AI RESUME SHORTLISTING
USING NLP AND MACHINE LEARNING

A MINI PROJECT REPORT

Submitted by

SABARI KAMESWARAN. S

KIRSHOTH. S

THARUN. J. P

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

KONGUNADU COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)

ANNA UNIVERSITY :: CHENNAI 600 025

MAY 2023

KONGUNADU COLLEGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

Tholurpatti (Po), Thottiam (Tk), Trichy (Dt)

COLLEGE VISION & MISSION

VISION

“To become an Internationally Renowned Institution in Technical Education, Research and Development by Transforming the Students into Competent Professionals with Leadership Skills and Ethical Values.”

MISSION

- Providing the Best Resources and Infrastructure.
- Creating Learner centric Environment and continuous –Learning.
- Promoting Effective Links with Intellectuals and Industries.
- Enriching Employability and Entrepreneurial Skills.
- Adapting to Changes for Sustainable Development.

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

VISION

To produce globally competent engineers in the field of Artificial Intelligence and Data Science with a focus on emerging computing needs of the industry and society.

MISSION

- Enrich the students' skills, knowledge with interdisciplinary skill sets by cognitive learning environment and industrial collaboration.
- Promote quality and value based education towards emerging computing needs of the industry and entrepreneurship skills among students.
- Provide for students with leadership qualities, ethical and human values to serve the nation and focus on students' overall development.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO I: Graduates will be able to Artificial Intelligence professionals with expertise in the fields of Artificial Intelligence, Big Data Analytics and Data Science.

PEO II: Graduates will be able to develop problem solving skills and ability to provide solution for real time problems.

PEO III: Graduates shall have professional ethics, team spirit, life-long learning, communication skills and adopt corporate culture, core values and leadership skills.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Professional skills: Students shall excel in software development including Artificial Intelligence technologies to solve complex computation task with soft skills.

PSO2: Competency: Students shall qualify at the State, National and International level competitive examination for employment, higher studies and research.

PROGRAM OUTCOMES (PO's)

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identity, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using the first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and the ural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis, and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineeringsolutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and teamwork**: Function effectively as an individual, and as a member or leader in diverse teams, and multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective

presentations, and give and receiveclear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-Long Learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

KONGUNADU COLLEGE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this Mini Project report **"AI RESUME SHORTLISTING USING NLP & MACHINE LEARNING"** is the bonafide work of **"SABARI KAMESWARAN S (621321243042), KIRSHOTH S (621321243019), THARUN J P (621321243057)"** who carried out the Mini Project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.K.Baskar, M.Tech., Ph.D.,	Mr.M.Sivakumar, M.E.(Ph.D).,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Department of Artificial Intelligence and Data Science, Kongunadu College of Engineering and Technology, Thottiam, Trichy.	Assistant Professor, Department of Artificial Intelligence and Data Science, Kongunadu College of Engineering and Technology, Thottiam, Trichy.

Submitted for the Mini Project Viva-Voce examination held on ________________

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

We wish to express our sincere thanks to our beloved Chairman **Dr.PSK.R.PERIASWAMY** for providing immense facilities in our institution.

We proudly render our thanks to our Principal **Dr.R.ASOKAN, M.S., M.Tech., Ph.D.,** for the facilities and the encouragement was given by him tothe progress and completion of our mini project.

We proudly render our immense gratitude and sincere thanks to our Head of the Department of Artificial Intelligence and Data Science **Dr.K.Baskar, M.Tech., Ph.D.,** for his effective leadership,encouragement, and, guidance in the mini project.

We are highly indebted to provide our heart full thanks to our supervisor **Mr.M.Sivakumar, M.E., (Ph.D).,** for his valuable suggestion during execution of our project work and for continued encouragement in conveying us for making many constructive comments for improving comments the operation of this mini project report.

We are highly indebted to provide our heart full thanks to our mini project coordinator **Mr.M.Sivakumar, M.E., (Ph.D).,** for his valuable ideas, constant encouragement, and supportive guidance throughout the project.

We wish to extend our sincere thanks to all teaching and non-teaching staffof the Artificial Intelligence and Data Science department for their valuable suggestion, cooperation, and encouragement in the successful completion of this mini project.

ABSTRACT

The growing demand for technical and business expertise in the IT industry has created a shortage of qualified professionals in the market. With the need for companies to quickly find and hire the best-suited candidates for their open positions, traditional hiring methods have proven to be time-consuming and inefficient. To address this issue, companies have turned to AI-based screening software like Resume Screening to streamline the hiring process. This software is designed to analyze resumes and shortlist the best candidates based on the required skill sets for the specified job positions. It reduces the burden on hiring professionals who would otherwise have to sort through a large number of resumes, saving them valuable time and effort. The system utilizes natural language processing (NLP) techniques to clean the resumes and remove stop words. The data is then visualized using data visualization tools like SEA-BORN and MATPLOTLIB, making it easier for hiring managers to analyze the information and make informed decisions. The AI-based screener has already been trained with the necessary skill sets for most job positions, further streamlining the hiring process. This ensures that only the most qualified candidates are shortlisted, reducing the chances of a bad hire and ultimately saving the company time and money. The results are displayed on a website created using HTML, CSS, and JS, which is integrated with a MongoDB database for efficient data management. Professionals can easily upload their resumes on the portal, which will be sent to the AI system for analysis. Overall, the use of AI-based screening software like Resume Screening has revolutionized the hiring process for the IT industry. It provides an efficient and cost-effective way to identify and shortlist the best-suited candidates for the job, helping companies stay competitive in the market.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	**A**pplication **P**rogramming **I**nterface
NLP	**N**atural **L**anguage **P**rocessing
CSS	**C**ascading **S**tyle **S**heet
HTML	**H**ypertext **M**arkup **L**anguage
JS	**J**ava**S**cript
JSON	**J**ava**S**cript **O**bject **N**otation
ML	**M**achine **L**earning
NO-SQL	**N**ot only **SQL**
PKL	**P**ic**k**e**l**
CSV	**C**omma **S**eparated **V**alues
IPYNB	**IPY**thon **N**ote**B**ook

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The IT industry is facing a shortage of qualified professionals, which has created a pressing need for companies to quickly find and hire the best-suited candidates for their open positions. The traditional hiring process can be time-consuming, expensive, and inefficient, leading many organizations to look for alternative solutions.

This project proposes the use of AI-based screening software called Resume Screening to address these challenges and streamline the hiring process. The software utilizes NLP techniques to clean the resumes, remove stop words, and analyze the data, making it easier for hiring managers to make informed decisions. The AI-based screener has alreay been trained with the necessary skill sets for most job positions, further simplifying the hiring process.

To present the data, data visualization tools like SEA-BORN and MATPLOTLIB are used, providing a clear and concise overview of the information. The results are displayed on a website created using HTML, CSS, and JS and integrated with a MongoDB database, providing efficient data management.

Professionals can easily upload their resumes on the portal, which will be sent to the AI system for analysis, reducing the time and effort required for manual resume screening. This approach helps HR departments find the best-suited candidates for the specified jobs, ultimately saving time and money for the company.

Overall, this project provides an innovative and cost-effective solution to the challenges faced by the IT industry in finding and hiring qualified professionals quickly and efficiently.

In this project, Natural Language Processing (NLP) techniques are utilized to analyze and process resumes for efficient and accurate screening. NLP is a field of artificial intelligence that focuses on the interaction between computers and human language. It enables the system to understand, interpret, and extract meaningful information from textual data.

The NLP techniques used in this project include:

1. Data Cleaning: NLP is employed to preprocess the resume data by removing irrelevant characters, punctuation, and special symbols. This step ensures that the text is clean and ready for further analysis.

2. Stopword Removal: NLP techniques are used to identify and remove common stopwords (e.g., articles, prepositions) from the resumes. This helps in eliminating noise and focusing on the more significant content.

3. Skill Extraction: NLP algorithms are applied to extract relevant skills and qualifications from the resumes. This involves techniques such as named entity recognition, keyword extraction, and semantic analysis to identify specific skills mentioned in the text.

4. Semantic Analysis: NLP methods are utilized to understand the contextual meaning of the resume content. This involves analyzing the relationships between words, phrases, and sentences to extract the intended meaning, such as identifying the level of proficiency in a particular skill.

5. Sentiment Analysis: NLP algorithms are employed to analyze the sentiment expressed in the resumes. This helps in understanding the candidate's attitude, enthusiasm, and overall fit for the job position.

6. Language Modeling: NLP techniques are used to build language models that capture the structure and patterns of the resume text. These models enable the system to predict and generate meaningful responses, such as matching the candidate's skills with the job requirements.

By leveraging these NLP techniques, the system can process and analyze resumes in a structured and meaningful way. This enables HR professionals to effectively screen and shortlist candidates based on their qualifications, skills, and compatibility with the job requirements. The NLP component plays a crucial role in improving the accuracy, efficiency, and objectivity of the resume screening process.

In this project, machine learning techniques are employed to enhance the resume screening process. Machine learning is a subset of artificial intelligence that enables computers to learn from data and make predictions or decisions without explicit programming.

The machine learning techniques used in this project include:

1. Training the AI Model: Machine learning algorithms are trained using a dataset that consists of pre-screened resumes and their corresponding labels (e.g., shortlisted or not shortlisted). The model learns patterns and relationships within the data to make accurate predictions.

2. Feature Extraction: Machine learning algorithms extract relevant features from the resumes, such as skills, qualifications, work experience, and education. These features are used as input for the model to understand the candidate's profile.

3. Classification: The trained machine learning model performs classification by assigning probabilities or labels to each resume, indicating the likelihood of a candidate being shortlisted for a particular job. This classification helps in effectively filtering and prioritizing the resumes.

4. Model Evaluation: The performance of the machine learning model is evaluated using various metrics, such as accuracy, precision, recall, and F1 score. This evaluation helps in assessing the model's effectiveness and fine-tuning it for better performance.

5. Model Optimization: The machine learning model is optimized by adjusting hyperparameters, selecting appropriate algorithms, and employing techniques such as cross-validation and regularization. Optimization ensures that the model performs well on unseen data and generalizes to different resume profiles.

By incorporating machine learning techniques, the system can learn from past resume screening decisions and make predictions based on learned patterns. This enhances the efficiency and accuracy of the screening process, enabling HR professionals to focus their attention on the most qualified candidates.

It's important to note that the machine learning component of the project works in tandem with the NLP techniques. The NLP techniques extract relevant features from resumes, which are then used as inputs for the machine learning model. This combination of NLP and machine learning enhances the system's ability to accurately evaluate and shortlist candidates based on their qualifications and skills.

1.2 OBJECTIVES AND GOALS

The primary objectives of the project "AI Resume Shortlisting using NLP & Machine Learning " are:

1. To develop an AI-based screening software called Resume Screening to streamline the hiring process for the IT industry.
2. To utilize natural language processing (NLP) techniques to clean the resumes and remove stop words, making it easier for hiring managers to analyze the information and make informed decisions.
3. To train the AI-based screener with necessary skill sets for most job positions, further simplifying the hiring process.
4. To present the data using data visualization tools like SEA-BORN and MATPLOTLIB, providing a clear and concise overview of the information.
5. To develop a website integrated with a MongoDB database, enabling professionals to easily upload their resumes for analysis by the AI system, reducing the time and effort required for manual resume screening and ultimately helping HR departments find the best-suited candidates for the specified jobs.

Goals:

The primary goals of the project " AI Resume Shortlisting using NLP & Machine Learning " are:

1. To improve the efficiency of the hiring process in the IT industry by reducing the time and effort required for manual resume screening.
2. To increase the accuracy of candidate selection by utilizing AI-based screening software that analyzes resumes and shortlists the best-suited candidates based on the required skill sets for the specified job positions.
3. To reduce costs associated with the hiring process by streamlining the screening process and minimizing the need for dedicated screening officers.
4. To enhance the user experience for professionals by providing an easy-to-use portal for uploading resumes and receiving feedback on their applications.
5. To promote diversity and inclusion in the hiring process by reducing the potential for unconscious bias in the screening process, resulting in a more diverse pool of candidates.

CHAPTER 2
LITERATURE SURVEY

[1] TITLE: Resume Screening Using Natural Language Processing and Machine Learning (2021)

AUTHOR: Arvind Kumar Sinha, Md. Amir Khusru Akhtar and Ashwani Kumar.

Curriculum vitae or resume screening is a time-consuming procedure. Natural language processing and machine learning have the capability to understand and parse the unstructured written language, and extract the desired information. The idea is to train the machine to analyze the written documents like a human being. This paper presents a systematic review on resume screening and enlightens the comparison of recognized works. Several techniques and approaches of machine learning for evaluating and analyzing the unstructured data have been discussed. Existing resume parsers use semantic search to understand the context of the language in order to find the reliable and comprehensive results. A review on the use of semantic search for context-based searching has been explained. In addition, this paper also shows the research challenges and future scope of resume parsing in terms of writing style, word choice and syntax of unstructured written language.

Techniques: Natural Language Processing and Machine Learning.

Merits: The use of NLP and machine learning in resume screening can save time, improve accuracy, and provide reliable and comprehensive results. Semantic search enables context-based searching, and ongoing research can address challenges and improve the process.

Overcome: The quality and quantity of data utilized to train the ML model may have an impact on how accurately crops are recommended. Estimates that are incorrect might result in poor crop yields and financial losses for farmers.

[2] TITLE: Web Application for Screening Resume (2020)

AUTHOR: Atharva Jadhav, Nihar Riswadkar, Pranay Jadhav,Yash Gogawale.

This paper focuses majorly on the design of the web application which will be used to screen resumes (Curriculum Vitae) for a particular job posting. In the proposed system, a web application will encourage the job applicant candidates as well as the recruiters to use it for job applications and screening of resumes. Recruitment is a tedious process wherein the first task for any recruiter is to screen the resumes. The proposed web application is designed in such a way that job applicant as well as recruiters can use it with ease for applying for job openings and screening respectively. The recruiters from various companies can post the details of the job openings available in their respective companies. The interactive web application will allow the job applicants to submit their resume and apply for their job postings they may still be interested in. The resumes submitted by the candidates are then compared with the job profile requirement posted by the company recruiter by using techniques like machine learning and Natural Language Processing (NLP). Scores can then be given to the resumes and they can be ranked from highest match to lowest match. This ranking is made visible only to the company recruiter who is interested to select the best candidates from a large pool of candidates.

Techniques: Machine Learning, NLP , HTML , CSS , JS .

Merits: Proposed web app streamlines recruitment by allowing easy submission of resumes and job postings. Machine learning and NLP used to compare resumes with job profiles, enabling efficient screening and ranking of candidates.

Overcome: The proposed web app aims to overcome the challenges of time-consuming and manual resume screening processes by using advanced techniques like machine learning and NLP to automate and streamline the process, benefiting both job applicants and recruiters.

[3] TITLE: Recruiters perceptions and use of biographical information in resume screening. (2020)

AUTHOR: Shreya S. Bhanose, Kalyani A. Bogawar.

Recruiters' phenomenological perceptions of biodata in resumes were examined along with their use of biodata for making applicant screening decisions. The thesis was that biodata were interpreted as indicating abilities and other attributes. Three complementary studies were conducted with 344 recruiters from 28 companies. Study 1 survey results indicated recruiters judged biodata to reflect both ability (language, math, physical) and other (interpersonal, leadership, motivation) attributes. Both types of attributes were judged with high interrecruiter reliability, and attributes judged to be reflected depended partly on the job considered. Study 2 experimental results indicated recruiters rated resumes more attractive to the degree that biodata in the resumes reflected attributes required by the jobs. Study 3 protocol analysis results confirmed that recruiters considered these attributes with substantial frequency.

Techniques: Neural networks and Machine Learning and Artificial Neural Network.

Merits: This research found that recruiters perceive biodata in resumes as indicating abilities and other attributes, and use them to make screening decisions. The attributes judged depended on the job, and resumes were rated more attractive when they reflected job requirements.

Overcome: The study provides valuable insights for job seekers to tailor their resumes to job requirements and highlight relevant attributes. Recruiters should be aware of potential biases in their interpretation of biodata.

[4] TITLE: Data Extraction Using NLP Techniques and Its Transformation to Linked Data (2019)

AUTHOR: Nasreen Taj M B, Kavya H C, Nayana R R, Bindu H S, Meghana D.P.

We present a system that extracts a knowledge base from raw unstructured texts that is designed as a set of entities and their relations and represented in an ontological framework. The extraction pipeline processes input texts by linguistically-aware tools and extracts entities and relations from their syntactic representation. Consequently, the extracted data is represented according to the Linked Data principles. The system is designed both domain and language independent and provides users with data for more intelligent search than full-text search. We present our first case study on processing Czech legal texts. **Techniques:** K-Nearest Neighbor, Recommendation System.

Merits: This system provides a method for extracting a knowledge base from unstructured text, represented in an ontological framework, using Linked Data principles. It's domain and language independent, enabling more intelligent search than full-text search. The case study demonstrates its effectiveness in processing Czech legal texts.

Overcome: Potential challenges for this system include ensuring accuracy and completeness of the extracted entities and relations, and the need for continuous updating as new information becomes available. Additionally, the system may require customization for specific domains or languages

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system for resume shortlisting based on use data analytics and machine learning algorithms to analyze various HR-related data, including resumes and job descriptions. They provide insights into various metrics, such as candidate quality and time-to-hire, helping companies make data-driven recruitment decisions.

3.1.1 Disadvantages:

1. **Time-consuming :** The manual process of resume shortlisting is often time-consuming, especially for companies receiving a large number of job applications. The hiring team must go through each resume individually, which can take a lot of time and effort.
2. **Bias :** The manual process of resume shortlisting can be influenced by personal biases of the hiring team, such as race, gender, and education. This can lead to potential discrimination and exclusion of qualified candidates.
3. **Inconsistency :** The manual process of resume shortlisting can lead to inconsistencies in candidate selection, as different hiring team members may have different evaluation criteria and methods.
4. **Costly :** The manual process of resume shortlisting can be costly, as it requires hiring dedicated screening officers or outsourcing the task to third-party companies.
5. **Limited scalability :** The manual process of resume shortlisting may not be scalable, especially for companies experiencing rapid growth or receiving a large number of job applications. The hiring team may struggle to keep up with the workload, leading to delays and potential loss of qualified candidates.

3.2 PROPOSED SYSTEM

The proposed system is an automated resume screening software that utilizes AI and NLP algorithms to analyze resumes and shortlist the best-suited candidates based on required skill sets for the specified job positions. It will provide an online portal for professionals to upload their resumes, and the system will clean and remove any irrelevant information using NLP techniques. Then, it will match the required skills with the resume and provide feedback to the user. The system aims to improve the efficiency and accuracy of the hiring process, reduce the cost, and minimize biases in the hiring process while promoting diversity and inclusion.

3.2.1 Advantages

The proposed resume shortlisting offers several advantages over the existing manual shortlisting process, including:

1. **Increased efficiency :** The proposed system will significantly increase the efficiency of the resume screening process by automating it, saving time, and reducing the workload of the hiring team.
2. **Enhanced accuracy :** The system utilizes AI and NLP algorithms to provide more accurate shortlisting of candidates based on required skill sets, reducing the chances of human error and biases.
3. **Cost-effective :** The proposed system will help companies save costs by eliminating the need to hire dedicated screening officers or outsourcing the task to third-party companies.
4. **Promotes diversity and inclusion :** The system will eliminate biases in the hiring process, ensuring that the best-suited candidates are selected regardless of their race, gender, or education.
5. **Scalability :** The proposed system can easily handle a large volume of job applications, making it scalable for companies that experience rapid growth or receive a high volume of job applications.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

Processor	Multi-core processor with a clock speed of at least 2.5 GHz Or higher.
RAM	8GB
Hard disk	Solid-State Drives (SSDs)- 500GB or higher
Keyboard	Standard keyboard and mouse
Monitor	LCD or LED displays with at least 1920 x 1080 resolution

4.2 SOFTWARE REQUIREMENTS

TOOL	Visual Studio Code
Frontend	HTML, CSS, JS
Framework	Python Flask Framework
Languages	Python 3 (Python 3.7 and above must be installed)
Operating system	Windows 11
Technologies	Python 3.7, Flask, NLP, Machine Learning

CHAPTER 5

SYSTEM DESIGN

5.1 ARCHITECTURE DIAGRAM

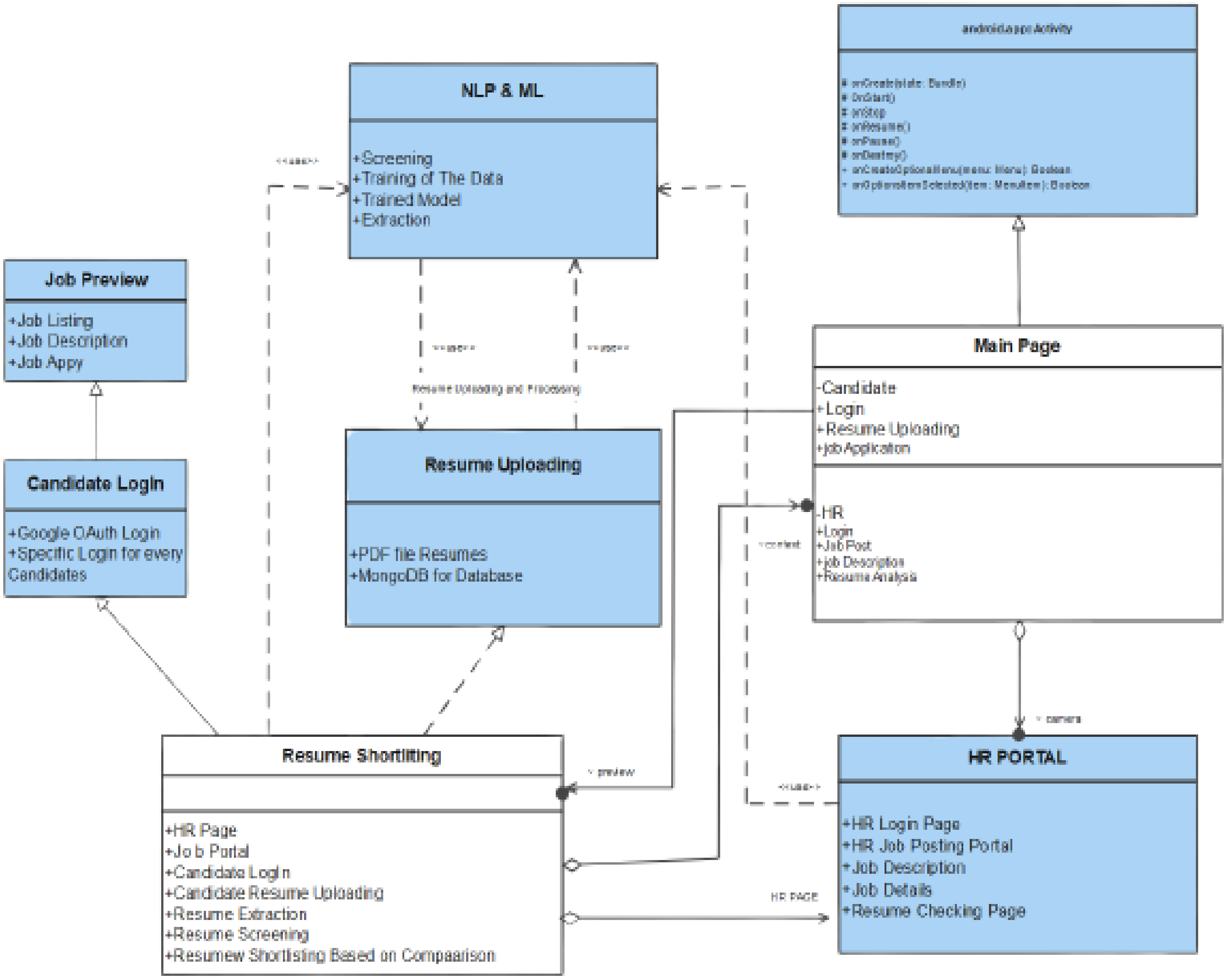


Figure 5.1 Architecture diagram

The Visual Representation shows the different modules of an AI Resume Shortlisting System. There are Three Module, Candidate Module, HR Module, and Resume Screening.

1. **CANDIDATE MODULE: -** The Candidate Module has Login Page for the Candidate Login to the system and then The Candidate will allowed for the Resume Uploading page. Then the Candidates can apply the jobs for them.

2. **HR MODULE: -** The HR Module has Login Page for the HR Login to the System. Then the HR is allowed to Post Jobs and Job description. After the application date. The HR will have the analysis of the Resumes that have been Uploaded.
3. **RESUME SCREENING: -** The Resume Screening module has two technologies NLP and ML. Both the technologies plays an vital role in this. The NLP is used for training the data.

Natural Language Processing:-

Natural Language Processing (NLP) is a field of artificial intelligence that focuses on the interaction between computers and human language. In the proposed system, NLP techniques are employed to analyze and process resumes to enhance the effectiveness of the screening process. Here is a brief overview of NLP and its key components:

1. **Text Preprocessing:** NLP involves various preprocessing steps to clean and prepare the text data for further analysis. Common preprocessing tasks include removing punctuation, tokenization (breaking text into individual words or tokens), and removing stop words (common words that do not carry significant meaning).
2. **Named Entity Recognition (NER):** NER is a technique used to identify and classify named entities such as names, locations, organizations, and dates within the resumes. This helps in extracting important information and identifying key skills, educational background, and work experience.
3. **Part-of-Speech (POS) Tagging:** POS tagging is the process of labeling each word in a sentence with its corresponding part of speech (e.g., noun, verb, adjective). POS tagging assists in understanding the grammatical structure of the text, which can be useful for extracting relevant information from resumes.
4. **Entity Linking:** Entity linking is the process of linking named entities in the text to a knowledge base or external sources, providing additional context and information. It helps in disambiguating entities and associating them with relevant entities in the real world.

5. **Sentiment Analysis:** Sentiment analysis aims to determine the sentiment or emotional tone expressed in the text, whether positive, negative, or neutral. This can be useful in evaluating the overall tone of the resume or identifying subjective information provided by the candidates.
6. **Information Extraction:** Information extraction techniques are used to extract specific information from the resumes, such as educational qualifications, work experience, and skills. This involves parsing the text to identify relevant patterns and extracting structured information for further analysis.
7. **Document Classification:** NLP can be employed for document classification tasks, where resumes can be classified into different categories based on job positions or skill sets. This allows for efficient organization and filtering of resumes based on specific job requirements.
8. **Language Modeling:** Language modeling involves developing statistical models that capture the patterns and relationships between words in a given language. These models can be utilized to predict the likelihood of certain phrases or words occurring, enabling more accurate understanding and generation of text.
9. **Machine Translation:** NLP techniques can also be applied to machine translation tasks, where resumes written in different languages can be automatically translated to the desired language, facilitating global recruitment processes.
10. **Increased efficiency :** The proposed system will significantly increase the efficiency of the resume screening process by automating it, saving time, and reducing the workload of the hiring team.
11. **Promotes diversity and inclusion :** The system will eliminate biases in the hiring process, ensuring that the best-suited candidates are selected regardless of their race, gender, or education.
12. **Enhanced accuracy :** The system utilizes AI and NLP algorithms to provide more accurate shortlisting of candidates based on required skill sets, reducing the chances of human error and biases.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 MODULES

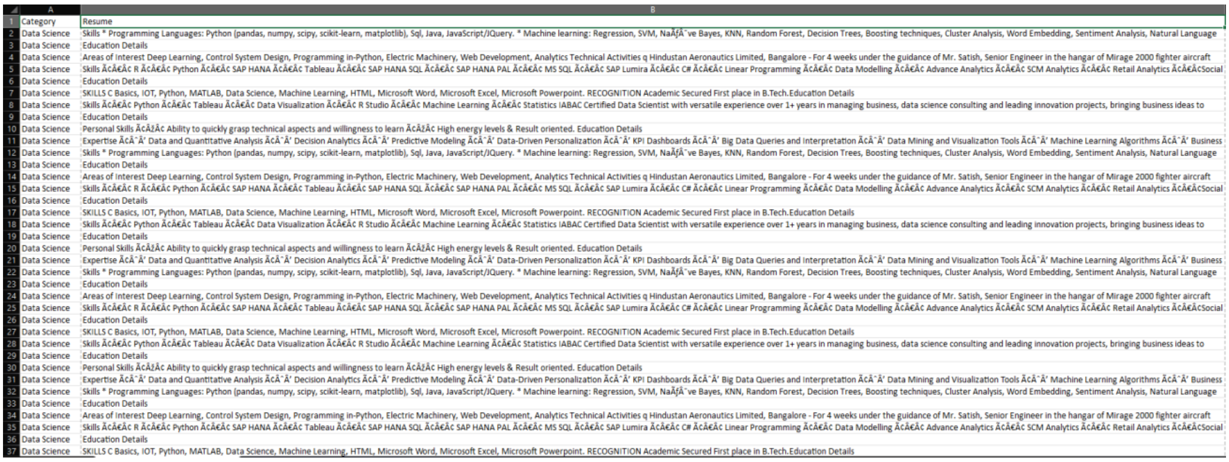
To effectively implement a Resume Shortlisting System, it is important to categorize the different stages involved in the process. The different segments of the system implementation can be categorized as follows:

- Data Collection
- Data Processing
- Model Building
- Data Training
- HR & Candidate Login Page

6.2 MODULE DESCRIPTION

6.2.1. Data Collection

The first module would involve collecting data about the jobs, such as its type, it’s job category, skills and all about Jobs. This data could be collected from various sources such as Jobs, Skills. Kaggle offers a wide range of job skill recommendation data, which can improve the accuracy of the system's predictions. Kaggle also provides valuable information and feedback that can help users make better decisions. To effectively implement a Resume Shortlisting System, it is important to categorize the different stages involved in the process. This data could be collected from various sources such as Jobs, Skills. Language modeling involves developing statistical models that capture the patterns and relationships between words in a given language. NLP techniques can also be applied to machine translation tasks, where resumes written in different languages can be automatically translated to the desired language, facilitating global recruitment processes.



	A	B
1	Category	Resume
2	Data Science	Skills * Programming Languages: Python (pandas, numpy, scipy, scikit-learn, matplotlib), Sql, Java, JavaScript/JQuery. * Machine learning: Regression, SVM, NaÃƒÂ¯ve Bayes, KNN, Random Forest, Decision Trees, Boosting techniques, Cluster Analysis, Word Embedding, Sentiment Analysis, Natural Language
3	Data Science	Education Details
4	Data Science	Areas of Interest Deep Learning, Control System Design, Programming in-Python, Electric Machinery, Web Development, Analytics Technical Activities q Hindustan Aeronautics Limited, Bangalore - For 4 weeks under the guidance of Mr. Satish, Senior Engineer in the hangar of Mirage 2000 fighter aircraft
5	Data Science	Skills ÃcÂ€Âc R ÃcÂ€Âc Python ÃcÂ€Âc SAP HANA ÃcÂ€Âc Tableau ÃcÂ€Âc SAP HANA SQL ÃcÂ€Âc SAP HANA PAL ÃcÂ€Âc MS SQL ÃcÂ€Âc SAP Lumira ÃcÂ€Âc C# ÃcÂ€Âc Linear Programming ÃcÂ€Âc Data Modelling ÃcÂ€Âc Advance Analytics ÃcÂ€Âc SCM Analytics ÃcÂ€Âc Retail Analytics ÃcÂ€ÂcSocial
6	Data Science	Education Details
7	Data Science	SKILLS C Basics, IOT, Python, MATLAB, Data Science, Machine Learning, HTML, Microsoft Word, Microsoft Excel, Microsoft Powerpoint. RECOGNITION Academic Secured First place in B.Tech.Education Details
8	Data Science	Skills ÃcÂ€Âc Python ÃcÂ€Âc Tableau ÃcÂ€Âc Data Visualization ÃcÂ€Âc R Studio ÃcÂ€Âc Machine Learning ÃcÂ€Âc Statistics IABAC Certified Data Scientist with versatile experience over 1+ years in managing business, data science consulting and leading innovation projects, bringing business ideas to
9	Data Science	Education Details
10	Data Science	Personal Skills ÃcÂžÂc Ability to quickly grasp technical aspects and willingness to learn ÃcÂžÂc High energy levels & Result oriented. Education Details
11	Data Science	Expertise ÃcÂˆÂ' Data and Quantitative Analysis ÃcÂˆÂ' Decision Analytics ÃcÂˆÂ' Predictive Modeling ÃcÂˆÂ' Data-Driven Personalization ÃcÂˆÂ' KPI Dashboards ÃcÂˆÂ' Big Data Queries and Interpretation ÃcÂˆÂ' Data Mining and Visualization Tools ÃcÂˆÂ' Machine Learning Algorithms ÃcÂˆÂ' Business
12	Data Science	Skills * Programming Languages: Python (pandas, numpy, scipy, scikit-learn, matplotlib), Sql, Java, JavaScript/JQuery. * Machine learning: Regression, SVM, NaÃƒÂ¯ve Bayes, KNN, Random Forest, Decision Trees, Boosting techniques, Cluster Analysis, Word Embedding, Sentiment Analysis, Natural Language
13	Data Science	Education Details
14	Data Science	Areas of Interest Deep Learning, Control System Design, Programming in-Python, Electric Machinery, Web Development, Analytics Technical Activities q Hindustan Aeronautics Limited, Bangalore - For 4 weeks under the guidance of Mr. Satish, Senior Engineer in the hangar of Mirage 2000 fighter aircraft
15	Data Science	Skills ÃcÂ€Âc R ÃcÂ€Âc Python ÃcÂ€Âc SAP HANA ÃcÂ€Âc Tableau ÃcÂ€Âc SAP HANA SQL ÃcÂ€Âc SAP HANA PAL ÃcÂ€Âc MS SQL ÃcÂ€Âc SAP Lumira ÃcÂ€Âc C# ÃcÂ€Âc Linear Programming ÃcÂ€Âc Data Modelling ÃcÂ€Âc Advance Analytics ÃcÂ€Âc SCM Analytics ÃcÂ€Âc Retail Analytics ÃcÂ€ÂcSocial
16	Data Science	Education Details
17	Data Science	SKILLS C Basics, IOT, Python, MATLAB, Data Science, Machine Learning, HTML, Microsoft Word, Microsoft Excel, Microsoft Powerpoint. RECOGNITION Academic Secured First place in B.Tech.Education Details
18	Data Science	Skills ÃcÂ€Âc Python ÃcÂ€Âc Tableau ÃcÂ€Âc Data Visualization ÃcÂ€Âc R Studio ÃcÂ€Âc Machine Learning ÃcÂ€Âc Statistics IABAC Certified Data Scientist with versatile experience over 1+ years in managing business, data science consulting and leading innovation projects, bringing business ideas to
19	Data Science	Education Details
20	Data Science	Personal Skills ÃcÂžÂc Ability to quickly grasp technical aspects and willingness to learn ÃcÂžÂc High energy levels & Result oriented. Education Details
21	Data Science	Expertise ÃcÂˆÂ' Data and Quantitative Analysis ÃcÂˆÂ' Decision Analytics ÃcÂˆÂ' Predictive Modeling ÃcÂˆÂ' Data-Driven Personalization ÃcÂˆÂ' KPI Dashboards ÃcÂˆÂ' Big Data Queries and Interpretation ÃcÂˆÂ' Data Mining and Visualization Tools ÃcÂˆÂ' Machine Learning Algorithms ÃcÂˆÂ' Business
22	Data Science	Skills * Programming Languages: Python (pandas, numpy, scipy, scikit-learn, matplotlib), Sql, Java, JavaScript/JQuery. * Machine learning: Regression, SVM, NaÃƒÂ¯ve Bayes, KNN, Random Forest, Decision Trees, Boosting techniques, Cluster Analysis, Word Embedding, Sentiment Analysis, Natural Language
23	Data Science	Education Details
24	Data Science	Areas of Interest Deep Learning, Control System Design, Programming in-Python, Electric Machinery, Web Development, Analytics Technical Activities q Hindustan Aeronautics Limited, Bangalore - For 4 weeks under the guidance of Mr. Satish, Senior Engineer in the hangar of Mirage 2000 fighter aircraft
25	Data Science	Skills ÃcÂ€Âc R ÃcÂ€Âc Python ÃcÂ€Âc SAP HANA ÃcÂ€Âc Tableau ÃcÂ€Âc SAP HANA SQL ÃcÂ€Âc SAP HANA PAL ÃcÂ€Âc MS SQL ÃcÂ€Âc SAP Lumira ÃcÂ€Âc C# ÃcÂ€Âc Linear Programming ÃcÂ€Âc Data Modelling ÃcÂ€Âc Advance Analytics ÃcÂ€Âc SCM Analytics ÃcÂ€Âc Retail Analytics ÃcÂ€ÂcSocial
26	Data Science	Education Details
27	Data Science	SKILLS C Basics, IOT, Python, MATLAB, Data Science, Machine Learning, HTML, Microsoft Word, Microsoft Excel, Microsoft Powerpoint. RECOGNITION Academic Secured First place in B.Tech.Education Details
28	Data Science	Skills ÃcÂ€Âc Python ÃcÂ€Âc Tableau ÃcÂ€Âc Data Visualization ÃcÂ€Âc R Studio ÃcÂ€Âc Machine Learning ÃcÂ€Âc Statistics IABAC Certified Data Scientist with versatile experience over 1+ years in managing business, data science consulting and leading innovation projects, bringing business ideas to
29	Data Science	Education Details
30	Data Science	Personal Skills ÃcÂžÂc Ability to quickly grasp technical aspects and willingness to learn ÃcÂžÂc High energy levels & Result oriented. Education Details
31	Data Science	Expertise ÃcÂˆÂ' Data and Quantitative Analysis ÃcÂˆÂ' Decision Analytics ÃcÂˆÂ' Predictive Modeling ÃcÂˆÂ' Data-Driven Personalization ÃcÂˆÂ' KPI Dashboards ÃcÂˆÂ' Big Data Queries and Interpretation ÃcÂˆÂ' Data Mining and Visualization Tools ÃcÂˆÂ' Machine Learning Algorithms ÃcÂˆÂ' Business
32	Data Science	Skills * Programming Languages: Python (pandas, numpy, scipy, scikit-learn, matplotlib), Sql, Java, JavaScript/JQuery. * Machine learning: Regression, SVM, NaÃƒÂ¯ve Bayes, KNN, Random Forest, Decision Trees, Boosting techniques, Cluster Analysis, Word Embedding, Sentiment Analysis, Natural Language
33	Data Science	Education Details
34	Data Science	Areas of Interest Deep Learning, Control System Design, Programming in-Python, Electric Machinery, Web Development, Analytics Technical Activities q Hindustan Aeronautics Limited, Bangalore - For 4 weeks under the guidance of Mr. Satish, Senior Engineer in the hangar of Mirage 2000 fighter aircraft
35	Data Science	Skills ÃcÂ€Âc R ÃcÂ€Âc Python ÃcÂ€Âc SAP HANA ÃcÂ€Âc Tableau ÃcÂ€Âc SAP HANA SQL ÃcÂ€Âc SAP HANA PAL ÃcÂ€Âc MS SQL ÃcÂ€Âc SAP Lumira ÃcÂ€Âc C# ÃcÂ€Âc Linear Programming ÃcÂ€Âc Data Modelling ÃcÂ€Âc Advance Analytics ÃcÂ€Âc SCM Analytics ÃcÂ€Âc Retail Analytics ÃcÂ€ÂcSocial
36	Data Science	Education Details
37	Data Science	SKILLS C Basics, IOT, Python, MATLAB, Data Science, Machine Learning, HTML, Microsoft Word, Microsoft Excel, Microsoft Powerpoint. RECOGNITION Academic Secured First place in B.Tech.Education Details

Figure 6.1(a). Data Collection (Train and Test Data)

Let's see what different categories we have in the data.

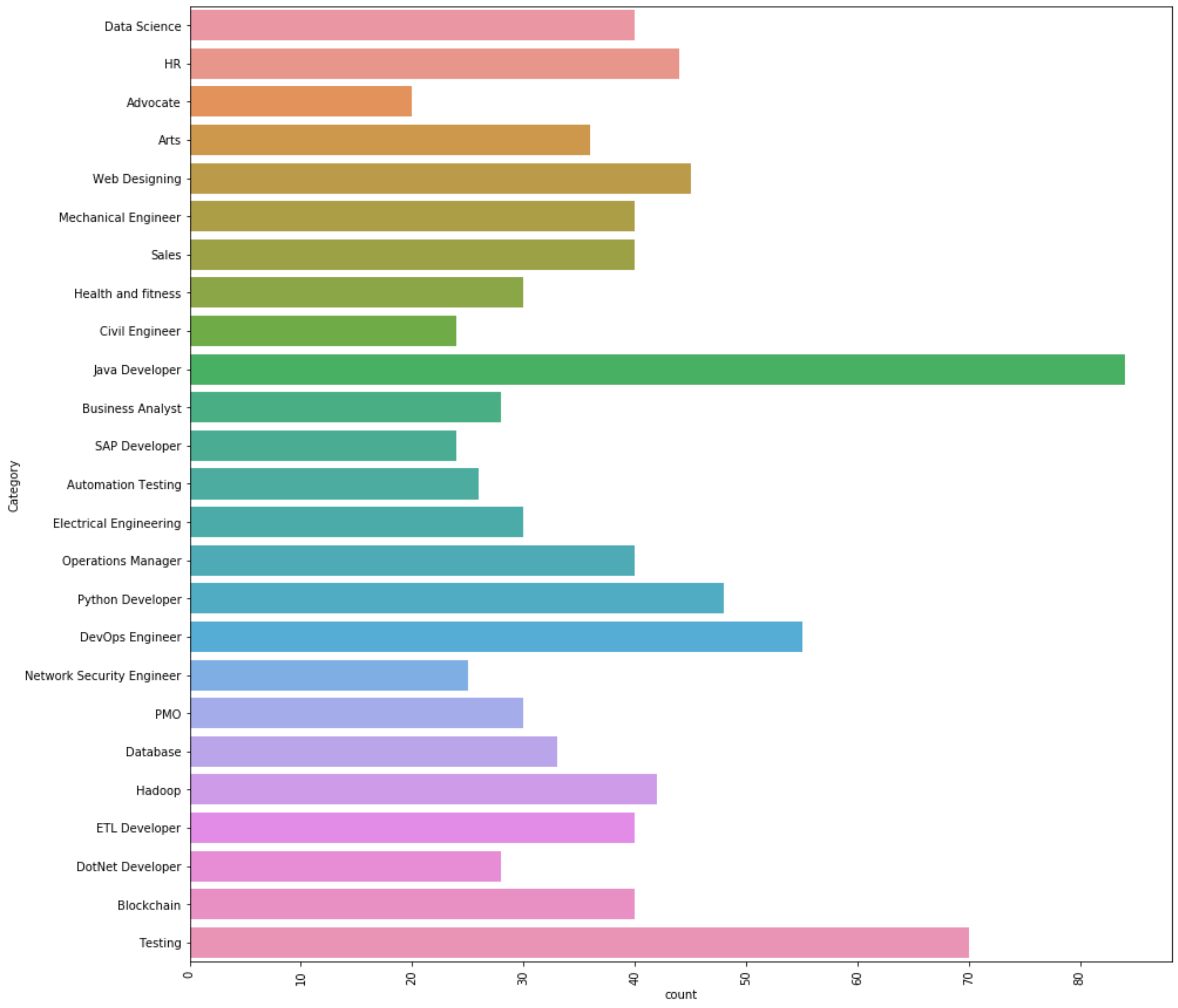


Figure 6.1(b). Data Collection (Train and Test Data)

6.2.1.Data Processing

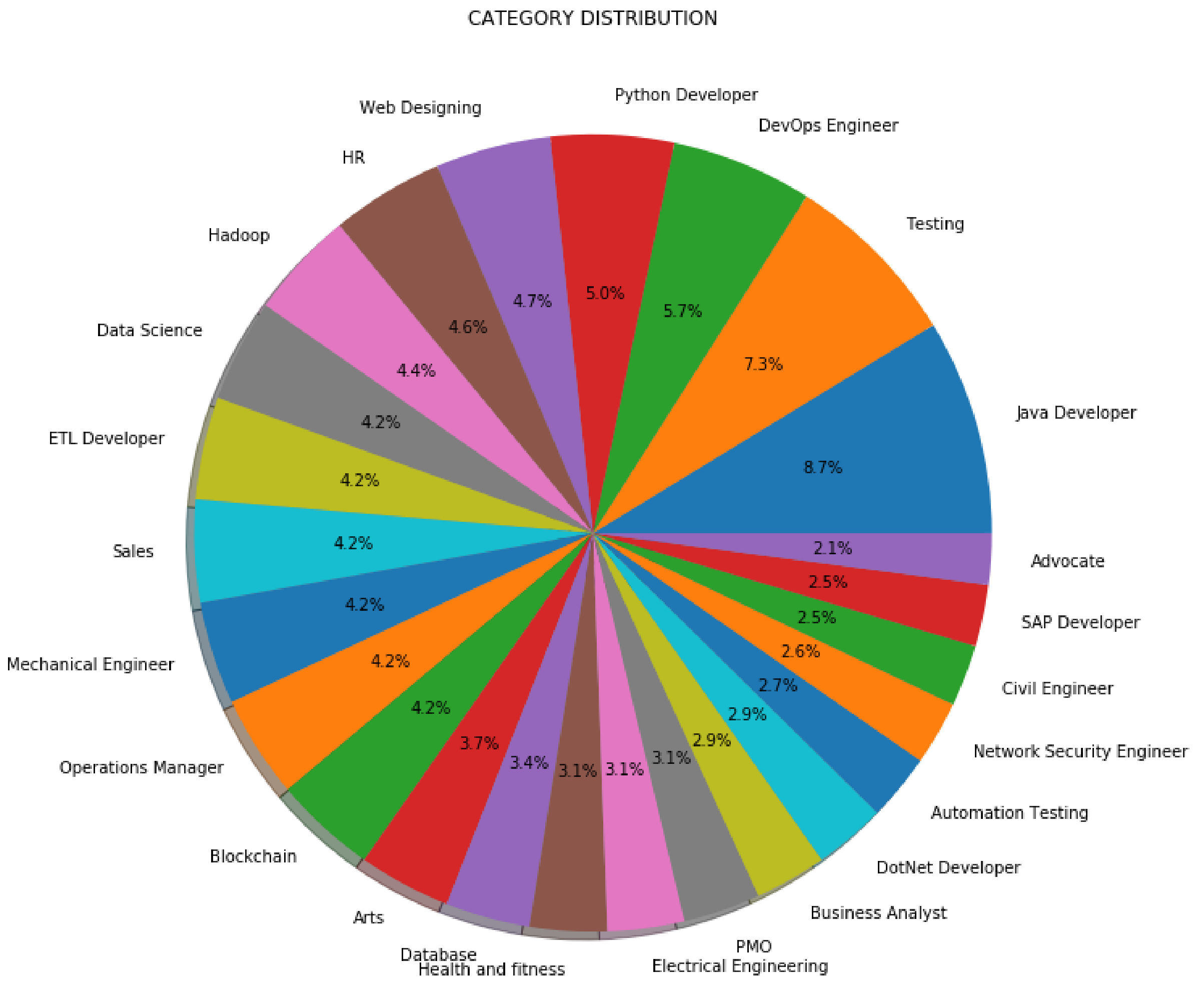


Figure 6.1(c). Data Collection (Train and Test Data)

Step 1: Clean the ‘Resume’ column

In this step, we remove any unnecessary information from resumes like URLs, hashtags, and special characters.

```
def cleanResume(resumeText)

resumeText = re.sub('httpS+s*', ' ', resumeText)  # remove URLs

resumeText = re.sub('RT|cc', ' ', resumeText)  # remove RT and cc
resumeText = re.sub('#S+', '', resumeText)  # remove hashtags
resumeText = re.sub('@S+', '  ', resumeText)  # remove mentions
resumeText = re.sub('[%s]' % re.escape("""!"#$%&'()*+,-./:;<=>?@[]^_`{|}~"""), ' ', resumeText)  # remove punctuations
resumeText = re.sub(r'[^x00-x7f]',r' ', resumeText)
resumeText = re.sub('s+', ' ', resumeText)  # remove extra whitespace
return resumeText
resumeDataSet['cleaned_resume'] = resumeDataSet.Resume.apply(lambda x: cleanResume(x))
```

Step 2: Encoding 'Category'

Now, we will encode the 'Category' column using LabelEncoding. Even though the 'Category' column is 'Nominal' data we are using LabelEncong because the 'Category' column is our 'target' column. By performing LabelEncoding each category will become a class and we will be building a multiclass classification model.

```
var_mod = ['Category']

le = LabelEncoder()

for i in var_mod:

resumeDataSet[i] = le.fit_transform(resumeDataSet[i])

resumeText = re.sub(r'[^x00-x7f]',r' ', resumeText)

resumeText = re.sub('s+', ' ', resumeText)  # remove extra whitespace

return resumeText

resumeDataSet['cleaned_resume'] = resumeDataSet.Resume.apply(lambda)
```

Step 3: Preprocessing ‘cleaned_resume’ column

Here we will preprocess and convert the ‘cleaned_resume’ column into vectors. There are many ways to do that like ‘Bag of Words’, ‘Tf-Idf’, ‘Word2Vec’ and a combination of these methods.

We will be using the ‘Tf-Idf’ method to get the vectors in this approach.

```
requiredText = resumeDataSet['cleaned_resume'].values

requiredTarget = resumeDataSet['Category'].values

word_vectorizer = TfidfVectorizer(

sublinear_tf=True,

stop_words='english',

max_features=1500)

word_vectorizer.fit(requiredText)

WordFeatures = word_vectorizer.transform(requiredText)
```

We have ‘WordFeatures’ as vectors and ‘requiredTarget’ and target after this step.

6.2.2. Model Building

We will be using the ‘One vs Rest’ method with ‘KNeighborsClassifier’ to build this multiclass classification model.

We will use 80% data for training and 20% data for validation. Let’s split the data now into training and test set.

```
X_train,X_test,y_train,y_test = test_size=0.2)

print(X_train.shape)

print(X_test.shape).
```

6.2.3. Data Training

The Data training is done using the NLP

NLP Code for Data Training:

```
!pip install --user -U nltk
import matplotlib.pyplot as plt
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import numpy as np
import pandas as pd
import pickle
import re
import seaborn as sns
from sklearn import metrics
from sklearn.metrics import accuracy_score
from sklearn.utils import shuffle
import string
import tensorflow as tf
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import Tokenizer
import warnings
warnings.filterwarnings('ignore')
np.set_printoptions(precision=4)
nltk.download('stopwords')
nltk.download('punkt')
# Load dataset
Data=pd.read_csv('C:\Users\kirsh\OneDrive\Desktop\Project\resumeproj\ass
 ets\data\UpdatedResumeDataSet.csv', engine='python')
data.head()
# Print unique categories of resumes
print(data['Category'].value_counts())
# Drop rows where category is "Testing" and store new size of dataset
data = data[data.Category != 'Testing']
data_size = len(data)
# Bar graph visualization
plt.figure(figsize=(15,15))
plt.xticks(rotation=90)
sns.countplot(y="Category", data=data)
# Get set of stopwords
stopwords_set = set(stopwords.words('english')+['``',"''"])
def clean_text(resume_text):
    resume_text = re.sub('http\S+\s*', ' ', resume_text)  # remove URLs
    resume_text = re.sub('RT|cc', ' ', resume_text)  # remove RT and cc
    resume_text = re.sub('#\S+', '', resume_text)  # remove hashtags
    resume_text = re.sub('@\S+', '  ', resume_text)  # remove mentions
    resume_text = re.sub(r'[^\x00-\x7f]',r' ', resume_text)
    resume_text = re.sub('\s+', ' ', resume_text)  # remove extra whitespace
    resume_text = resume_text.lower()  # convert to lowercase
    resume_text_tokens = word_tokenize(resume_text)  # tokenize
    filtered_text = [w for w in resume_text_tokens if not w in stopwords_set]
    return ' '.join(filtered_text)
print('--- Original resume ---')
print(data['Resume'][0])
data['cleaned_resume'] = data.Resume.apply(lambda x: clean_text(x))
print('--- Cleaned resume ---')
print(data['cleaned_resume'][0])
features = data['cleaned_resume'].values
original_labels = data['Category'].values
labels = original_labels[:]
for i in range(data_size):
  labels[i] = str(labels[i].lower())
  labels[i] = labels[i].replace(" ", "")
features, labels = shuffle(features, labels)
print(features[0])
print(labels[0])
train_split = 0.8
train_size = int(train_split * data_size)
train_features = features[:train_size]
train_labels = labels[:train_size]
test_features = features[train_size:]
test_labels = labels[train_size:]
print(len(train_labels))
print(len(test_labels))

vocab_size = 6000

oov_tok = '<OOV>'

feature_tokenizer = Tokenizer(num_words=vocab_size, oov_token=oov_tok)

feature_tokenizer.fit_on_texts(features)

feature_index = feature_tokenizer.word_index

print(dict(list(feature_index.items())))

train_feature_sequencesfeature_tokenizer.texts_to_sequences(train_features)

print(train_feature_sequences[0])

test_feature_sequences = feature_tokenizer.texts_to_sequences(test_features)

print(test_feature_sequences[0])

label_tokenizer = Tokenizer(lower=True)

label_tokenizer.fit_on_texts(labels)

label_index = label_tokenizer.word_index

print(dict(list(label_index.items())))

train_label_sequences = label_tokenizer.texts_to_sequences(train_labels)

print(train_label_sequences[0])

test_label_sequences = label_tokenizer.texts_to_sequences(test_labels)

print(test_label_sequences[0])

max_length = 300

trunc_type = 'post'

pad_type = 'post'

train_feature_paddedpad_sequences(train_feature_sequences,maxlen=max)

test_feature_paddedpad_sequences(test_feature_sequences)
```

```
print(train_feature_padded[0])
print(test_feature_padded[0])
embedding_dim = 64
model = tf.keras.Sequential([
tf.keras.layers.Embedding(vocab_size, embedding_dim, input_length=1),
  tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(embedding_dim))
tf.keras.layers.Dense(embedding_dim, activation='relu'),
tf.keras.layers.Dense(25, activation='softmax')
])
model.summary()
model.compile(loss='sparse_optimizer='adam', metrics=['accuracy'])
train_feature_padded = np.array(train_feature_padded)
test_feature_padded = np.array(test_feature_padded)
train_label_sequences = np.array(train_label_sequences)
test_label_sequences = np.array(test_label_sequences)
print(encoding_to_label[np.argmax(prediction[0])])
```

6.2.4. HR & Candidate Login Page

HR login page is an essential component of the proposed system, providing access and authentication for HR professionals to use the system's functionalities. Here are some key features and considerations for the HR login page:

1. **Secure Authentication:** The login page should ensure secure authentication by implementing strong password requirements, encryption protocols, and measures to prevent unauthorized access. This ensures that only authorized HR professionals can access the system.
2. **User-Friendly Interface:** The login page should have a user-friendly interface that is easy to navigate and understand. It should include clear instructions on

how to log in, a username and password input field, and an option for password recovery or account registration if necessary.

3. **Integration with HR Database:** The login page should be connected to the HR database or user management system to authenticate the HR professionals' credentials. This integration ensures that only valid and authorized users can access the system.
4. **Multi-Factor Authentication (MFA):** For added security, the login page can implement multi-factor authentication. MFA requires HR professionals to provide an additional authentication factor, such as a verification code sent to their registered email or a one-time password generated by an authentication app.
5. **Responsive Design:** The login page should have a responsive design that adapts to different screen sizes and devices, ensuring accessibility for HR professionals who may need to log in from various devices, including desktop computers, laptops, or mobile devices.

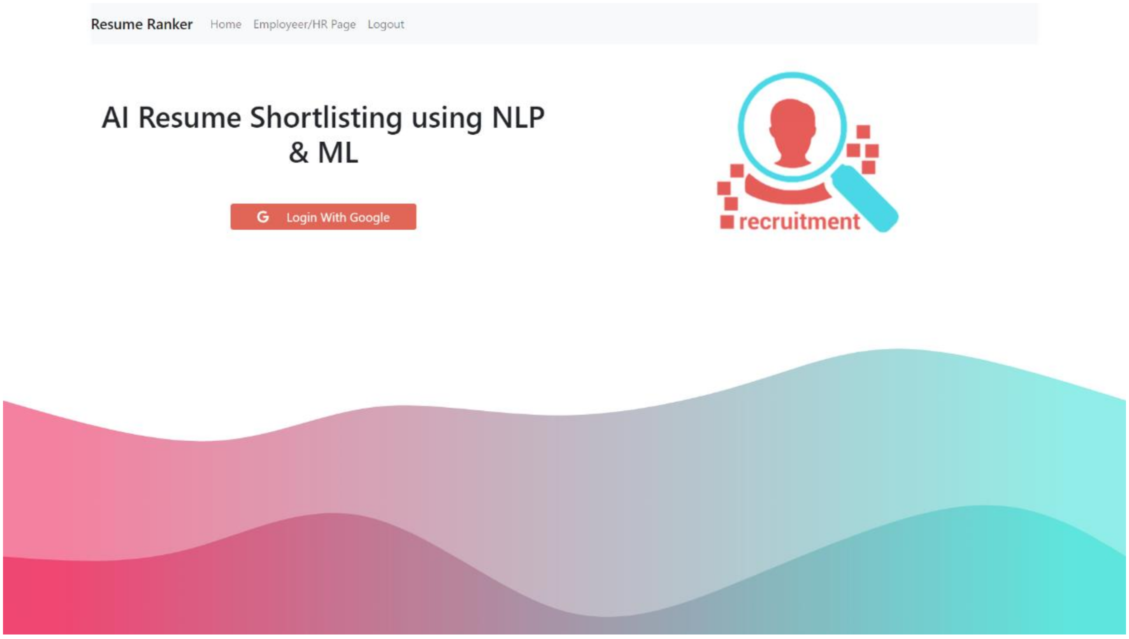


Figure 6.3 HR Login Page

CHAPTER 7
SYSTEM TESTING

7.1 TEST PLAN

The purpose of this test plan is to describe the approach and procedures that will be used to analyse and select the resumes of the candidate by their skills for the specific jobs.

Functional Testing:

- Testing that the system login page works and the Google Authentication System.
- Testing that the system Correctly uploads the resume to the Database.
- Testing that the system can analyse the Resume using the NLP.
- Testing that the system allows the HR to post a Job.

Performance Testing:

- Testing system response time under different load conditions.
- Testing system time for uploads the Resume.
- Testing that the time it takes for analysing the Resume.

Regression Testing:

- Test the system after changes or updates have been made to ensure that existing features still function as expected.
- Test to ensures that the application remains stable and reliable after changes have been made.
- Test to reducing the risk of unexpected behavior or failures in the application.
- Test to improve the quality of the application by ensuring that existing functionalities works correctly.

7.2 TEST CASES

This is based on the training and testing the model we used in this application. This eligibility can be checked by using the details entered by the users.

Testing system's existing functionality to ensure that it has not been affected by any changes.

7.2.1. Test Cases in this Recommendation Application:

Test Case 1 **:** Functional Testing – Verify that the system correctly authenticates The Candidate and the HR.

Input **:** Login using Google Account in HR/Candidate Login page.

Expected Output **:** Logins the HR/Candidate to the portal.

Actual Output **:** Logins to the Portal.

Test Case 2 **:** Performance Testing – Checking the Time for Resume uploading

Input **:** A Resume was uploaded through the Candidate Upload page.

Expected Output **:** The Upload must complete and the Buffering of the upload must be low.

Actual Output **:** The Resume was seccuessfully uploaded and the Uploaded time was 1.3 seconds.

Test Case 3 : Accuracy Testing – Test the accuracy of the predictions Of the NLP.

Input : Submit an Abstract and apply for a job in the job portal.

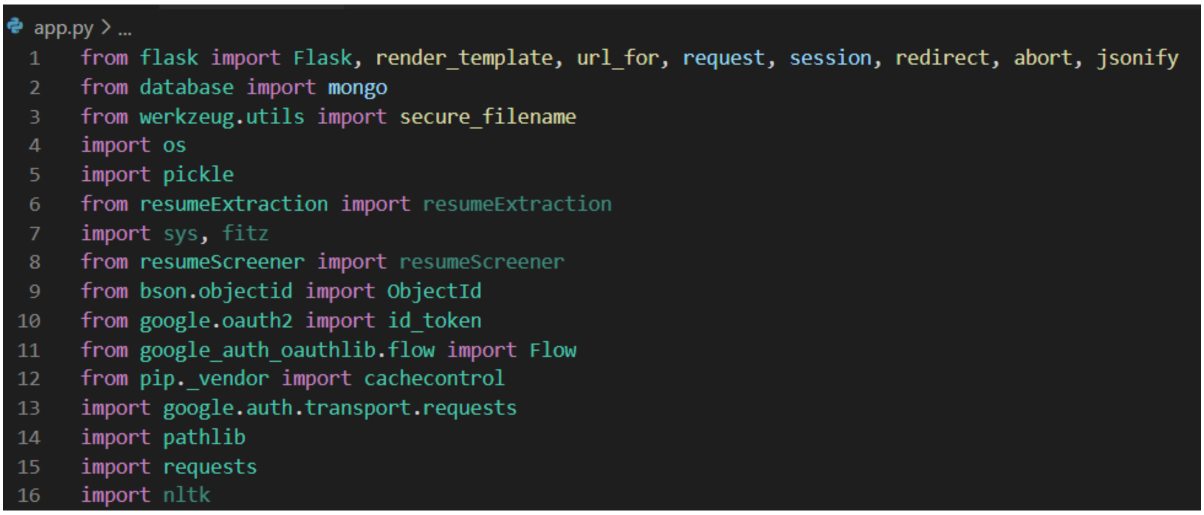
Expected Output : The data of the resume must be perfectly screened and the trained model will provide the percentage of the Resume matching with the Job Description.

Actual Output : The predictions generated by NLP was Accurate.

CHAPTER 8
SIMULATION RESULTS

8.1 DATA COLLECTION AND PACKAGE IMPORTING

The data collection process for this project involved gathering information from various sources, including public datasets, skill testing, and job description data. We collected data on various factors such as skills that is required for a job. The simulation results of this project indicate that the implemented resume shortling system has a high accuracy rate in predicting the appropriate resumes for a given jobs based on factors such as job description. The system also showed a high accuracy rate in suggesting appropriate Resumes based on skill levels.



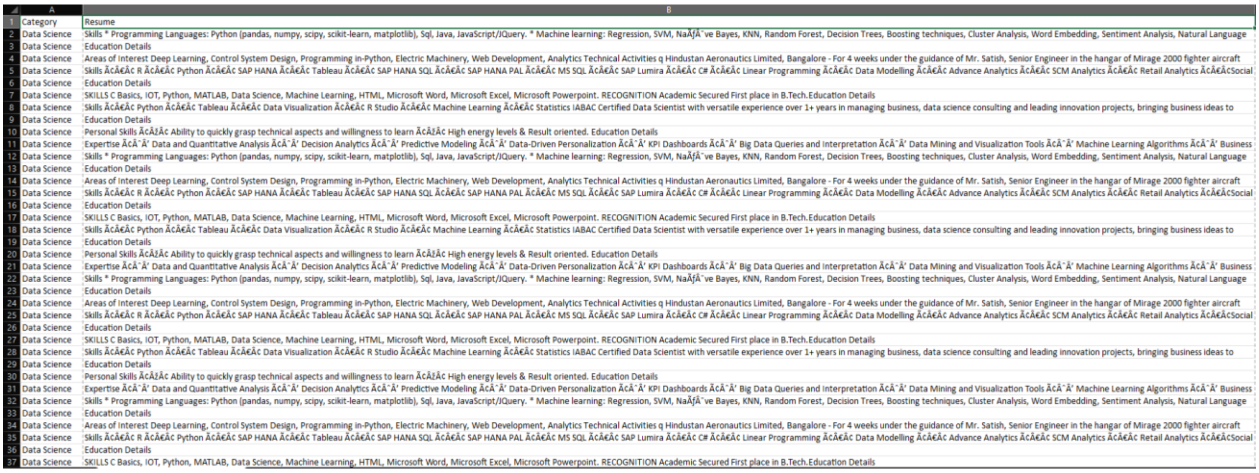
```python
from flask import Flask, render_template, url_for, request, session, redirect, abort, jsonify
from database import mongo
from werkzeug.utils import secure_filename
import os
import pickle
from resumeExtraction import resumeExtraction
import sys, fitz
from resumeScreener import resumeScreener
from bson.objectid import ObjectId
from google.oauth2 import id_token
from google_auth_oauthlib.flow import Flow
from pip._vendor import cachecontrol
import google.auth.transport.requests
import pathlib
import requests
import nltk
```

Figure 8.1 Importing Essential Libraries

Additionally, we also ensured compliance with relevant data privacy packages and security regulations to protect the confidentiality and integrity of the collected data. Overall, the data collection process was crucial in building a robust and reliable crop recommendation system that can accurately assess soil health, climate conditions, and other relevant factors to provide personalized and effective resume recommendations.

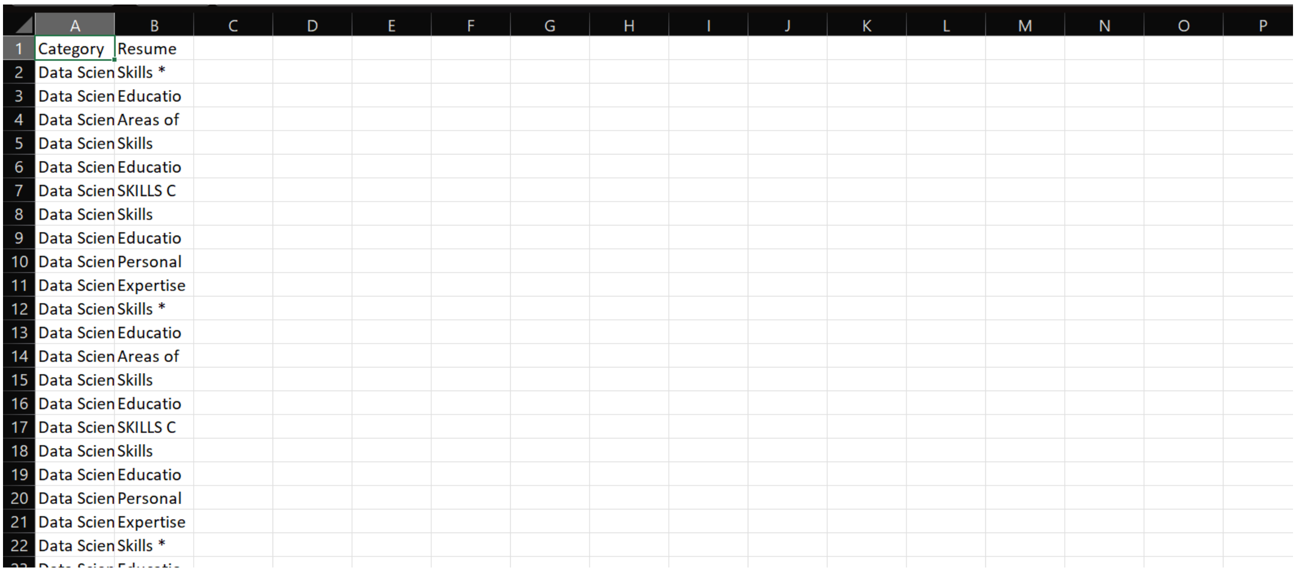
8.2 DATA STORED

The data has been collected from the internet sources it has give with the Jobs and the Skills required. Either it has any updation can need to be changed in the file.



	A	B
1	Category	Resume
2	Data Science	Skills * Programming Languages: Python (pandas, numpy, scipy, scikit-learn, matplotlib), Sql, Java, JavaScript/JQuery. * Machine learning: Regression, SVM, NaÃƒÂ¯ve Bayes, KNN, Random Forest, Decision Trees, Boosting techniques, Cluster Analysis, Word Embedding, Sentiment Analysis, Natural Language
3	Data Science	Education Details
4	Data Science	Areas of Interest Deep Learning, Control System Design, Programming in-Python, Electric Machinery, Web Development, Analytics Technical Activities q Hindustan Aeronautics Limited, Bangalore - For 4 weeks under the guidance of Mr. Satish, Senior Engineer in the hangar of Mirage 2000 fighter aircraft
5	Data Science	Skills Ã¢Â€Â¢ R Ã¢Â€Â¢ Python Ã¢Â€Â¢ SAP HANA Ã¢Â€Â¢ Tableau Ã¢Â€Â¢ SAP HANA SQL Ã¢Â€Â¢ SAP HANA PAL Ã¢Â€Â¢ MS SQL Ã¢Â€Â¢ SAP Lumira Ã¢Â€Â¢ C# Ã¢Â€Â¢ Linear Programming Ã¢Â€Â¢ Data Modelling Ã¢Â€Â¢ Advance Analytics Ã¢Â€Â¢ SCM Analytics Ã¢Â€Â¢ Retail Analytics Ã¢Â€Â¢Social
6	Data Science	Education Details
7	Data Science	SKILLS C Basics, IOT, Python, MATLAB, Data Science, Machine Learning, HTML, Microsoft Word, Microsoft Excel, Microsoft Powerpoint. RECOGNITION Academic Secured First place in B.Tech.Education Details
8	Data Science	Skills Ã¢Â€Â¢ Python Ã¢Â€Â¢ Tableau Ã¢Â€Â¢ Data Visualization Ã¢Â€Â¢ R Studio Ã¢Â€Â¢ Machine Learning Ã¢Â€Â¢ Statistics IABAC Certified Data Scientist with versatile experience over 1+ years in managing business, data science consulting and leading innovation projects, bringing business ideas to
9	Data Science	Education Details
10	Data Science	Personal Skills Ã¢ÂžÂ¢ Ability to quickly grasp technical aspects and willingness to learn Ã¢ÂžÂ¢ High energy levels & Result oriented. Education Details
11	Data Science	Expertise Ã¢ÂˆÂ’ Data and Quantitative Analysis Ã¢ÂˆÂ’ Decision Analytics Ã¢ÂˆÂ’ Predictive Modeling Ã¢ÂˆÂ’ Data-Driven Personalization Ã¢ÂˆÂ’ KPI Dashboards Ã¢ÂˆÂ’ Big Data Queries and Interpretation Ã¢ÂˆÂ’ Data Mining and Visualization Tools Ã¢ÂˆÂ’ Machine Learning Algorithms Ã¢ÂˆÂ’ Business
12	Data Science	Skills * Programming Languages: Python (pandas, numpy, scipy, scikit-learn, matplotlib), Sql, Java, JavaScript/JQuery. * Machine learning: Regression, SVM, NaÃƒÂ¯ve Bayes, KNN, Random Forest, Decision Trees, Boosting techniques, Cluster Analysis, Word Embedding, Sentiment Analysis, Natural Language
13	Data Science	Education Details
14	Data Science	Areas of Interest Deep Learning, Control System Design, Programming in-Python, Electric Machinery, Web Development, Analytics Technical Activities q Hindustan Aeronautics Limited, Bangalore - For 4 weeks under the guidance of Mr. Satish, Senior Engineer in the hangar of Mirage 2000 fighter aircraft
15	Data Science	Skills Ã¢Â€Â¢ R Ã¢Â€Â¢ Python Ã¢Â€Â¢ SAP HANA Ã¢Â€Â¢ Tableau Ã¢Â€Â¢ SAP HANA SQL Ã¢Â€Â¢ SAP HANA PAL Ã¢Â€Â¢ MS SQL Ã¢Â€Â¢ SAP Lumira Ã¢Â€Â¢ C# Ã¢Â€Â¢ Linear Programming Ã¢Â€Â¢ Data Modelling Ã¢Â€Â¢ Advance Analytics Ã¢Â€Â¢ SCM Analytics Ã¢Â€Â¢ Retail Analytics Ã¢Â€Â¢Social
16	Data Science	Education Details
17	Data Science	SKILLS C Basics, IOT, Python, MATLAB, Data Science, Machine Learning, HTML, Microsoft Word, Microsoft Excel, Microsoft Powerpoint. RECOGNITION Academic Secured First place in B.Tech.Education Details
18	Data Science	Skills Ã¢Â€Â¢ Python Ã¢Â€Â¢ Tableau Ã¢Â€Â¢ Data Visualization Ã¢Â€Â¢ R Studio Ã¢Â€Â¢ Machine Learning Ã¢Â€Â¢ Statistics IABAC Certified Data Scientist with versatile experience over 1+ years in managing business, data science consulting and leading innovation projects, bringing business ideas to
19	Data Science	Education Details
20	Data Science	Personal Skills Ã¢ÂžÂ¢ Ability to quickly grasp technical aspects and willingness to learn Ã¢ÂžÂ¢ High energy levels & Result oriented. Education Details
21	Data Science	Expertise Ã¢ÂˆÂ’ Data and Quantitative Analysis Ã¢ÂˆÂ’ Decision Analytics Ã¢ÂˆÂ’ Predictive Modeling Ã¢ÂˆÂ’ Data-Driven Personalization Ã¢ÂˆÂ’ KPI Dashboards Ã¢ÂˆÂ’ Big Data Queries and Interpretation Ã¢ÂˆÂ’ Data Mining and Visualization Tools Ã¢ÂˆÂ’ Machine Learning Algorithms Ã¢ÂˆÂ’ Business
22	Data Science	Skills * Programming Languages: Python (pandas, numpy, scipy, scikit-learn, matplotlib), Sql, Java, JavaScript/JQuery. * Machine learning: Regression, SVM, NaÃƒÂ¯ve Bayes, KNN, Random Forest, Decision Trees, Boosting techniques, Cluster Analysis, Word Embedding, Sentiment Analysis, Natural Language
23	Data Science	Education Details
24	Data Science	Areas of Interest Deep Learning, Control System Design, Programming in-Python, Electric Machinery, Web Development, Analytics Technical Activities q Hindustan Aeronautics Limited, Bangalore - For 4 weeks under the guidance of Mr. Satish, Senior Engineer in the hangar of Mirage 2000 fighter aircraft
25	Data Science	Skills Ã¢Â€Â¢ R Ã¢Â€Â¢ Python Ã¢Â€Â¢ SAP HANA Ã¢Â€Â¢ Tableau Ã¢Â€Â¢ SAP HANA SQL Ã¢Â€Â¢ SAP HANA PAL Ã¢Â€Â¢ MS SQL Ã¢Â€Â¢ SAP Lumira Ã¢Â€Â¢ C# Ã¢Â€Â¢ Linear Programming Ã¢Â€Â¢ Data Modelling Ã¢Â€Â¢ Advance Analytics Ã¢Â€Â¢ SCM Analytics Ã¢Â€Â¢ Retail Analytics Ã¢Â€Â¢Social
26	Data Science	Education Details
27	Data Science	SKILLS C Basics, IOT, Python, MATLAB, Data Science, Machine Learning, HTML, Microsoft Word, Microsoft Excel, Microsoft Powerpoint. RECOGNITION Academic Secured First place in B.Tech.Education Details
28	Data Science	Skills Ã¢Â€Â¢ Python Ã¢Â€Â¢ Tableau Ã¢Â€Â¢ Data Visualization Ã¢Â€Â¢ R Studio Ã¢Â€Â¢ Machine Learning Ã¢Â€Â¢ Statistics IABAC Certified Data Scientist with versatile experience over 1+ years in managing business, data science consulting and leading innovation projects, bringing business ideas to
29	Data Science	Education Details
30	Data Science	Personal Skills Ã¢ÂžÂ¢ Ability to quickly grasp technical aspects and willingness to learn Ã¢ÂžÂ¢ High energy levels & Result oriented. Education Details
31	Data Science	Expertise Ã¢ÂˆÂ’ Data and Quantitative Analysis Ã¢ÂˆÂ’ Decision Analytics Ã¢ÂˆÂ’ Predictive Modeling Ã¢ÂˆÂ’ Data-Driven Personalization Ã¢ÂˆÂ’ KPI Dashboards Ã¢ÂˆÂ’ Big Data Queries and Interpretation Ã¢ÂˆÂ’ Data Mining and Visualization Tools Ã¢ÂˆÂ’ Machine Learning Algorithms Ã¢ÂˆÂ’ Business
32	Data Science	Skills * Programming Languages: Python (pandas, numpy, scipy, scikit-learn, matplotlib), Sql, Java, JavaScript/JQuery. * Machine learning: Regression, SVM, NaÃƒÂ¯ve Bayes, KNN, Random Forest, Decision Trees, Boosting techniques, Cluster Analysis, Word Embedding, Sentiment Analysis, Natural Language
33	Data Science	Education Details
34	Data Science	Areas of Interest Deep Learning, Control System Design, Programming in-Python, Electric Machinery, Web Development, Analytics Technical Activities q Hindustan Aeronautics Limited, Bangalore - For 4 weeks under the guidance of Mr. Satish, Senior Engineer in the hangar of Mirage 2000 fighter aircraft
35	Data Science	Skills Ã¢Â€Â¢ R Ã¢Â€Â¢ Python Ã¢Â€Â¢ SAP HANA Ã¢Â€Â¢ Tableau Ã¢Â€Â¢ SAP HANA SQL Ã¢Â€Â¢ SAP HANA PAL Ã¢Â€Â¢ MS SQL Ã¢Â€Â¢ SAP Lumira Ã¢Â€Â¢ C# Ã¢Â€Â¢ Linear Programming Ã¢Â€Â¢ Data Modelling Ã¢Â€Â¢ Advance Analytics Ã¢Â€Â¢ SCM Analytics Ã¢Â€Â¢ Retail Analytics Ã¢Â€Â¢Social
36	Data Science	Education Details
37	Data Science	SKILLS C Basics, IOT, Python, MATLAB, Data Science, Machine Learning, HTML, Microsoft Word, Microsoft Excel, Microsoft Powerpoint. RECOGNITION Academic Secured First place in B.Tech.Education Details

Figure 8.2 Data Collection for Jobs with Skills



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Category	Resume														
2	Data Scien	Skills *														
3	Data Scien	Educatio														
4	Data Scien	Areas of														
5	Data Scien	Skills														
6	Data Scien	Educatio														
7	Data Scien	SKILLS C														
8	Data Scien	Skills														
9	Data Scien	Educatio														
10	Data Scien	Personal														
11	Data Scien	Expertise														
12	Data Scien	Skills *														
13	Data Scien	Educatio														
14	Data Scien	Areas of														
15	Data Scien	Skills														
16	Data Scien	Educatio														
17	Data Scien	SKILLS C														
18	Data Scien	Skills														
19	Data Scien	Educatio														
20	Data Scien	Personal														
21	Data Scien	Expertise														
22	Data Scien	Skills *														

Figure 8.3 Categorized Data

CHAPTER 9

APPENDICES

9.1 SAMPLE CODE

========================app.py=============================

```
from flask import Flask, render_template, url_for, request, session, redirect, abort,
jsonify
from database import mongo
from werkzeug.utils import secure_filename
import os
import pickle
from resumeExtraction import resumeExtraction
import sys, fitz
from resumeScreener import resumeScreener
from bson.objectid import ObjectId
from google.oauth2 import id_token
from google_auth_oauthlib.flow import Flow
from pip._vendor import cachecontrol
import google.auth.transport.requests
import pathlib
import requests
import nltk
def allowedExtension(filename):
    return '.' in filename and filename.rsplit('.',1)[1].lower() in ['docx','pdf']
app = Flask(__name__)
app.secret_key = "Resume_screening"
os.environ["OAUTHLIB_INSECURE_TRANSPORT"] = "1"
GOOGLE_CLIENT_ID = '488494246810-
```

```
b2lhd6g4etaupc1d57lu6o3o0fq04udd.apps.googleusercontent.com'
client_secrets_file = os.path.join(pathlib.Path(__file__).parent, "client_secret.json")
flow = Flow.from_client_secrets_file(
    client_secrets_file=client_secrets_file,
    scopes=["https://www.googleapis.com/auth/userinfo.profile",
"https://www.googleapis.com/auth/userinfo.email", "openid"],
    redirect_uri="http://127.0.0.1:5000/callback")
UPLOAD_FOLDER = 'static/resumes'
app.config['UPLOAD_FOLDER']=UPLOAD_FOLDER
app.config['MONGO_URI']=
'mongodb+srv://Sabari:Sabari@resumeextraction.fibwg7g.mongodb.net/test'
mongo.init_app(app)
resumeFetchedData = mongo.db.resumeFetchedData
Ranked_resume = mongo.db.Ranked_resume
IRS_USERS = mongo.db.IRS_USERS
JOBS = mongo.db.JOBS
from Job_post import job_post
app.register_blueprint(job_post,url_prefix="/HR1")
extractorObj = pickle.load(open("resumeExtractor.pkl","rb"))
screenerObj = pickle.load(open("resumeScreener.pkl","rb"))
@app.route('/')
def index():
    return render_template("index.html")
@app.route('/emp')
def emp():
    if 'user_id' in session and 'user_name' in session:
        return render_template("EmployeeDashboard.html")
    else:
        return render_template("index.html", errMsg="Login First")
@app.route('/login')
def login():
    authorization_url, state = flow.authorization_url()
    session["state"] = state
    return redirect(authorization_url)
@app.route("/callback")
def callback():
    flow.fetch_token(authorization_response=request.url)
    if not session["state"] == request.args["state"]:
        abort(500)
    credentials = flow.credentials
    request_session = requests.session()
    cached_session = cachecontrol.CacheControl(request_session)
    token_request = google.auth.transport.requests.Request(session=cached_session)
    id_info = id_token.verify_oauth2_token(
        id_token=credentials._id_token,
        request=token_request,
        audience=GOOGLE_CLIENT_ID
    )
    result = None
    result = IRS_USERS.find_one({"Email":id_info.get("email")},{"_id":1})
    if result == None:
        session['user_id'] =
str(IRS_USERS.insert_one({"Name":id_info.get("name"),"Email":id_info.get("email
"),"Google_id":id_info.get("sub")}).inserted_id)
```

```
        session['user_name'] = str(id_info.get("name"))
    else:
        session['user_id'] = str(result['_id'])
        session['user_name'] = str(id_info.get("name"))
    return redirect("/emp")
@app.route('/signup', methods=["POST"])
def signup():
    if request.method == 'POST':
        name = str(request.form.get('name'))
        email = str(request.form.get('email'))
        password = str(request.form.get('password'))
        status = None
        status = 
IRS_USERS.insert_one({"Name":name,"Email":email,"Password":password})
        if status == None:
            return render_template("index.html",errMsg="Problem in user creation check 
data or try after some time")
        else:
            return render_template("index.html",successMsg="User Created 
Successfully!")
@app.route("/logout")
def logout():
    session.pop('user_id',None)
    session.pop('user_name',None)
    return redirect(url_for("index"))
@app.route('/HR')
def HR():
    return render_template("CompanyDashboard.html")
@app.route('/test')
def test():
    return "Connection Successful"
@app.route("/uploadResume", methods=['POST'])
def uploadResume():
    if 'user_id' in session and 'user_name' in session:
        try:
            file = request.files['resume']
            filename = secure_filename(file.filename)
            print("Extension:",file.filename.rsplit('.',1)[1].lower())
            if file and allowedExtension(file.filename):
                temp =
resumeFetchedData.find_one({"UserId":ObjectId(session['user_id'])},{"ResumeTitle"
:1})
                if temp == None:
                    print("HELLO")
                else:
                    print("hello")
                    resumeFetchedData.delete_one({"UserId":ObjectId(session['user_id'])})
                    Ranked_resume.delete_one({"UserId":ObjectId(session['user_id'])})
        os.remove(os.path.join(app.config['UPLOAD_FOLDER'],temp['ResumeTitle']))
                file.save(os.path.join(app.config['UPLOAD_FOLDER'],filename))
fetchedData=extractorObj.extractorData("static/resumes/"+filename,file.filename.rspl
it('.',1)[1].lower())
                skillsPercentage = screenerObj.screenResume(fetchedData[5])
                result = result1 = None
                print("FetchedData:",fetchedData)
                result = resumeFetchedData.insert_one({"UserId":ObjectId(session['user_id']),"Name":fetchedData[0],"Mobile_no":fetchedData[1],"Email":fetchedData[2],"Skills":list(fetchedData[3]),"Education":fetchedData[4],"Appear":0,"ResumeTitle":filename,"ResumeData":fetchedData[5]})
                if result == None:
                    return render_template("EmployeeDashboard.html",errorMsg="Problem in Resume Data Storage")
                else:
                    result1 = Ranked_resume.insert_one({"UserId":ObjectId(session['user_id']),"Top_skills":dict(skillsPercentage)})
                    if result1 == None:
                        return render_template("EmployeeDashboard.html",errorMsg="Problem in Skills Data Storage")
                    else:
                        return render_template("EmployeeDashboard.html",successMsg="Resume Screen Successfully!!")
            else:
                return render_template("EmployeeDashboard.html",errorMsg="Document Type Not Allowed")
        except:
            print("Exception Occured")
    else:
        return render_template("index.html", errMsg="Login First")
```

```
@app.route('/viewdetails', methods=['POST', 'GET'])
def viewdetails():
   employee_id = request.form['employee_id']
   result = resumeFetchedData.find({"UserId":ObjectId(employee_id)})
   dt=result[0]
   name=dt['Name']
   email=dt['Email']
   mobile=dt['Mobile_no']
   skills=dt['Skills']
   education=dt['Education']
   return
jsonify({'name':name,'email':email,'mobile':mobile,'skills':skills,'education':education}
)
@app.route("/empSearch",methods=['POST'])
def empSearch():
   if request.method == 'POST':
      category = str(request.form.get('category'))
      TopEmployeers=None
TopEmployeers=Ranked_resume.find({"Top_skills."+category:{"$ne":None}},{"Top
_skills."+category:1,"UserId":1}).sort([("Top_skills."+category,-1)])
      if TopEmployeers == None:
         return render_template("CompanyDashboard.html",errorMsg="Problem in
Category Fetched")
      else:
         selectedResumes={}
         cnt = 0
```

```
            for i in TopEmployeers:
    se=IRS_USERS.find_one({"_id":ObjectId(i['UserId'])},{"Name":1,"Email":1,"_id":1
    })
                selectedResumes[cnt] =
    {"Name":se['Name'],"Email":se['Email'],"_id":se['_id']}
                se = None
                cnt += 1
            return render_template("CompanyDashboard.html",len =
    len(selectedResumes), data = selectedResumes)
    if __name__=="__main__":
     app.run(debug=True)
```

========================job_post.html=========================

```
{% extends "base.html" %}
{% block title %}Job POST{% endblock %}
{% block content %}
<div class="container-fluid">
    <div class="container">
      {% if errorMsg %}
        <h2 class='alert alert-danger' style='text-align: center;'>{{errorMsg}}</h2>
      {% endif %}
      {% if successMsg %}
        <h2 class='alert alert-success' style='text-align: center;'>{{successMsg}}</h2>
      {% endif %}
        <div class="d-grid gap-2 d-md-flex justify-content-md-end">
            <button type="button" class="btn btn-info me-md-5 mb-3" data-bs-
```

```
toggle="modal" data-bs-target="#jobpost"> + Post JOB</button>
    </div>
    <div class="row">
     {% for i in range(0,data|length) %}
     <div class="col-lg-4 col-md-6 col-xs-12">
       <div class="card mb-3" style="max-width: 540px;">
         <div class="row g-0">
            <div class="col-auto" style="margin:auto;width:60%;padding:10px;">
             <div class="card-body">
              <h5 class="card-title text-center">{{data[i]['Job_Profile']}}<span class="badge rounded-pill bg-warning text-dark">New</span></h5>
              <p class="card-text text-center"><strong>Company:</strong>{{data[i]['CompanyName']}}</p>
              <p class="card-text text-center"><strong>{{ data[i]['Salary'] }}</strong></p>
              <p class="card-text text-center"><small class="text-muted">{{data[i]['LastDate']}}</small></p>
              <p><button class="btn btn-outline-success view_candidates" id="{{ data[i]['job_id'] }}" data-bs-toggle="modal" data-bs-target="#view_candidates">View Applied Candidates</button></p>
             </div>
            </div>
         </div>
       </div>
     </div>
     {% endfor %}
```

```
      </div>
    </div>
</div>
<div class="modal fade" id="jobpost" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog modal-fullscreen-sm-down">
      <div class="modal-content">
        <div class="modal-header">
            <h5 class="modal-title" id="exampleModalLabel">Add JOB</h5>
            <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
          </div>
        <div class="modal-body">
          <div class="container">
              <div class="row">
                  <div class="col-12">
                      <form class="form-horizontal" action="/HR1/add_job" method="POST" enctype="multipart/form-data">
                        <div class="form-floating mb-3">
                            <input type="text" class="form-control" id="companyInput" placeholder="Enter Company Name" name='company' required>
                            <label for="companyInput">Company:</label>
                        </div>
                        <div class="form-floating mb-3">
                            <input type="text" class="form-control" id="jbInput"
```

```
placeholder="Enter Job Profile" name='jp' required>
              <label for="jbInput">Job Profile:</label>
            </div>
            <div class="form-floating mb-3">
             <input type="text" class="form-control" id="SalaryInput"
placeholder="Enter Package in Rs." name='salary' required>
             <label for="SalaryInput">Salary:</label>
           </div>
            <div class="form-floating mb-3">
             <input type="date" class="form-control" id="LastDateInput"
placeholder="Enter Last Date" name='last_date' required>
             <label for="LastDateInput">Last Date:</label>
            </div>
            <div class="form-group mb-3">
             <label for="jd">Upload Job Description:</label>
             <input type='file' class="form-control" id="jd" name='jd' required/>
            </div>
            <div>
              <input type='submit' class="btn btn-danger me-md-5 mb-3"
value='Submit'/>
            </div>
           </form>
         </div>
       </div>
     </div>
```

```
        </div>
      </div>
    </div>
  </div>
  <div class="modal fade" id="view_candidates" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog modal-fullscreen-sm-down">
      <div class="modal-content">
        <div class="modal-header">
            <h5 class="modal-title" id="exampleModalLabel">Applied Candidates</h5>
            <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
          </div>
        <div class="modal-body">
          <div class="container">
              <div class="row">
                  <div class="col-12">
                      <table class="table table-striped table-responsive">
                        <thead>
                          <tr>
                            <th class='text-center'>No.</th>
                            <th class='text-center'>Name</th>
                            <th class='text-center'>Matching %</th>
                          </tr>
                        </thead>
            <tbody id="data"></tbody>
           </table>
         </div>
       </div>
     </div>
    </div>
   </div>
  </div>
 </div>
 <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>
<script>var $j = jQuery.noConflict();</script>
<script>
  $(document).on('click', '.view_candidates', function(){
    var job_id = $(this).attr("id");
    $j.ajax({
      url:"/HR1/view_applied_candidates",
      method:"POST",
      data:({job_id:job_id}),
      success:function(data){
        if(data[1]==200)
        {
          var strline="";
          for(var i=0;i<data[0];i++)
```

```
                    {
                      strline+="<tr><td class='text-center'>"+(i+1)+"</td><td class='text-center'>"+data[i+2]['Name']+"</td><td class='text-center'>"+data[i+2]['Match']+"</td></tr>";
                    }
                    document.getElementById("data").innerHTML=strline;
                }
                if(data.StatusCode==400)
                {
                    alert("Failed");
                }
            }
        });
    });
</script>
{%  endblock %}
```

======================Index.html================================

```
{% extends "base.html" %}
{% block title %}Resume Ranking{% endblock %}
{% block Addnav %}
<li class="nav-item active">
    <a class="nav-link" href="/HR">Employeer/HR Page</a>
</li>
{% endblock %}
{% block styleInclude %}
<style>
    @media screen and (min-width: 992px) {
        a.sign_in {
            width:60%;
        }
    }
    @media screen and (min-width: 1300px) {
        a.sign_in {
            width:40%;
        }
    }
    @media screen and (min-width: 300px) {
        .main_card{
            margin-top:30px;
        }
    }

    .sign_in{
        border: none;
        border-radius: 4px;
        margin: 5px 0;
        opacity: 0.85;
        display: inline-block;
        font-size: 17px;
        line-height: 20px;
        text-decoration: none;
        background-color: #dd4b39;
        color: white;
    }
</style>
{% endblock %}
{% block content %}
```

```
<div class='container-fluid'>
    <div class="container">
    <div class='row'>
        <div class="col-md-6 col-xs-12 main_card" style="margin-top:50px;z-index:10;">
            <h1 class="text-center mt-4 hh">AI Resume Shortlisting using NLP & ML</h1>
            <a href="/login" class="btn btn-lg btn-danger mt-5 sign_in mx-auto d-block" style="z-index:10;"><i class="fab fa-google me-4"></i>Login With Google</a>
        </div>
        <div class="col-md-6 col-xs-12">
            <img class="img-fluid mx-auto d-block" src="{{ url_for('static',filename='images/re.png') }}"/>
        </div>
    </div>
</div>
<svg style="position:absolute;bottom:0;left:0;"  id="svg" viewBox="0 0 1440 500" xmlns="http://www.w3.org/2000/svg" class="transition duration-300 ease-in-out delay-150"><defs><linearGradient id="gradient"><stop offset="5%" stop-color="#eb144c88"></stop><stop offset="95%" stop-color="#32ded488"></stop></linearGradient></defs><path d="M 0,600 C 0,600 0,200 0,200 C 94.34615384615384,227.51538461538462 188.69230769230768,255.03076923076924 267,252 C 345.3076923076923,248.96923076923076 407.5769230769231,215.3923076923077 479,208 C 550.4230769230769,200.6076923076923 631,219.4 715,219 C 799,218.6 886.4230769230769,199.0076923076923 958,177 C 1029.576923076923,154.9923076923077 1085.3076923076924,130.56923076923076 1163,134 C 1240.6923076923076,137.43076923076924 1340.3461538461538,168.7153846153846 1440,200 C 1440,200 1440,600 1440,600
```

```
 Z" stroke="none" stroke-width="0" fill="url(#gradient)" class="transition-all
 duration-300 ease-in-out delay-150"></path><defs><linearGradient
 id="gradient"><stop offset="5%" stop-color="#eb144cff"></stop><stop
 offset="95%" stop-color="#32ded4ff"></stop></linearGradient></defs><path d="M
 0,600 C 0,600 0,400 0,400 C 76.46666666666667,405.8179487179487
 152.93333333333334,411.6358974358974 229,393 C
 305.06666666666666,374.3641025641026
 380.73333333333335,331.27435897435896 462,348 C
 543.2666666666667,364.72564102564104
 630.1333333333332,441.26666666666665 709,467 C
 787.8666666666668,492.73333333333335 858.7333333333333,467.6589743589744
 950,430 C 1041.2666666666667,392.3410256410256
 1152.9333333333334,342.0974358974359 1238,335 C
 1323.0666666666666,327.9025641025641
 1381.5333333333333,363.9512820512821 1440,400 C 1440,400 1440,600 1440,600
 Z" stroke="none" stroke-width="0" fill="url(#gradient)" class="transition-all
 duration-300 ease-in-out delay-150"></path></svg>
 </div>
{% endblock %}
```

===================Company_Dashboard.html====================

```
  {% extends "base.html" %}
 {% block title %}HR Dashboard{% endblock %}
 {% block Addnav %}
 <li class="nav-item active">
    <a class="nav-link" href="/HR1/post_job">Post JOB</a>
 </li>
 {% endblock %}
 {% block content %}
 <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
       <script
```

```
src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>
        <script>var $j = jQuery.noConflict();</script>
        <style>
            @media screen and (max-width: 560px) {
                #categoryInput{
                    width:100%;
                }
                input.sBto{
                    width:100%;
                    height:20%;
                    margin-bottom:20px;
                }
            }
            @media screen and (min-width: 560px ) {
                input.sBto{
                    width:60%;
                    margin-left: 20%;
                    height:30%;
                    margin-bottom:50px
                }
            }
        </style>
<div class="contain-fluid">
    <div class="container">
        <div class="row">
            <h2>{{errorMsg}}</h2>

            <div class='col-xs-12'>
                <form class="form-horizontal" action="/empSearch" method='POST'>
                    <div class='form-floating mb-3 categoryInput1'>
            <input type='text' name='category' id="categoryInput" class='form-control'/>
            <label for="categoryInput">Enter Category:</label>
          </div>
          <div class='form-group'>
            <input type='submit' class='btn btn-danger sBto'  name='submit' value='Search'/>
          </div>
        </form>
      </div>
    </div>
    <div class="row">
      <div class='col-xs-12  table-responsive-sm'>

      <table class="table table-bordered">
        <thead>
          <tr>
            <th class="text-center">No</th>
            <th class="text-center">Name</th>
            <th class="text-center">Email</th>
            <th class="text-center">Details</th>
          </tr>
        </thead>
        <tbody>
          {% for i in range(0,data|length) %}
          <tr>
            <td class="text-center">{{(i+1)}}</td>
            <td class="text-center">{{data[i]['Name']}}</td>
            <td class="text-center">{{data[i]['Email']}}</td>
            <td><input type="button" name="view" value="View"
```

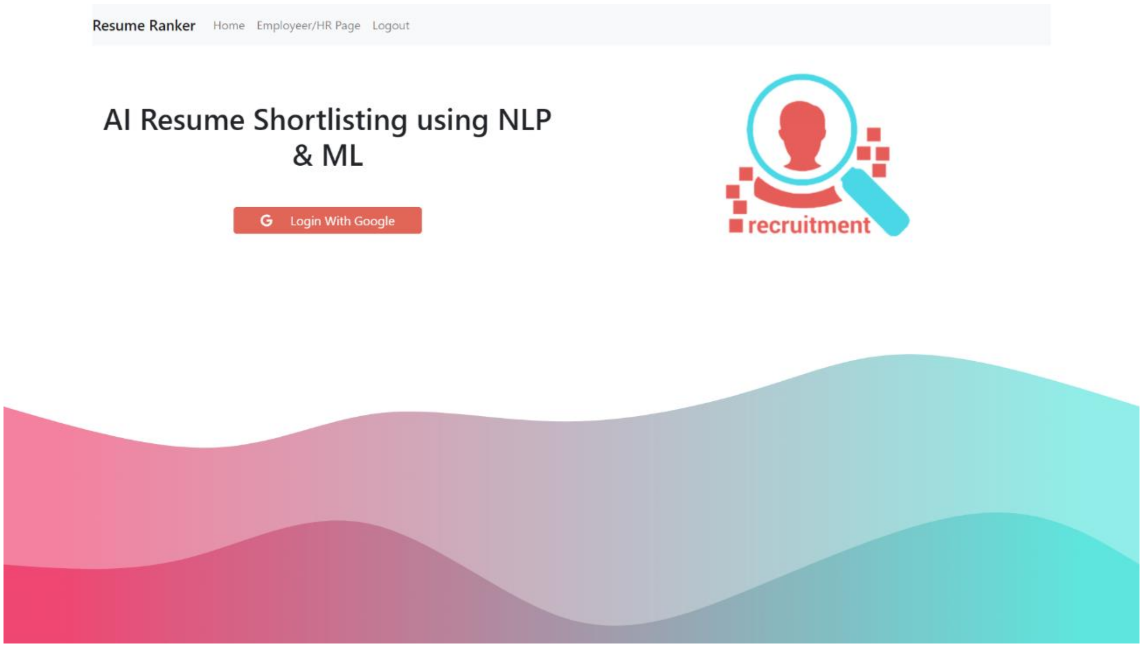
```
id="{{data[i]['_id']}}" data-bs-toggle="modal" data-bs-target="#viewdetails"
class="btn btn-secondary view_data" style="margin-left:10%;" /></td>
                    </tr>
                    {% endfor %}
                </tbody>
            </table>

            </div>
        </div>
    </div>
</div>
<div class="modal fade" id="viewdetails" tabindex="-1" role="dialog" aria-
labelledby="viewdetails" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered modal-fullscreen-sm-down"
role="document">
        <div class="modal-content">
        <div class="modal-header">
            <h3 class="modal-title" id="exampleModalLongTitle">Details of
Candidate:</h3>
            <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
        </div>
        <div class="modal-body">
        <table class="table table-responsive">
            <br>
            <tr> <label>Name:</label>   <input type="text" class="form-control-
plaintext" readonly  id="view_name"> </tr><br>
            <tr> <label>Email:</label>   <input type="text" class="form-control-
plaintext" readonly id="view_email">  </tr><br>
            <tr> <label>Mobile No.:</label>   <input type="text" class="form-control-
```

```
plaintext" readonly id="view_mobile">  </tr><br>
      <tr> <label>Skills:</label>   <textarea rows="4" cols="50" class="form-control-plaintext" readonly id="view_skills" > </textarea> </tr><br>
      <tr> <label>Education:</label>   <input type="text" class="form-control-plaintext" readonly  id="view_education">  </tr><br>
    </table>
    </div>
    <div class="modal-footer">
      <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
    </div>
    </div>
  </div>
  </div>
<script>
  $(document).on('click', '.view_data', function(){
  var employee_id = $(this).attr("id");
  $j.ajax({
  url:"/viewdetails",
  method:"POST",
  data:({employee_id:employee_id}),
  success:function(data){
    $('#view_name').val(data.name);
    $('#view_email').val(data.email);
    $('#view_mobile').val(data.mobile);
    $('#view_skills').val(data.skills);
    $('#view_education').val(data.education);
    }
  });
  });{% endblock %}
```

9.1 SCREENSHOTS

9.1.1 Home Page

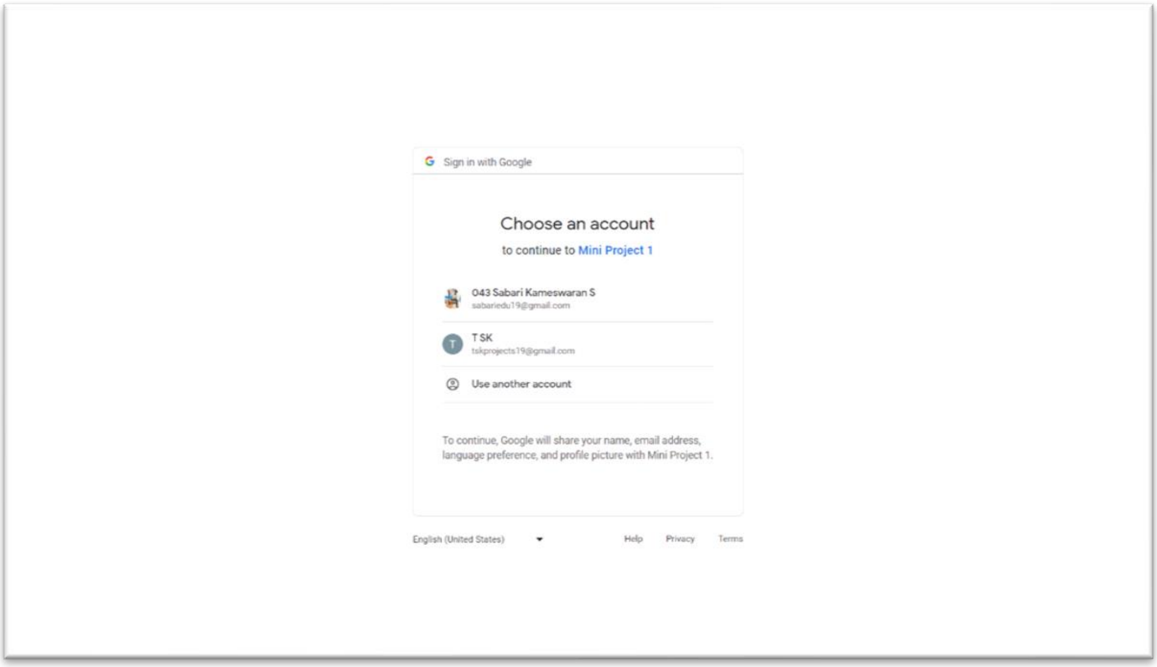


AI Resume Shortlisting using NLP & ML

Figure 9.1 Home Page

The home page for the resume shortlisting system is the login page for the HR/Candidate. It designed with the language of HTML, CSS and JS.

9.1.2 Login Page



Choose an account

to continue to Mini Project 1

043 Sabari Kameswaran S
sabariedu19@gmail.com

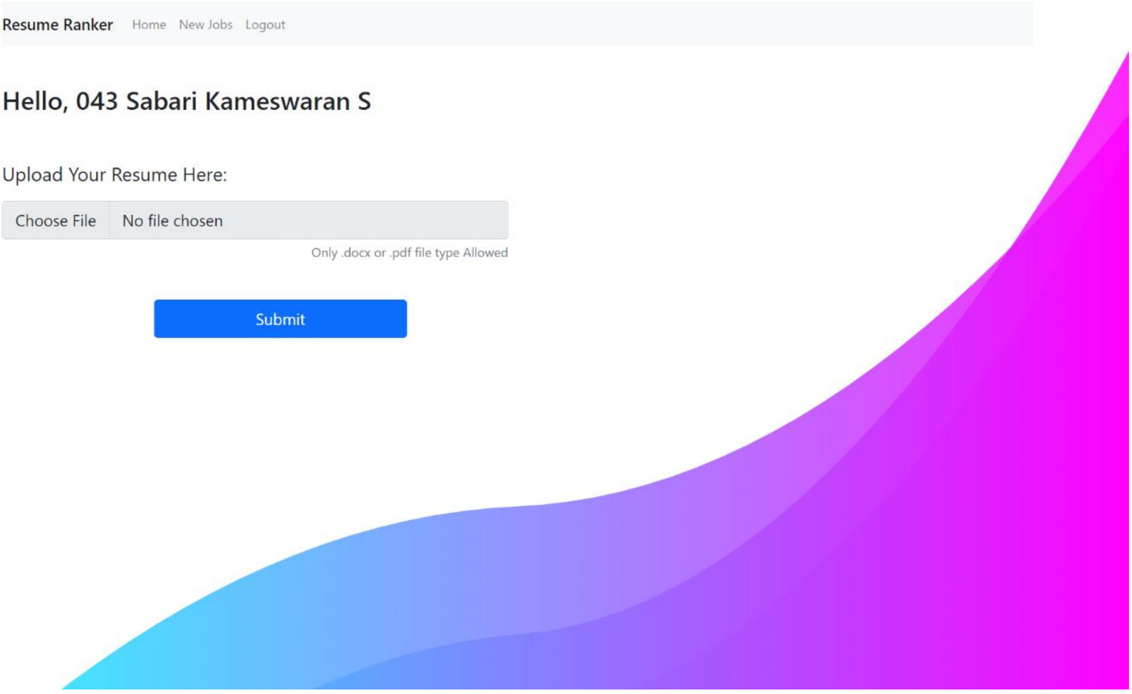
T SK
tskprojects19@gmail.com

Use another account

To continue, Google will share your name, email address, language preference, and profile picture with Mini Project 1.

Figure 9.2 HR/Candidate Login Page

9.1.3 Resume Uploading Page



Hello, 043 Sabari Kameswaran S

Upload Your Resume Here:

Figure **9.3. Resume Uploading Page**

The Resume of the candidate will be uploaded here after logging in with candidate login.

9.1.4 HR Job Posting

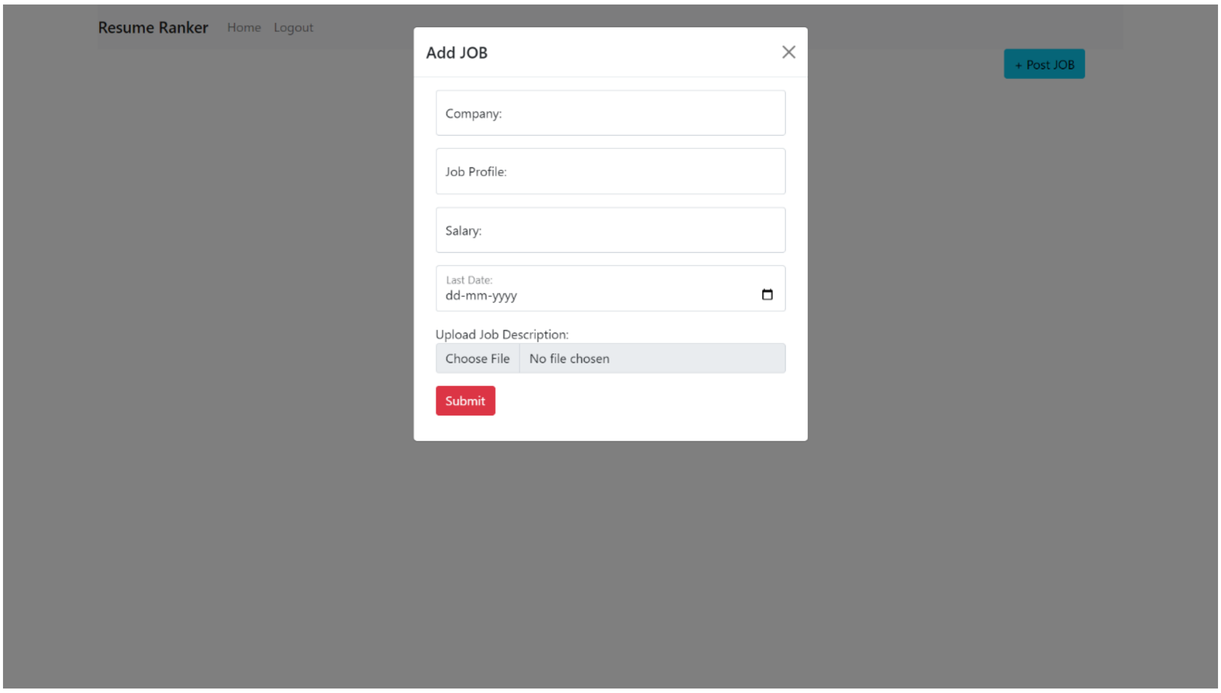


Figure 9.4. Job Posting

The HR will Post the Job with the job description and all the details needed.

9.1.5 Applying for the Job

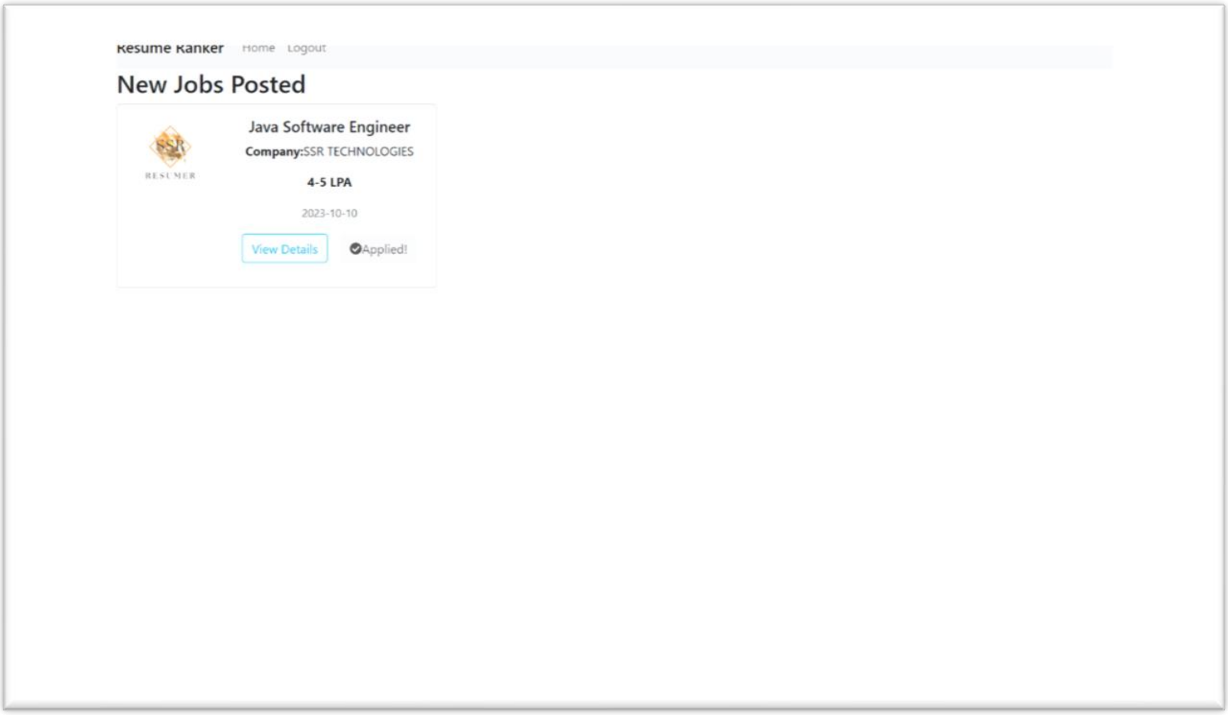
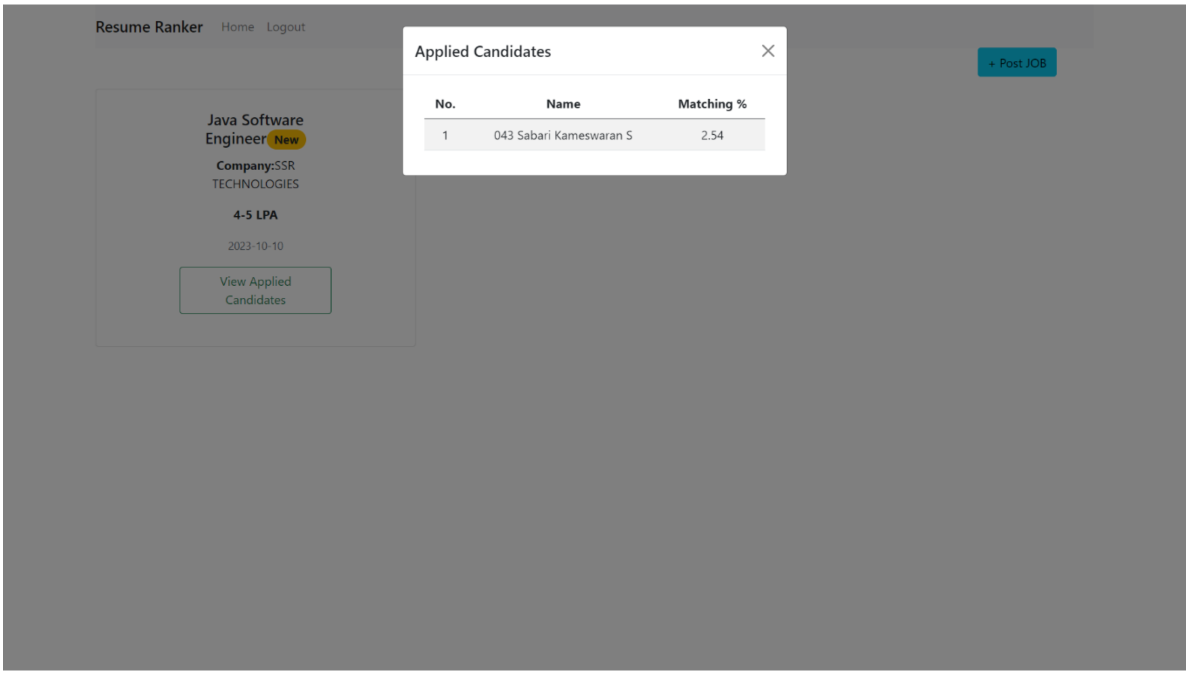


Figure 9.5. Applying for the Job

The Candidate applies for the job after uploading their Resumes.

9.1.6 Output for the NLP and the Matching Percentage



No.	Name	Matching %
1	043 Sabari Kameswaran S	2.54

Figure 9.6 Output of the NLP

It shows the main output of the project with the matching percentage.

CHAPTER 10

CONCLUSION AND FUTURE ENHANCEMENT

10.1 CONCLUSION

In conclusion, the proposed AI-based resume screening system using NLP techniques presents a promising solution to the limitations of the manual resume screening process. By leveraging advanced technologies, the system aims to streamline the hiring process, improve efficiency, and enhance the accuracy of candidate selection.

The project addresses the challenges faced by HR departments, such as the overwhelming volume of resumes received, the time-consuming nature of manual screening, and the potential for human biases and errors. The proposed system automates the screening process, utilizing NLP techniques to extract relevant information from resumes, match candidate skills with job requirements, and generate shortlists of qualified candidates.

The system offers several advantages, including reduced time and effort spent on screening, enhanced objectivity in evaluating resumes, improved efficiency in candidate selection, and a user-friendly web interface for HR professionals. The integration of data visualization tools further aids in interpreting and presenting the analysis results effectively.

It is important to note that the success of the system relies on the quality and completeness of the resume data provided by candidates. Additionally, continuous updates and maintenance of the skillset database are necessary to ensure accurate matching with job requirements. Nonetheless, the proposed system holds great potential in revolutionizing the resume screening process, empowering HR professionals to make informed decisions and ultimately facilitating the hiring of the best-suited professionals for specific positions.

10.2 FUTURE ENHANCEMENT

Some potential future enhancements for this AI-based resume screening system using NLP techniques include:

1. **Intelligent Feedback Mechanism:** Implementing an intelligent feedback mechanism where HR professionals can provide feedback on the system's shortlisting decisions. This feedback can be used to continuously improve the algorithm's performance and make it more accurate over time.
2. **Deep Learning Models:** Exploring the use of deep learning models, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), to extract more intricate features from resumes and improve the system's ability to identify relevant skills and qualifications.
3. **Contextual Understanding:** Enhancing the system's contextual understanding by considering factors such as industry-specific jargon, domain expertise, and job-specific requirements. This can help in better matching candidates to specific job positions.
4. **Sentiment Analysis:** Incorporating sentiment analysis techniques to analyze the tone and sentiment expressed in the resumes. This can provide insights into a candidate's attitude, enthusiasm, and overall fit within the company culture.
5. **Integration with External APIs:** Integrating the system with external APIs, such as job portals or professional networking platforms, to access additional data sources and enrich the analysis. This can include gathering data on candidate recommendations, endorsements, or public projects.

By implementing these future enhancements, the AI-based resume screening system can become more intelligent, accurate, and adaptable, ultimately improving the efficiency and effectiveness of the hiring process for HR professionals.

REFERENCES

1. Jurczyk, T., & Yang, D. (2019). Resume Screening Using Natural Language Processing. In 2019 IEEE International Conference on Big Data (Big Data) (pp. 3223-3229). IEEE. DOI: 10.1109/BigData47090.2019.9006274
2. Saini, A., & Singla, A. (2020). AI-Based Resume Screening System using NLP Techniques. In 2020 IEEE 8th International Conference on Advanced Computing (IACC) (pp. 587-592). IEEE. DOI: 10.1109/IACC49677.2020.9181013
3. Singh, A., & Bharti, M. (2020). Implementation of a Resume Screening System using NLP. In 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) (pp. 262-266). IEEE. DOI: 10.1109/ICIMIA48442.2020.9124252
4. Mishra, S., & Reddy, A. S. (2020). Automated Resume Screening using Machine Learning and Natural Language Processing. International Journal of Advanced Science and Technology, 29(6), 2887-2897.
5. Gu, Y., & Zhang, X. (2020). Automated Resume Screening and Job Recommendation System using NLP and Machine Learning. In 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE) (pp. 111-115). IEEE. DOI: 10.1109/ICAICE48611.2020.00026
6. Kumar, A., & Bansal, V. (2019). Artificial Intelligence Based Resume Screening. In 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU) (pp. 1-4). IEEE. DOI: 10.1109/IoT-SIU.2019.8776561
7. Li, H., & Yu, H. (2020). Design and Implementation of Resume Screening System based on Natural Language Processing. In 2020 7th International Conference on Information Science and Control Engineering (ICISCE) (pp. 948-952). IEEE. DOI: 10.1109/ICISCE49899.2020.00043
8. Chaudhari, K. P., & Shah, A. R. (2019). An Intelligent Resume Screening System using NLP. In 2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 109-114). IEEE. DOI: 10.1109/ICECA.2019.8822563

9. Shukla, A., & Singh, N. (2019). Design and Implementation of a Smart Resume Screening System using NLP. In 2019 IEEE 2nd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech) (pp. 161-164). IEEE. DOI: 10.1109/IEMENTech.2019.8782322
10. Rane, V., & Ganorkar, S. (2019). AI-Based Resume Screening System using NLP Techniques. In 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC) (pp. 723-727). IEEE. DOI: 10.1109/ICCMC46356.2019.8942689
11. Sagar, S., & Sharma, V. (2021). A Review Paper on Resume Screening Using NLP and Machine Learning. In 2021 International Conference on Emerging Trends in Engineering Technology and Management (ICETETM) (pp. 1-5). IEEE. DOI: 10.1109/ICETETM51521.2021.9488919
12. Sharma, A., et al. (2019). A Hybrid Approach for Resume Screening using NLP and Machine Learning. In 2019 International Conference on Intelligent Sustainable Systems (ICISS) (pp. 571-576). IEEE. DOI: 10.1109/ISS45674.2019.8987492
13. Li, H., & Li, X. (2019). Design of Resume Screening System Based on Machine Learning and NLP Technology. In 2019 7th International Conference on Advanced Computing (ICoAC) (pp. 250-254). IEEE. DOI: 10.1109/ICoAC48258.2019.9034951
14. Yadav, A., & Jain, S. (2018). Resume Screening System Using Machine Learning and Natural Language Processing. International Journal of Computer Sciences and Engineering, 6(6), 283-289.
15. Kulkarni, A., et al. (2019). Design and Implementation of Resume Screening System using NLP Techniques. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 5(1), 44-49.