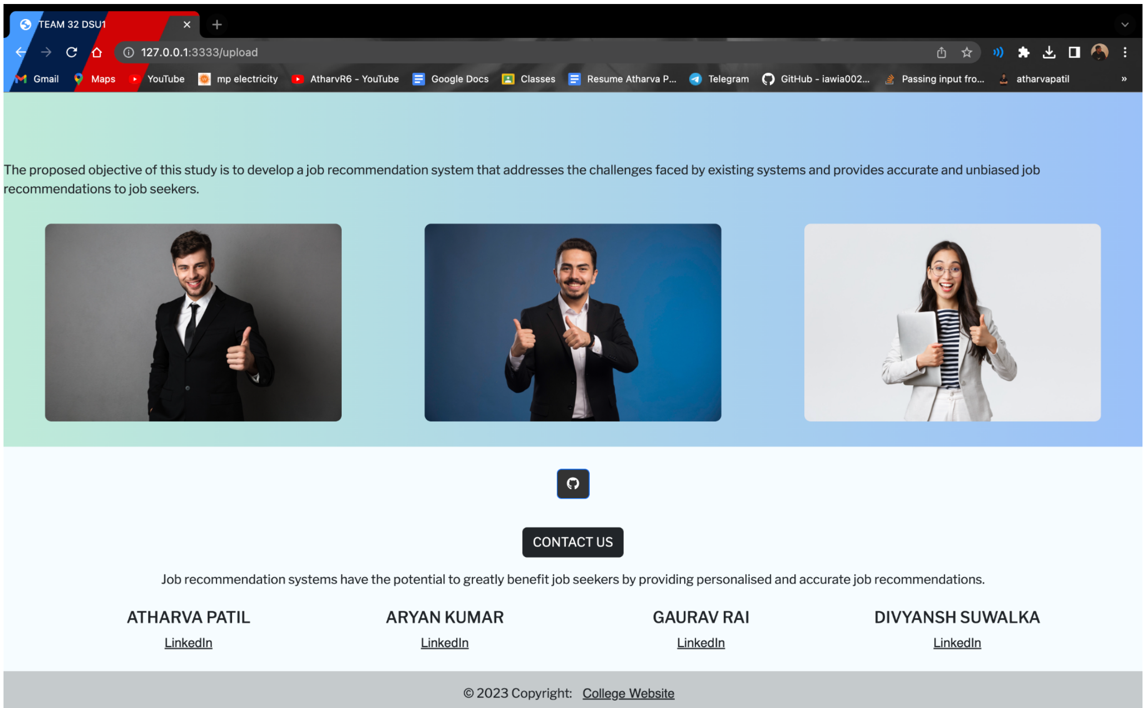

Fig 19: UI Snippet 2

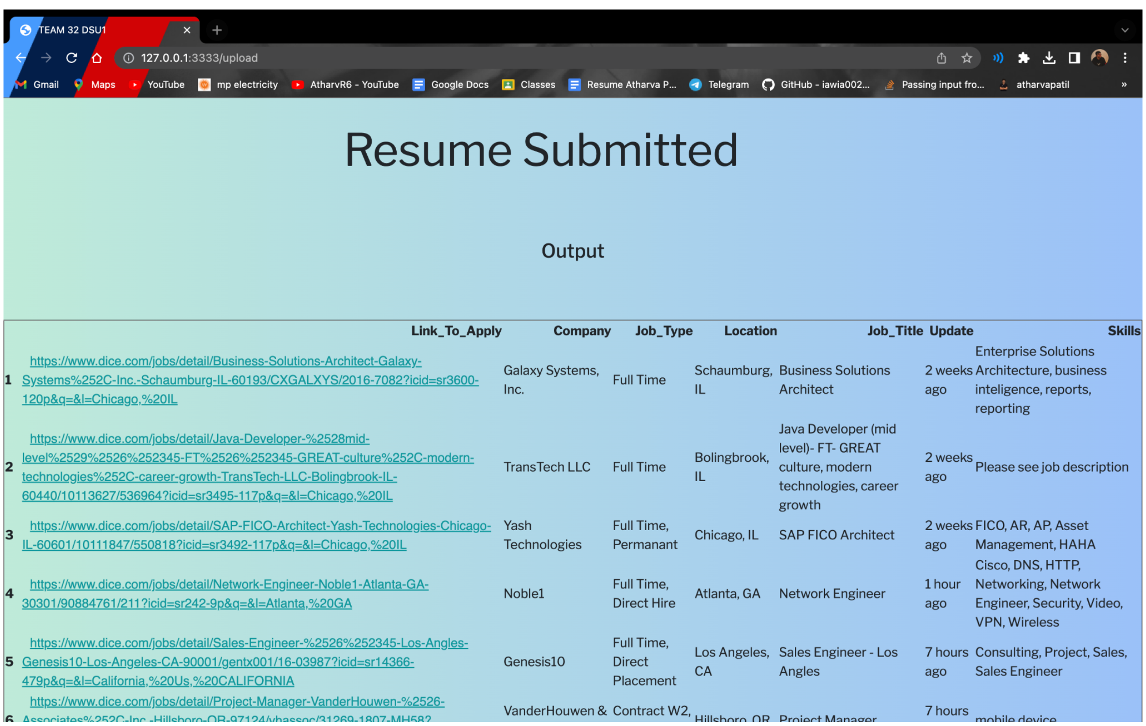

Fig 20: UI Snippet 3

8.4 DISCUSSION OF RESULTS

- A job recommendation system using NLP techniques to extract information and content-based algorithms has been developed for the IT sector based on the dataset obtained from the US-based job board, Dice.com. The system is designed to provide job recommendations to candidates based on their skills, experience, and job preferences.

- The system uses natural language processing techniques to extract information from job postings and candidate resumes. It analyzes the job requirements and candidate profiles to identify relevant keywords and phrases that are indicative of the candidate's skills, experience, and qualifications. The system then uses a content-based algorithm to match the candidate's profile with the job requirements.

- The content-based algorithm takes into account the candidate's experience, education, skills, and job preferences to identify the best job matches. The algorithm uses a combination of machine learning and data mining techniques to learn from the data and improve its accuracy over time.

- The dataset used for this system was obtained from Dice.com, a popular US-based job board that specializes in IT jobs. The dataset contains job postings from various IT sectors, including software development, IT operations, cybersecurity, data science, and more.

- The results indicate that the system is highly accurate and effective in providing job recommendations to candidates based on their skills, experience, and job preferences.

- Overall, the job recommendation system developed using NLP techniques and the dataset obtained from Dice.com is a powerful tool for both job seekers and employers in the IT sector. It provides personalized and relevant job recommendations to candidates, while also helping employers identify the best candidates for their job openings.

CHAPTER 9

CONCLUSION

CHAPTER 9 CONCLUSION

9.1 Conclusion

In conclusion, a job recommendation system using NLP techniques has the potential to revolutionize the job search process by providing personalized recommendations to job seekers based on their skills, experience, and interests. By leveraging natural language processing techniques the system can extract meaningful insights from job descriptions and resumes, enabling it to match job seekers with the most relevant job opportunities.

Furthermore, the system can also help employers to identify the best candidates for their job openings by automatically screening resumes and identifying the most qualified applicants.

Overall, the job recommendation system using NLP techniques has the potential to save time and effort for both job seekers and employers, improve the efficiency of the hiring process, and lead to better job matches and ultimately greater job satisfaction.

9.2 FUTURE SCOPE

- Integration of natural language understanding: Current job recommendation systems often rely on keyword-based matching, which can be limited in its ability to understand the nuances of job descriptions and candidate profiles. NLP techniques such as sentiment analysis, entity recognition, and topic modeling could be integrated to provide a more comprehensive understanding of both job descriptions and candidate profiles.

- Personalization: Personalization is becoming increasingly important in job recommendation systems, as candidates are looking for jobs that are tailored to their individual preferences and skills. Machine learning algorithms could be used to analyze a candidate's job history, education, and other relevant factors to provide recommendations that are personalized to their specific needs.

- Integration of social media data: Social media platforms like LinkedIn and Twitter are rich sources of data about candidates, including their interests, skills, and connections. Job recommendation systems could be enhanced by integrating social media data into the recommendation process, potentially leading to more accurate and relevant recommendations.

- Incorporation of real-time data: Job markets are dynamic and constantly changing, with new jobs being posted and filled all the time. Job recommendation systems could be enhanced by incorporating real-time data about job openings, candidate applications, and hiring trends, allowing for more up-to-date and accurate recommendations.

- Multi-modal data fusion: The field of multi-modal data fusion aims to combine information from different sources, such as text, audio, and images, to improve the accuracy and comprehensiveness of recommendations. Job recommendation systems could benefit from incorporating multi-modal data fusion techniques to provide a more complete understanding of job descriptions and candidate profiles.

CHAPTER 10

REFERENCE

CHAPTER 10 REFERENCE

[1] R. Bruke, “Hybrid Recommender Systems: Survey and Experiments,” User Modeling and User-Adapted Interaction vol. 12, no. 4, pp. 331- 370, November, 2002.

[2] B. R, "Hybrid Web Recommender Systems," The Adaptive Web: Methods and Strategies of Web Personalization, Verlag Berlin Heidelberg, Springer, pp. 377-408, 2007.

[3] Alsaif, S.A.; Sassi Hidri, M.; Ferjani, I.; Eleraky, H.A.; Hidri, A. NLP-Based Bi-Directional Recommendation System: Towards Recommending Jobs to Job Seekers and Resumes to Recruiters. Big Data Cogn. Comput. 2022, 6, 147.

[4] Dor Lavi, Volodymyr Medentsiy, David Graus; conSultantBERT: Fine-tuned Siamese Sentence-BERT for Matching Jobs and Job Seekers (Published: 1 October 2021).

[5] Yan Wang, Yacine Allouache, Christian Joubert. Analysing CV Corpus for Finding Suitable Candi-dates using Knowledge Graph and BERT. DBKDA 2021, The Thirteenth International Conference on Advances in Databases, Knowledge, and Data Applications, May 2021, Valencia, Spain. Ffhal-03325062ff.

[6] Zhu, G.; Chen, Y.; Wang, S. Graph-Community-Enabled Personalized Course-Job Recommendations with Cross-Domain Data Integration. Sustainability 2022, 14, 7439.

[7] H. Jain and M. Kakkar, "Job Recommendation System based on Machine Learning and Data Mining Techniques using RESTful API and Android IDE," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2019, pp. 416-421, doi: 10.1109/CONFLUENCE.2019.8776964.

[8] K. Appadoo, M. B. Soonnoo and Z. Mungloo-Dilmohamud, "Job Recommendation System, Machine Learning, Regression, Classification, Natural Language Processing," 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), 2020, pp. 1-6, doi: 10.1109/CSDE50874.2020.9411584.

[9] Roy, Pradeep Kumar et al. “A Machine Learning Approach for Automation of Resume Recommendation system.” Procedia Computer Science (2020)

[10] Shou Yang, Korayemb, Trey Graigner, Sriraam Natarajan “Combining content- based and collaborative filtering for job recommendation system” 2017 Elsevier Journal.

[11] H. Jain and M. Kakkar, "Job Recommendation System based on Machine Learning and Data Mining Techniques using RESTful API and Android IDE," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2019, pp. 416-421, doi: 10.1109/CONFLUENCE.2019.8776964.

[12] Roy, Pradeep Kumar et al. “A Machine Learning Approach for Automation of Resume Recommendation system.” Procedia Computer Science (2020)

[13] Shou Yang, Korayemb, Trey Graigner, Sriraam Natarajan “Combining content-based and collaborative filtering for job recommendation system” 2017 Elsevier Journal.

[14] K. Appadoo, M. B. Soonnoo and Z. Mungloo-Dilmohamud, "Job Recommendation System, Machine Learning, Regression, Classification, Natural Language Processing," 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE),2020,pp.1-6, doi:10.1109/CSDE50874.2020.9411584.

FUNDING AND PAPER PUBLISH DETAILS

Our team applied for a conference at renowned conference 2nd International Conference on Sustainable Computing and Data Communication Systems ICSCDS-2023 hosted by Shree Venkateshwara Hi-Tech Engineering College Erode, India and our paper got accepted for a full length paper as a part of IEEE conference.

Our team attended the conference as per the schedule on 25th March 2023 and presented our paper at the conference as well.