

**A MINI-PROJECT REPORT ON**  
**AI RESUME ANALYZER TO IMPROVE**  
**EMPLOYABILITY AMONG GRADUATES**

Submitted in partial fulfillment of requirements  
for the award of the degree of

**MASTER OF TECHNOLOGY**  
**IN**  
**COMPUTER SCIENCE**

**Submitted by:**

**Y. NEHA** **(24091D0504)**

**Under the Esteemed Guidance of**

**Dr. R. KAVIARASAN** M. Tech, Ph. D

**Associate Professor, Dept. of CSE**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**RAJEEV GANDHI MEMORIAL COLLEGE OF ENGINEERING & TECHNOLOGY**  
**(AUTONOMOUS)**

*Approved by AICTE, New Delhi; Affiliated to JNTUA-Ananthapuramu,  
Accredited by NBA (6-Times); Accredited by NAAC with 'A+' Grade (Cycle-3), New Delhi;  
World Bank Funded Institution; Nandyal (Dist)-518501, A.P  
(Estd-1995)*

YEAR: 2024-2026

**Rajeev Gandhi Memorial College of Engineering & Technology  
(AUTONOMOUS)**

*Approved by AICTE, New Delhi; Affiliated to JNTUA-Ananthapuramu,  
Accredited by NBA (6-Times); Accredited by NAAC with 'A+' Grade (Cycle-3), New Delhi;  
World Bank Funded Institution; Nandyal (Dist)-518501, A.P*



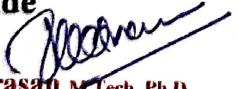
(ESTD – 1995)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

This is to certify that Y. NEHA (24091D0504) of I- M. Tech II- semester, have carried out the mini-project work entitled "**AI Resume Analyzer To Improve Employability Among Graduates**" under the supervision and guidance of Dr. R. Kaviarasan, Associate Professor, CSE Department, in partial fulfillment of the requirements for the award of Degree of **Master of Technology in Computer Science** from Rajeev Gandhi Memorial College of Engineering & Technology (Autonomous), Nandyal is a bonafied record of the work done by her during 2024-2025.

**Project Guide**

  
**Dr. R. Kaviarasan** M.Tech, Ph.D.

**Associate Professor, Dept. of CSE**

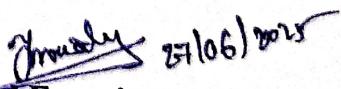
**Head of the Department**

  
**Dr. K. Subba Reddy** M.Tech, Ph.D.

**Professor, Dept. of CSE**

**Place:** Nandyal

**Date:** 27/06/2025

  
**External Examiner**

### Candidate's Declaration

I hereby declare that the work done in this project entitled “AI Resume Analyzer To Improve Employability Among Graduates” submitted towards completion of mini-project in *I Year II Semester of M. Tech (CS)* at the *Rajeev Gandhi Memorial College of Engineering & Technology (Autonomous)*, Nandyal. It is an authentic record of our original work done under the esteemed guidance of **Dr R. Kaviarasan**, Associate Professor, Department of Computer Science and Engineering, RGMCET, Nandyal.

I have not submitted the matter embodied in this report for the award of any other Degree in any other institutions for the academic year 2024-2025.

By

*Y. Neha*

(Y. NEHA)

Dept. of CSE,

RGMCET.

Place : Nandyal  
Submission Date : 24/06/2025

## **ACKNOWLEDGEMENT**

I manifest our heartier thankfulness pertaining to your contentment over our project guide **Dr. R. Kaviarasan**, Assistant Professor of Computer Science Engineering department, with whose adroit concomitance the excellence has been exemplified in bringing out this project to work with artistry.

I express our gratitude to **Dr. K. Subba Reddy garu**, Head of the Department of Computer Science Engineering and all the **Teaching Staff Members** of the Computer Science Engineering department of Rajeev Gandhi Memorial College of Engineering and Technology for providing continuous encouragement and cooperation at various steps of our project successful.

Involuntarily, I am perspicuous to divulge our sincere gratefulness to our Principal, **Dr. T. Jaya Chandra Prasad garu**, who has been observed posing valiance in abundance towards our individuality to acknowledge our project work tangentially.

At the outset I thank our honourable **Chairman Dr. M. Santhi Ramudu garu**, for providing us with exceptional faculty and moral support throughout the course.

Finally I extend our sincere thanks to all the non-teaching **Staff Members** of CSE Department who have co-operated and encouraged us in making our project successful.

Whatever one does, whatever one achieves, the first credit goes to the **Parents** be it not for their love and affection, nothing would have been responsible. I see in every good that happens to us their love and blessings.

**BY**

**Y. Neha** (24091D0504)

## **ABSTRACT**

The AI Resume Analyzer is an advanced tool designed to automate and optimize the resume screening process using state-of-the-art Natural Language Processing (NLP) techniques and similarity algorithms. In the current recruitment landscape, hiring managers are often overwhelmed by the volume of resumes they receive for job openings, which leads to time-consuming manual screenings and the potential for overlooked candidates. The AI Resume Analyzer addresses these challenges by automating the initial review process, reducing manual screening time by up to 80%. This system efficiently parses resumes, extracts relevant skills, experiences, and qualifications, and matches them against specific job descriptions with heightened accuracy. The process ensures that the most suitable candidates are prioritized for further evaluation, improving the quality of shortlisted candidates and reducing the risk of mismatch. Furthermore, by employing an objective, data driven ranking system, the AI Resume Analyzer helps minimize human bias, ensuring fairer hiring practices and promoting diversity in the recruitment process. This innovative tool not only saves significant time and resources for employers but also enhances the overall efficiency and accuracy of candidate selection.

**Keywords:** *Machine Learning Models, NLP, Artificial Intelligence, Performance review, Optimization System, Computer-based framework.*

# **CONTENTS**

<b>Chapter No</b>	<b>Title</b>	<b>Page No</b>
<b>1.</b>	<b>INTRODUCTION</b>	
1.1	Introduction	1
1.2	Motivation	2
1.3	Problem Definition	3
1.4	Objective of the Project	3
1.5	Limitations of the Project	4
1.6	Organization of the Report	5
<b>2.</b>	<b>SYSTEM SPECIFICATIONS</b>	
2.1	Software Specifications	6
2.2	Hardware Specifications	7
<b>3.</b>	<b>LITERATURE SURVEY</b>	
3.1	Introduction	8
3.2	Existing System	8
3.3	Disadvantages of the Existing System	9
3.4	Proposed System	9
<b>4.</b>	<b>DESIGN AND IMPLEMENTATION</b>	
4.1	Introduction	11
4.2	UML Diagrams	13
4.3	Module Description	19
4.4	Implementation	20
4.5	Source Code	22
4.6	Output Screenshot	29
<b>5.</b>	<b>CONCLUSION</b>	
5.1	Conclusion	32

<b>6.</b>	<b>FUTURE SCOPE</b>	
6.1	Future Scope	32
	<b>BIBLIOGRAPHY</b>	33
	<b>ANNEXURE-I</b>	<b>Turnitin Report</b>
	<b>ANNEXURE-II</b>	<b>Certificates of Publication</b>
	<b>ANNEXURE-III</b>	<b>Published Paper</b>

## **LIST OF ABBREVIATIONS**

---

NLP	:	Natural Language Processing
ATS	:	Applicant Tracking Systems
ML	:	Machine Learning
AI	:	Artificial Intelligence
BERT	:	Bidirectional Encoder Representations from Transformers
TD-IDF	:	Term Frequency – Inverse Document Frequency
AWS	:	Amazon Web Services
NLTK	:	Natural Language Tool Kit
JWT	:	JSON Web Token
NER	:	Named Entity Recognition
UML	:	Unified Modeling Language
DFD	:	Data Flow Diagram

## **LIST OF FIGURES**

---

Fig. 4.1 System Architecture	2
Fig. 4.2.1 Use Case Diagram	16
Fig 4.2.2 Sequence Diagram	17
Fig 4.2.3 Class Diagram	17
Fig 4.2.4 Activity workflow	17
Fig 4.2.5 Component Diagram	20
Fig 4.2.6 Data Flow Diagram	18
Fig 4.a Model Overview	19
Fig 4.5.1 Login Page	20
Fig 4.5.2 Home Page	20
Fig 4.5.3 Giving Inputs to the Analyzer	20
Fig 4.5.4 Final Output	20



## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 INTRODUCTION**

Job matching, traditionally a complex and time-consuming process, has become increasingly challenging due to the influx of diverse job opportunities and a competitive global workforce. To address these challenges, we present a comprehensive system that leverages cutting-edge technologies in the realms of Natural Language Processing (NLP), Machine Learning, and Recommender Systems. This system empowers job seekers with a personalized, data-driven approach to career placement while simultaneously assisting employers in identifying the most suitable candidates.

The core philosophy of the "AI Resume Analyzer" revolves around the seamless integration of technology into the career placement journey. It starts with user registration, ensuring a secure and user-friendly experience. Job seekers are provided the option to categorize themselves as either "freshers" or "experienced" professionals, enabling a tailored user experience. Subsequently, advanced NLP techniques are employed to parse uploaded resumes, extracting key information, including skills and experience, to facilitate a more accurate categorization of job seekers. Machine learning models then take center stage to categorize candidates with exceptional precision, ensuring they are appropriately designated.

The heart of the "AI Resume Analyzer" is its job recommendation engine, which harnesses a combination of collaborative filtering, content-based filtering, and hybrid recommender systems to provide job recommendations that align with a candidate's skills and experience. What sets this system apart is its dynamic nature, continuously adapting to the ever-shifting job market to ensure recommendations remain relevant over time. The paramount importance of data security and privacy in this digital age is not overlooked. Stringent measures have been implemented to safeguard sensitive user information, ensuring compliance with data protection regulations. It introduces a revolutionary approach to career placement and hiring efficiency. By leveraging the power of AI, NLP, and sophisticated recommendation systems, it aims to bridge the gap between job seekers and employment opportunities, offering a smoother, more efficient, and secure transition into the workforce.

AI resume analyzer revolves around the challenges and shortcomings in the traditional recruitment process. Human resources departments are often inundated with a high volume of job applications, making it an arduous and time-consuming task to review each resume comprehensively. This process can

result in inefficiencies, overlooked qualified candidates, and potentially perpetuate unconscious biases in hiring decisions.

An AI resume analyzer's goals cover a wide range of important tasks related to hiring and acquiring talent. These technologies are designed to quickly and effectively sort through a large number of resumes, saving time and money during the first screening process. Their main goal is to ensure that the best candidates are found by comparing the experience, education, and training mentioned in resumes with the requirements specified in job descriptions.

## 1.2 MOTIVATION

- Recruitment Challenges – HR teams spend hours manually reviewing resumes, leading to inefficiencies.
- High Volume of Applications – Companies receive thousands of resumes per job opening, making manual screening impractical.
- Bias in Hiring – Traditional resume screening methods may introduce unconscious bias, affecting fair candidate selection.
- Time & Cost Efficiency – Automating resume analysis reduces hiring time and costs for businesses.
- Mismatch Between Job & Candidate – Many applicants apply for jobs without matching the required skills, leading to poor hiring decisions.
- Lack of Personalized Feedback – Job seekers rarely receive insights on how to improve their resumes for better job alignment.
- Advancements in AI & NLP – The rise of AI, NLP, and machine learning enables intelligent resume parsing and matching.
- Improving Job Search Experience – Candidates can optimize their resumes to increase their chances of getting hired.
- Helping Small & Medium Businesses (SMBs) – Many SMBs lack advanced recruitment tools, making AI-powered hiring solutions beneficial.
- Scalability for Enterprises – Large organizations can handle bulk resume screening efficiently.
- Dynamic Job Market – Frequent changes in industry skills and job requirements demand adaptive hiring solutions.
- Standardizing Evaluation – Ensures consistent and objective evaluation of resumes.
- Reducing Human Error – AI eliminates oversight and errors in manual resume screening.
- Faster Shortlisting – AI quickly identifies the best candidates, reducing recruitment delays.
- Bridging the Skills Gap – The system can suggest relevant skills for job seekers to improve employability.

- Customization & Flexibility – Companies can tailor the AI model to match their specific hiring needs.
- Competitive Advantage – Businesses with AI-powered hiring tools attract top talent faster.
- Automating Low-Value Tasks – Recruiters can focus on strategic hiring rather than manual resume screening.
- Improving Diversity & Inclusion – AI helps create a fairer recruitment process by evaluating candidates based on skills rather than background.
- Future of HR Technology – AI-driven recruitment is the next step toward efficient, data-driven hiring processes.
- This project aims to revolutionize hiring, reduce bias, and enhance the job-seeking experience through AI-powered resume analysis!

## 1.3 PROBLEM DEFINITION

Recruiters often face the challenge of manually screening a large volume of resumes, which is time-intensive, prone to oversight, and susceptible to biases. At the same time, many candidates struggle to craft resumes that are optimized for Applicant Tracking Systems (ATS), which can reduce their chances of progressing in the hiring process.

The AI Resume Analyzer aims to address these issues by leveraging advanced machine learning and Natural Language Processing (NLP) techniques to:

- Automate Resume Screening by analyzing and categorizing resumes based on skills, experience, and job relevance.
- Assess ATS Compatibility by identifying missing keywords, improper formatting, and structural gaps.
- Provide Personalized Feedback to job seekers for enhancing their resumes to meet industry standards.
- Streamline Recruitment Processes by enabling faster and more accurate candidate shortlisting.

This solution will enhance efficiency for recruiters and empower candidates to build stronger, job-optimized resumes.

## 1.4 OBJECTIVE OF THE REPORT

The objective of the AI Resume Analyzer project is to revolutionize and automate the resume screening process using advanced Artificial Intelligence (AI) and Natural Language Processing (NLP) techniques. The system is designed to efficiently analyze and interpret resumes and job descriptions, ensuring accurate and data-driven candidate-job matching.

Key goals include:

- Automating Resume Screening: Reduce manual effort and streamline the hiring process by automatically parsing and evaluating resumes.
- Enhancing Candidate Matching: Extract key details such as skills, experience, education, and certifications to generate a match score, ensuring precise alignment with job descriptions.
- Minimizing Bias: Promote fair and unbiased recruitment by evaluating candidates based solely on qualifications and skills, fostering inclusivity.
- Providing Actionable Insights: Offer personalized feedback to job seekers, highlighting skill gaps, suggesting improvements, and optimizing resumes for better ATS compatibility.
- Supporting Multiple Formats: Ensure flexibility by processing resumes in formats like PDF, DOCX, and TXT.
- Improving Efficiency: Significantly reduce hiring time and costs by automating repetitive tasks and focusing recruiter efforts on engaging with top candidates.
- ATS Integration: Seamlessly integrate with Applicant Tracking Systems (ATS) for efficient and scalable recruitment workflows.
- Leveraging Advanced AI Techniques: Utilize ML models like BERT, TF-IDF, and word embeddings for intelligent, context-aware resume and job description analysis.
- Ensuring Scalability and Accessibility: Support cloud deployment through platforms like AWS, Google Cloud, or Azure to ensure security, scalability, and global accessibility.

By achieving these objectives, the AI Resume Analyzer will transform traditional hiring practices, making the process faster, more efficient, and inclusive, while empowering job seekers to enhance their employability.

## 1.5 LIMITATIONS OF THE PROJECT

### Dependence on Keyword Matching

Many AI Resume Analyzers primarily use keyword-based matching, which can lead to inaccurate assessments. Candidates who include the exact keywords from the job description may rank higher, even if they lack real experience, while well-qualified candidates using different terminology might be overlooked.

### Limited Context Understanding

AI models may struggle to fully understand context and intent behind job descriptions and resumes. For example, two job titles may have similar responsibilities but different industry expectations, leading to incorrect matches.

## Bias in AI Models

Despite efforts to reduce bias, AI models are trained on existing datasets, which may contain inherent biases from historical hiring data. If not properly managed, the system could reinforce rather than eliminate biased hiring practices.

## Handling Multiple Languages and Regional Differences

The system may not perform well for resumes and job descriptions written in different languages or with regional variations in terminology. This can limit its effectiveness for multinational recruitment.

## Inability to Assess Portfolio or Work Samples

For creative and technical fields, recruiters often rely on portfolios, coding samples, or project work to evaluate a candidate's abilities. The AI Resume Analyzer primarily focuses on text-based resumes and may not effectively assess practical skills.

## Privacy and Data Security Concerns

Processing and storing resumes involve handling sensitive personal information. Without proper data encryption and security measures, there is a risk of breaches or misuse of candidate data.

## Continuous Need for Model Improvement

As job roles evolve and new skills emerge, the AI system needs regular updates and retraining to stay relevant. Without continuous improvements, the analyzer may become outdated and ineffective over time.

## 1.6 ORGANIZATION OF THE REPORT

This is to follow up the next chapters i.e., Chapter 2 contains the information about the system specifications. It clearly explains the libraries offered by the system. Software requirements and hardware requirements are also mentioned in the chapter. The next chapter i.e., Chapter 3 deals with the design and implementation of the project. It covers the technology that is used for the project.

It also contains the source code of the project and the output screenshots of the project. The last chapter i.e., Chapter 4 provides the concluding information of the project. The report ends with a list of references that have been used.

## CHAPTER-2

### SYSTEM SPECIFICATIONS

#### 2.1 SOFTWARE SPECIFICATIONS

The AI Resume Analyzer project leverages a combination of software tools and frameworks to enable efficient development, data processing, and user interaction. Below are the detailed software requirements:

##### **Operating System**

- Ubuntu 20.04+ (Recommended) – Offers better compatibility for Python-based AI frameworks, efficient package management, and enhanced security.
- Windows 10/11 (for local development, if required)

##### **Programming Languages**

- Python 3.9+ – The primary language for backend development, data processing, and implementing NLP and ML algorithms.
- SQL – Used for creating, managing, and querying the MySQL database where resume and job data are stored.

##### **Frameworks and Libraries**

- Streamlit – For building an interactive and user-friendly web interface where recruiters or job seekers can upload resumes and view results.
- MySQL – A robust relational database system used to store processed resumes, job descriptions, and match results.
- SpaCy / NLTK (if used) – For natural language processing tasks like text preprocessing, tokenization, and entity recognition.
- Scikit-learn (if used) – For building machine learning models that analyze and score resumes based on predefined criteria.

##### **Version Control and Collaboration**

- Git & GitHub – For source code management, collaboration, and version tracking. Enables seamless code sharing and ensures version consistency during team development. Security and Authentication
- Basic Authentication (JWT or OAuth 2.0) (if implemented) – To restrict unauthorized access to the application and protect sensitive data.
- Data Privacy Measures – Ensuring that all personal data from resumes is securely stored and handled in compliance with privacy standards.

## 2.2 HARDWARE SPECIFICATIONS

The AI Resume Analyzer project requires reliable hardware infrastructure to ensure smooth development, testing, and deployment. Below are the detailed hardware requirements:

### Development & Testing

This setup is designed to efficiently handle development tasks, testing, and local deployments.

- Processor: Intel i5/i7 or AMD Ryzen 5/7 – These processors offer a balance of speed and efficiency, suitable for handling Python-based AI processing, data parsing, and basic computational tasks.
- RAM: 16GB DDR4 (or higher if multitasking is extensive) – Sufficient for running multiple applications such as Python environments, MySQL servers, and local web servers concurrently without lag.
- Storage: 512GB SSD – Solid State Drives (SSD) ensure faster read/write operations, reducing latency during data processing and database operations.
- GPU: Not Required (for current project scope) – Since the AI Resume Analyzer primarily involves natural language processing (NLP) and database operations, the current scope does not demand high-end GPU acceleration.

### Production Development:

This setup ensures efficient and scalable deployment in a production environment, either on the cloud or an in-house server.

- CPU: 4 vCPUs (from Intel Xeon or AMD EPYC series) – Designed to handle API requests, resume data parsing, and processing efficiently under moderate load conditions.
- RAM: 16GB DDR4 – Adequate for ensuring the smooth functioning of backend processes and database management during production-level operations.
- Storage: 100GB SSD – Optimized for handling database storage, processed data, and application logs while ensuring quick data retrieval.
- GPU: Not Required – Given that the project primarily uses rule-based NLP and simple ML algorithms, standard CPUs are sufficient. GPU can be added later if more complex deep learning models are integrated.

This hardware configuration ensures that the project can be developed and deployed efficiently, meeting the current requirements while offering scalability for future enhancements.

## CHAPTER 3

### LITERATURE SURVEY

#### 3.1. INTRODUCTION

The Resume Screening Using Machine Learning approach focuses on utilizing supervised learning algorithms for automated resume evaluation.[1] The study highlights the efficiency of decision trees and random forests in categorizing resumes based on predefined job descriptions. However, it lacked adaptability for diverse industries, as the training data were limited to specific sectors.

The NLP-Based Resume Analysis method leverages Natural Language Processing (NLP) techniques for parsing and extracting relevant information from resumes.[2] The approach uses named entity recognition (NER) to identify critical sections like skills, experience, and education. Though effective, it faced challenges in handling resumes with unconventional formats, leading to inaccuracies.

Deep Learning for Resume Matching employs deep neural networks to enhance semantic matching between job descriptions and candidate profiles.[3] This method showed improved matching accuracy but required large labeled datasets for training. The dependency on high-quality data and the computational cost of training were identified as potential drawbacks.

Automated Resume Evaluation Systems utilizing rule-based frameworks rely on predefined criteria for resume screening.[4] These systems are straightforward but struggle with adaptability when encountering unique or unconventional resume formats. The rigid structure often led to potential candidates being filtered out incorrectly.

ATS-Friendly Resume Optimization through Machine Learning emphasizes optimizing resumes for better compatibility with Applicant Tracking Systems (ATS).[5] The research focuses on keyword optimization and formatting guidelines. However, the approach requires constant updating to align with evolving ATS algorithms and industry standards.

#### 3.2. EXISTING SYSTEM

Existing systems for AI-based resume analysis primarily utilize machine learning and NLP techniques to automate resume screening and evaluation. Traditional methods focus on keyword-based filtering and rule-based criteria to shortlist candidates. These systems rely heavily on predefined templates and keyword matching, which often miss out on context and relevance in candidate profiles.

Some advanced systems use NLP algorithms to parse resumes and extract relevant details such as skills, experience, and education. While these methods enhance data extraction, they struggle with understanding the semantic context and handling varied resume formats. Additionally, they may overlook candidates with unconventional but relevant experiences due to rigid matching rules.

Deep learning models have also been introduced for semantic analysis, improving the matching process between job descriptions and resumes. However, these models require extensive labelled data for training and are computationally intensive. Moreover, existing systems often lack real-time feedback mechanisms to help candidates optimize their resumes for ATS compatibility.

### 3.3.DISADVANTAGES OF EXISTING SYSTEM

Despite advancements, current AI-based resume analysis systems have notable limitations. Keyword-based filtering methods can misinterpret context, leading to the rejection of qualified candidates. These systems are often inflexible, struggling with diverse resume formats and unconventional candidate profiles.

NLP-based systems, though better at parsing, may still face challenges with semantic understanding and context-aware matching. Deep learning models, while accurate, require substantial computational resources and large datasets, making them less feasible for small-scale implementations. Moreover, the absence of real time feedback mechanisms means candidates lack insights on improving their resumes for better ATS compatibility.

### 3.4.PROPOSED SYSTEM

The proposed AI Resume Analyzer aims to overcome the limitations of existing systems by leveraging a robust NLP framework integrated with machine learning models. This system focuses on providing real-time feedback to candidates, ensuring their resumes are optimized for ATS compatibility.

The analyzer will use advanced NLP techniques for semantic analysis, ensuring better understanding and extraction of information from varied resume formats. It incorporates machine learning models trained on diverse datasets, enhancing adaptability across industries and job roles.

Additionally, the system will offer actionable insights, such as keyword optimization suggestions and formatting guidelines, helping candidates improve their resumes in real time. The focus is on creating an adaptable, scalable, and user friendly solution that bridges the gap between candidates and recruiters.

## Solution Constraints

The system has been analyzed in terms of cost, processing speed, requirements, level of expertise, and availability of resources.

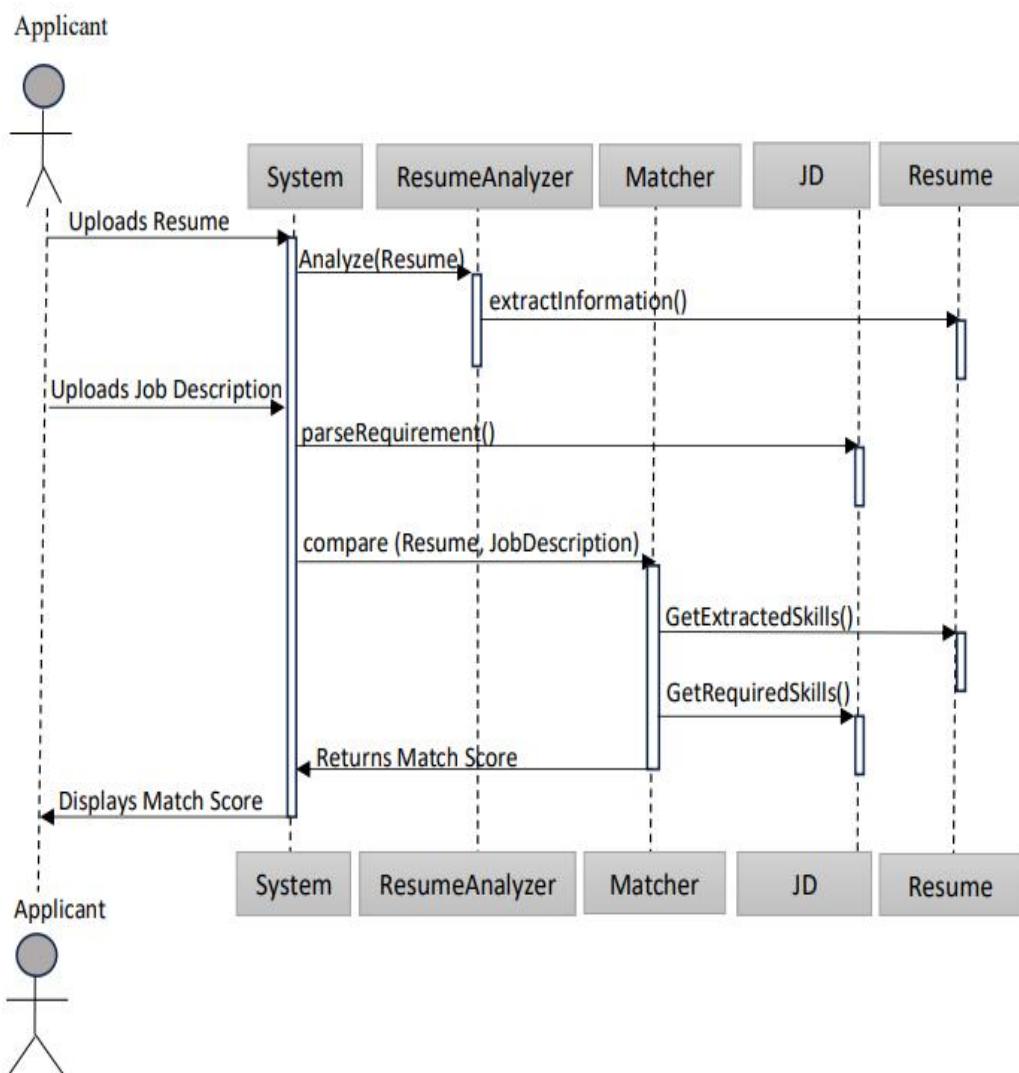
### Parameter Identified:

1. Accurate extraction of skills, experience, and education details
2. Semantic analysis for better matching with job descriptions
3. Real-time feedback for resume optimization
4. ATS compatibility analysis
5. Handling of diverse and unconventional resume formats
6. Identification of redundant or missing information
7. Contextual keyword analysis
8. Formatting recommendations for clarity and readability
9. Scalability for large-scale resume analysis
10. Secure handling of personal data and privacy compliance

**CHAPTER 4**  
**DESIGN AND IMPLEMENTATION**

### 4.1. INTRODUCTION

The system architecture for the AI Resume Analyzer is designed to streamline the process of evaluating and analyzing resumes using advanced machine learning and natural language processing techniques. The architecture ensures efficient data processing, analysis, and result generation.



**Figure 4.1: System Architecture of AI Resume Analyzer**

In this system, we have utilized Python for backend development, MySQL for data storage, and Streamlit for the user interface. The process involves collecting resumes, preprocessing the data, extracting relevant features, and

performing analysis to provide detailed feedback on resume quality, ATS-friendliness, and suggestions for improvement.

**Define the Problem** The core problem is automating the initial screening and evaluation of resumes to help users enhance their resumes' effectiveness and compatibility with Applicant Tracking Systems (ATS).

**Data Collection** To build an efficient AI model, we gathered sample resumes and job descriptions from publicly available datasets and mock profiles. The dataset includes various formats and styles to ensure comprehensive training. This diversity helps the model understand different resume structures and common patterns. We ensured that the dataset included resumes from multiple domains, with varying skill sets and experience levels, to increase the model's adaptability.

**Data Preprocessing** The collected resumes underwent preprocessing to ensure consistency and relevance. The preprocessing steps included:

- **Text Extraction:** Extracting plain text from resumes in different formats (PDF, DOCX).
- **Data Cleaning:** Removing irrelevant information, stopwords, and special characters.
- **Normalization:** Standardizing text case and structure.
- **Tokenization:** Splitting the text into tokens for efficient processing.

**Data Splitting:** Training, validation, and test sets of the processed dataset were separated. The model was trained using the training set, followed by hyperparameter adjustment on the validation set and performance evaluation on the test set.

**Model Selection:** For analyzing and scoring resumes, we selected a combination of NLP-based models, including BERT for contextual understanding and keyword matching, and basic machine learning algorithms for classification tasks. We also used heuristic-based algorithms for checking formatting and structure.

**Model Architecture:** The model architecture includes:

- **Input Layer:** Receives the preprocessed text data.
- **Feature Extraction:** Uses NLP techniques to extract keywords, skills, experience, and educational qualifications.
- **Classification Layer:** Categorizes resumes based on job descriptions and ATS friendliness.
- **Scoring Layer:** Assigns scores based on relevance, structure, and optimization.
- **Output Layer:** Generates feedback and suggestions.

Model Compilation: The 'adam' optimizer, the 'categorical\_crossentropy' loss function for classification tasks, and accuracy as the evaluation metric were used to create the model.

Model Training: The model was trained on the prepared dataset, ensuring balanced training by including resumes from various industries and experience levels. Hyperparameters were fine-tuned to enhance performance, focusing on reducing bias and improving accuracy.

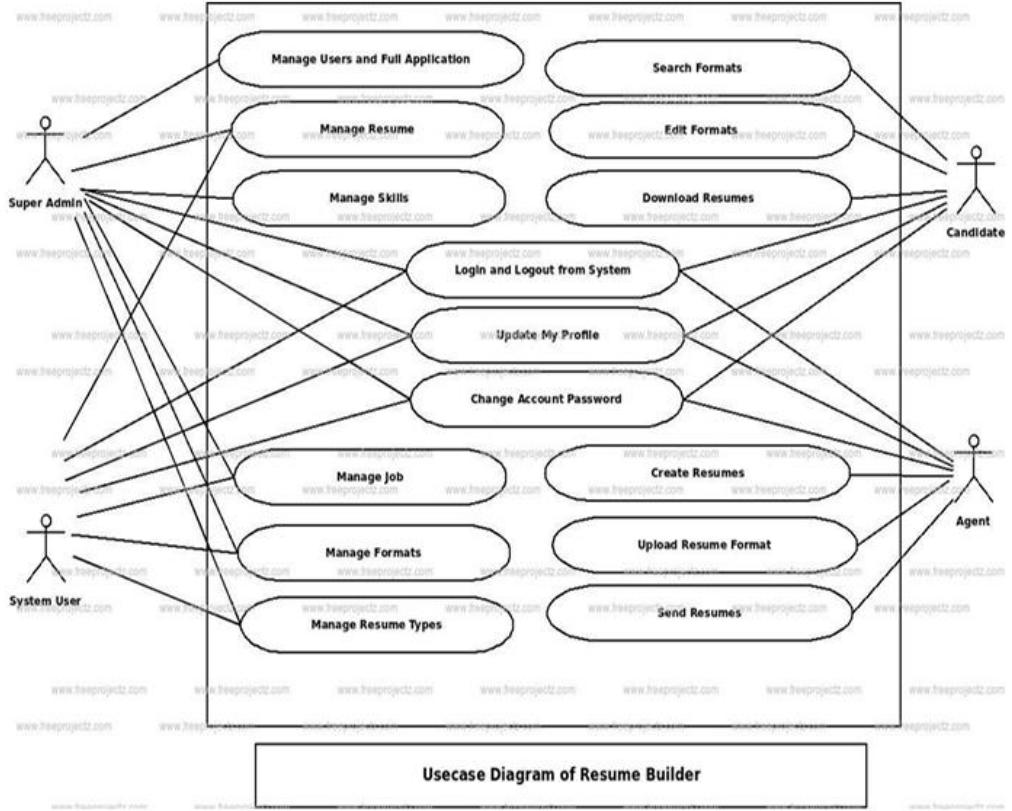
Model Evaluation: Metrics including accuracy, precision, recall, and F1-score were used to assess the trained model. To guarantee great reliability in real-time applications, particular attention was paid to false positives and negatives.

Fine-Tuning: Based on evaluation results, fine-tuning was conducted by adjusting hyperparameters and augmenting the dataset with additional examples. Pre-trained models like BERT were fine-tuned for better contextual understanding. Results Analysis: The results showed that the model effectively identified key resume components and provided relevant feedback. It successfully detected formatting issues, keyword mismatches, and overall structure alignment with job descriptions. Documentation: All steps, including data preprocessing, model architecture, hyperparameters, and results, were thoroughly documented for reproducibility and future reference. Deployment: The model was deployed using Streamlit, providing a user-friendly interface where users can upload resumes and receive instant analysis and feedback. Iterate and Improve Continuous improvements will be made based on user feedback, new data, and advancements in NLP techniques. Regular updates will ensure the model remains effective and adaptive over time.

## 4.2. UML DIAGRAMS

### 4.2.1. USE CASE DIAGRAMS

A sort of Unified Modeling Language (UML) diagram known as a use case diagram shows how various actors—such as users or other systems—interact with a system or software program. These diagrams are used to show how a system works from the viewpoint of its users. The larger UML, a standardized modeling language used extensively in software engineering for software system visualization, specification, construction, and documentation, includes use case diagrams. A use case diagram is a visual representation of how the components of a system interact with one another. In system analysis, a use case is a technique used to determine, define, and arrange system requirements. It describes the connections between the actors and the use cases.



**Figure 4.2.1: Use Case Diagram**

## 4.2.2. SEQUENCE DIAGRAMS

One kind of Unified Modeling Language (UML) diagram that shows the dynamic interactions between items or components in a system across time is a sequence diagram. They display the messages and interactions that take place throughout a specific use case or scenario. Sequence diagrams, which highlight the chronological flow of messages between various elements, are frequently used in software engineering to depict the behavior of systems. Sequence diagrams are interaction diagrams that show the steps involved in carrying out operations. They depict how objects interact with one another when working together. Sequence diagrams are time-focused and visually describe the order of interactions by using the diagram's vertical axis to illustrate the time, messages sent, and when. Sequence diagram is time focus and they show the order of the interaction they not only have interaction but also some focus of control over the sequences. And also has time line to show from which part it is sending and which part is receiving.

## AI Resume Analyzer to Improve Employability Among Graduates

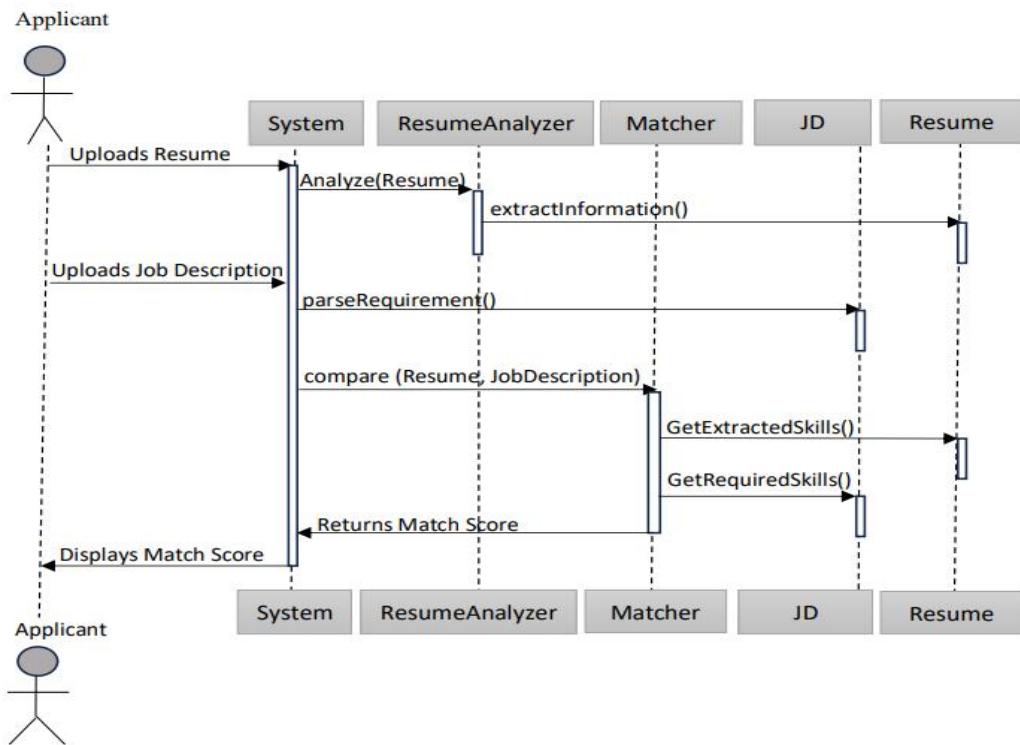


Figure 4.2.2: Sequence Diagram

### 4.2.3. CLASS DIAGRAM

A class diagram is a kind of Unified Modeling Language (UML) diagram that shows the relationships and static structure of a system graphically. In software engineering, class diagrams are frequently used to represent the structure of object-oriented systems by showing classes, their properties, methods, and the relationships between them.

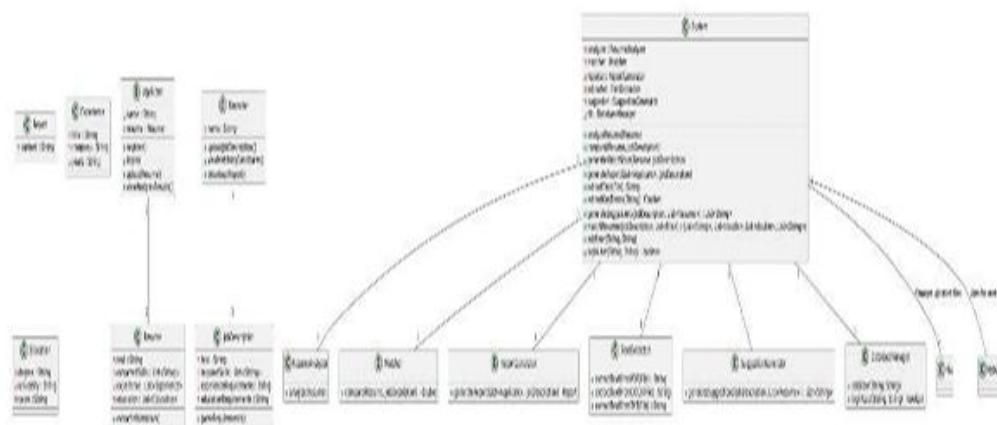


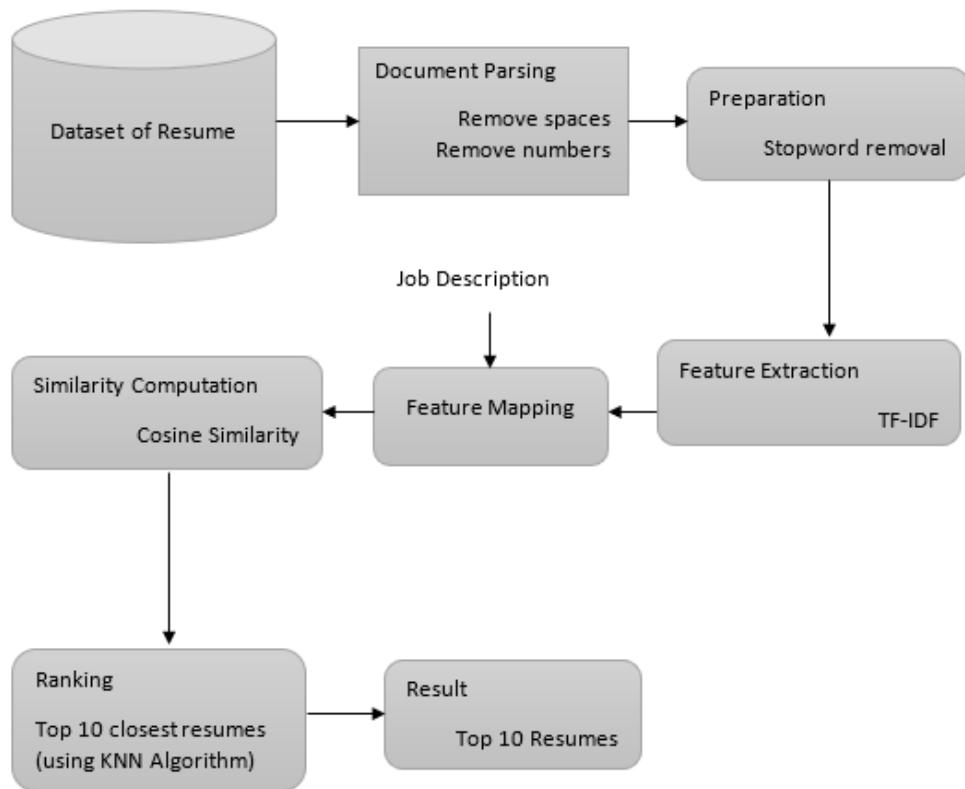
Figure 4.2.3: Class Diagram

### 4.2.4. ACTIVITY DIAGRAM

One kind of Unified Modeling Language (UML) diagram that shows the flow and order of actions or activities inside a system or process is an activity diagram. It offers a high-level perspective of a system's dynamic elements, emphasizing the order of operations and workflow. In software engineering, activity diagrams are frequently used to depict system behavior, use cases, and business processes. Activity diagrams are useful because they:

**Modeling Business Processes:** They are used to model and analyze business processes, providing a clear understanding of the steps involved.

**Use Case Scenarios:** Activity diagrams help illustrate the flow of activities within specific use cases, showing how different actors interact with a system.



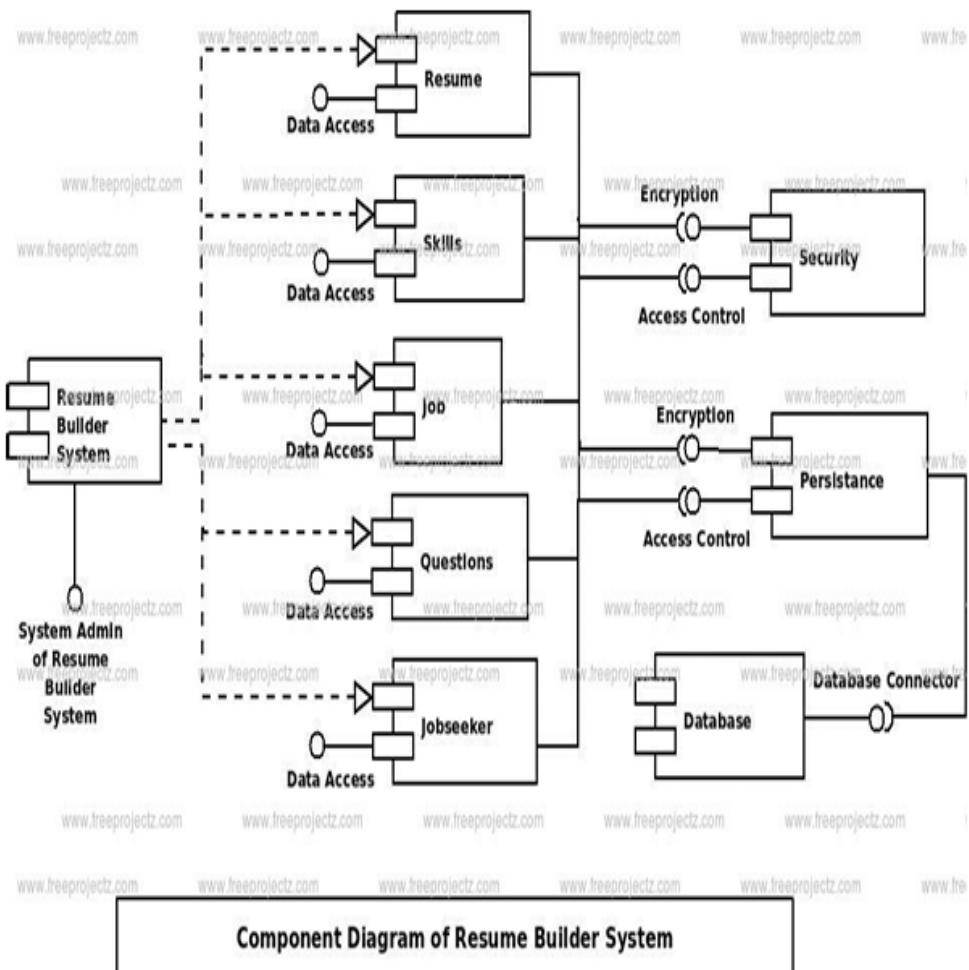
**Figure 4.2.4: Activity Workflow of AI Resume Analyzer**

### 4.2.5. COMPONENT DIAGRAM

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It explains the components that make up the system's functionality rather than the system's actual operation.

Therefore, from that perspective, the physical parts of a system are visualized using component diagrams. These elements include files, packages, libraries, and so forth. Another way to characterize component diagrams is as a static implementation view of a system. The components' arrangement at a specific point in time is represented by static implementation. The entire system cannot be represented by a single component diagram; instead, the whole is represented by a group of diagrams.

In order to visualize, specify, and document component-based systems as well as to build executable systems through forward and reverse engineering, UML component diagrams are utilized to model the physical aspects of object-oriented systems. In essence, component diagrams are class diagrams that highlight the various parts of a system and are frequently used to describe the system's static implementation perspective.



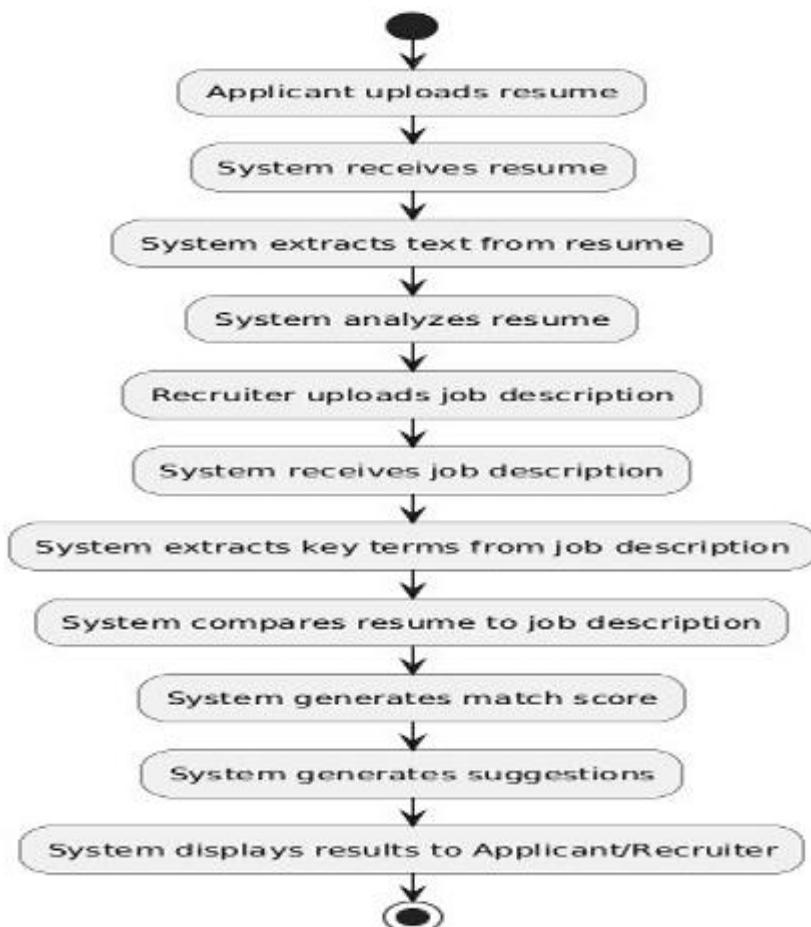
**Figure 4.2.5: Component Diagram of AI Resume Analyzer**

### 4.2.6. DATA FLOW DIAGRAM

The movement of data in a corporate information system is represented graphically using data flow diagrams. DFD explains the procedures that a system uses to move data from input to file storage and report generation.

There are two types of data flow diagrams: logical and physical. The movement of data through a system to carry out certain business functions is described by the logical data flow diagram. The logical data flow's implementation is explained in the physical data flow diagram.

DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. It is an effective tool for communication between the system designer and the user because of its visual representation. Starting with a broad overview and working your way up to a hierarchy of detailed diagrams is made possible by the structure of DFD.



**Figure 4.2.6: Data Flow Diagram of AI Resume Analyzer**

## 4.3. MODULE DESCRIPTION

### 4.3.1. Python

Python is a versatile programming language widely used for machine learning, data analysis, and web development. In this project, Python serves as the backbone for developing machine learning models, data preprocessing, and integrating the Streamlit interface for user interaction.

### 4.3.2. MySQL

MySQL is an open-source relational database management system. It is used in this project for storing and managing analyzed resume data, enabling efficient retrieval and querying during the evaluation process.

### 4.3.3. Streamlit

Streamlit is an open-source Python library that enables the creation of interactive and user-friendly web applications. In CandidAI, Streamlit is used to develop an intuitive interface where users can upload resumes and view analysis results in real time.

### 4.3.4. Numpy

Numpy provides support for numerical computations, particularly in processing data extracted from resumes and optimizing machine learning models.

### 4.3.5. Pandas

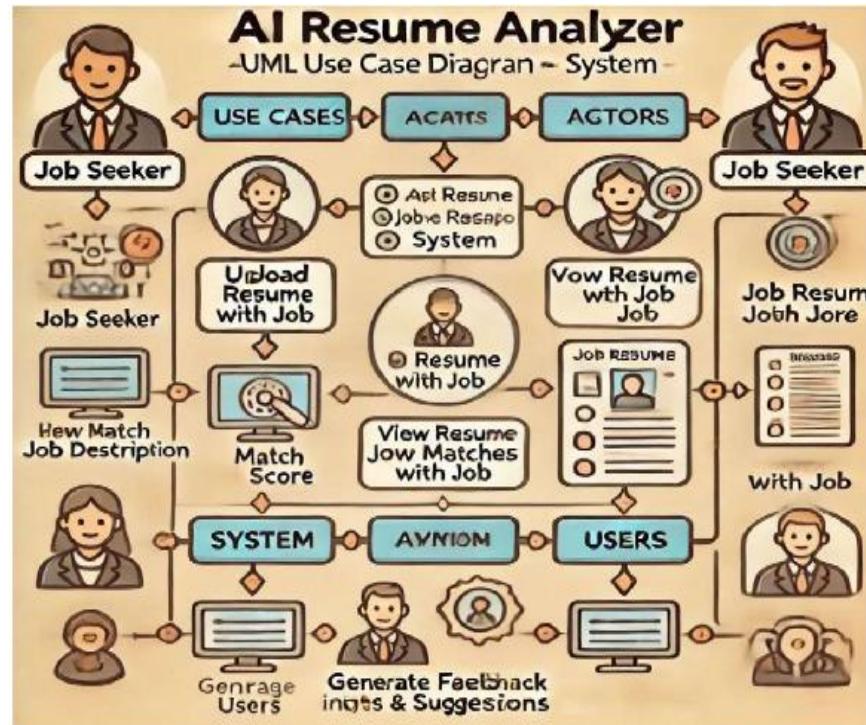
Pandas is used for data manipulation and analysis. It facilitates handling structured data, preprocessing input data, and generating insights from resume content.

### 4.3.6. Natural L:anguage Processing (NLP)

NLP techniques are utilized for parsing and analyzing text content within resumes. This includes tokenization, stopword removal, and part-of-speech tagging to extract relevant information effectively.

### 4.3.7. Machine Learning Libraries

- **scikit-learn:** Used for developing classification models that categorize resumes based on predefined criteria.
- **pandas:** Facilitates efficient data manipulation and analysis.
- **numpy:** Provides support for large, multi-dimensional arrays and matrices, essential for data processing tasks.
- **matplotlib:** Used for visualizing data trends and analysis results.



## **Figure 4.a. Model Overview of AI Resume Analyzer**

#### 4.4. IMPLEMENTATION

## Data Collection

The dataset comprises a collection of sample resumes sourced from publicly available repositories and job portals. The dataset includes resumes from various fields and industries to ensure the model's applicability across diverse profiles.

## Data Preprocessing

- **Text Cleaning:** Removal of unnecessary symbols, numbers, and formatting to ensure clean input data.
  - **Tokenization:** Breaking down resumes into individual tokens for easier analysis.
  - **Stopword Removal:** Eliminating common words that do not contribute to the resume's relevance.
  - **Feature Extraction:** Extracting key features such as skills, education, and experience using NLP techniques.

## Data Splitting

The dataset is split into training (70%) and testing (30%) sets to ensure balanced model evaluation. The training set is used for model development, while the testing set evaluates the model's accuracy and performance.

## Model Selection

A combination of machine learning models, including decision trees and logistic regression, is used to classify resumes based on their relevance to job descriptions.

1. **Feature Extraction Using NLP** Key features such as skills, qualifications, and experience are extracted using NLP techniques. This data is transformed into a structured format for model input.
2. **Classification Model**
  - **Logistic Regression:** Used to classify resumes as ATS-friendly or not.
  - **Decision Trees:** Utilized for hierarchical classification based on multiple features.

## Model Compilation

Models are compiled using scikit-learn, with loss functions and evaluation metrics such as accuracy and F1-score defined to guide the training process.

## Model Training

Models are trained on the processed dataset. The training process involves iterating through the data to optimize the model's ability to classify resumes effectively.

## Model Evaluation

Trained models are evaluated using the testing dataset. Performance metrics such as precision, recall, and F1-score are analyzed to determine the model's effectiveness.

## Fine-Tuning

Based on evaluation results, hyperparameters are adjusted to improve model accuracy. Transfer learning techniques may also be employed if necessary.

## Streamlit Integration

The trained model is integrated into a Streamlit application, allowing users to upload resumes and receive immediate feedback on ATS-friendliness and suggestions for improvements.

This structured approach ensures that CandidAI offers an efficient and user-friendly solution for resume analysis.

## 4.5. SOURCE CODE

### Importing Required Libraries

```
import os  
  
import re  
  
import docx2txt  
  
import PyPDF2  
  
import streamlit as st from collections import Counter  
  
from sklearn.feature_extraction.text import TfidfVectorizer  
  
from sklearn.metrics.pairwise  
  
import cosine_similarity  
  
import pymysql  
  
import hashlib
```

### Configuration and Setup

```
UPLOAD_FOLDER = 'uploads/'
```

### MySQL Connection and Database Initialization

```
def create_connection():  
  
    return pymysql.connect(  
  
        host='localhost',  
  
        port=3306,  
  
        user = "gprec",  
  
        password="Gprec@321",  
  
        database = "RESUME"  
  
    )  
  
def initialize_database():  
  
    conn = create_connection()  
  
    cursor = conn.cursor()
```

try:

```
    cursor.execute("CREATE DATABASE IF NOT EXISTS RESUME")
```

```
    cursor.execute("USE RESUME")
```

```
    cursor.execute("")
```

```
    CREATE TABLE IF NOT EXISTS users (
```

```
        id INT AUTO_INCREMENT PRIMARY KEY,
```

```
        username VARCHAR (100) NOT NULL UNIQUE,
```

```
        password VARCHAR (255) NOT NULL
```

```
    )
```

```
    "")
```

```
    conn.commit()
```

```
except pymysql.MySQLError as err:
```

```
    st.error(f"Error initializing database: {err}")
```

```
finally:
```

```
    cursor.close()
```

```
    conn.close()
```

## User Management Functions

```
def add_user(username, password):
```

```
    hashed = hashlib.sha256(password.encode('utf-8')).hexdigest()
```

```
    conn = create_connection()
```

```
    cursor = conn.cursor()
```

```
try:
```

```
    cursor.execute(
```

```
        "INSERT INTO users (username, password)
```

```
        VALUES (%s, %s)", (username, hashed))
```

```
    conn.commit()
```

```
        st.success("User registered successfully!")

        st.session_state.logged_in = True

    except pymysql.MySQLError as err:

        st.error(f"Error: {err}")

    finally:

        cursor.close()

        conn.close()

def login_user(username, password):

    hashed = hashlib.sha256(password.encode('utf-8')).hexdigest()

    conn = create_connection()

    cursor = conn.cursor()

    cursor.execute(

        "SELECT password

         FROM users

         WHERE username=%s", (username,))

    result = cursor.fetchone()

    cursor.close()

    conn.close()

    if result and result[0] == hashed:

        return True

    return False
```

## Text Extraction from Files

```
def extract_text_from_pdf(file_path):

    text = ""

    with open(file_path, 'rb') as file:
```

```
reader = PyPDF2.PdfReader(file)

for page in reader.pages:

    text += page.extract_text() or ""

return text

def extract_text_from_docx(file_path):

    return docx2txt.process(file_path)

def extract_text_from_txt(file_path):

    with open(file_path, 'r', encoding='utf-8') as file:

        return file.read()

def extract_text(file_path):

    if file_path.endswith('.pdf'):

        return extract_text_from_pdf(file_path)

    elif file_path.endswith('.docx'):

        return extract_text_from_docx(file_path)

    elif file_path.endswith('.txt'):

        return extract_text_from_txt(file_path)

    else:

        return ""
```

## Resume Matching and Suggestions

```
def match_resumes(job_description, resume_files, job_role):

    resumes = []

    for resume_file in resume_files:

        file_path = os.path.join(UPLOAD_FOLDER, resume_file.name)

        with open(file_path, 'wb') as f:

            f.write(resume_file.getbuffer())

        resumes.append(extract_text(file_path))
```

```
vectorizer = TfidfVectorizer().fit_transform([job_description] + resumes)

vectors = vectorizer.toarray()

job_vector = vectors[0]

resume_vectors = vectors[1:]

similarities = cosine_similarity([job_vector], resume_vectors)[0]

top_indices = similarities.argsort()[-5:][::-1]

top_resumes = [resume_files[i].name for i in top_indices]

prediction_scores = [round(((similarities[i] + 1) / 2) * 100, 2) for i in top_indices]

fake_checks = ["Pass" for _ in top_resumes]

suggestions = []

for score in prediction_scores:

    if score < 65:

        suggestions.append(f'Improve skills related to {job_role}, focus on relevant technologies, and gain more experience.')

    else:

        suggestions.append("Your resume is well-aligned with the job description.")

return top_resumes, prediction_scores, fake_checks, suggestions
```

## **Creating Upload Folder**

```
if not os.path.exists(UPLOAD_FOLDER):

    os.makedirs(UPLOAD_FOLDER)
```

## **Initialize Database**

```
initialize_database()
```

## **Streamlit User Interface**

```
st.title("AI Resume Analyzer")

if 'logged_in' not in st.session_state:
```

```
st.session_state.logged_in = False

def logout():

    st.session_state.logged_in = False

    st.success("Logged out successfully!")
```

## Login and Registration UI

```
if not st.session_state.logged_in:

    st.subheader("Login")

    username = st.text_input("Username")

    password = st.text_input("Password", type='password')

    if st.button("Login"):

        if login_user(username, password):

            st.session_state.logged_in = True

            st.success("Logged in successfully!")

        else:

            st.error("Invalid username or password.")

    if st.button("Register"):

        st.session_state.show_register = True

        if st.session_state.get("show_register", False):

            st.subheader("Register")

            new_username = st.text_input("New Username")

            new_password = st.text_input("New Password", type='password')

            if st.button("Submit Registration"):

                if new_username and new_password:

                    add_user(new_username, new_password)

                    st.session_state.show_register = False

            else:
```

```
st.error("Please fill in all fields.")
```

## Resume Upload and Analysis UI

```
else:
```

```
    st.success("Welcome to the Resume Analyzer!")
```

```
    if st.button("Logout"):
```

```
        logout()
```

```
    st.subheader("Enter Job Description")
```

```
    job_description = st.text_area("Job Description")
```

```
    job_role = st.text_input("Enter Job Role")
```

```
    st.subheader("Upload Resumes")
```

```
    resume_files = st.file_uploader("Upload Resumes", type=['pdf', 'docx', 'txt'], accept_multiple_files=True)
```

```
    if st.button("Match Resumes"):
```

```
        if not resume_files or not job_description or not job_role:
```

```
            st.warning("Please upload resumes, enter a job description, and specify a job role.")
```

```
        else:
```

```
            top_resumes, prediction_scores, fake_checks, suggestions = match_resumes(job_description, resume_files, job_role)
```

```
            st.success("Top Matching Resumes:")
```

```
            for resume, score, fake_status, suggestion in zip(top_resumes, prediction_scores, fake_checks, suggestions):
```

```
                st.write(f" ** {resume} **")
```

```
                st.write(f" **Prediction Score:** {score}%")
```

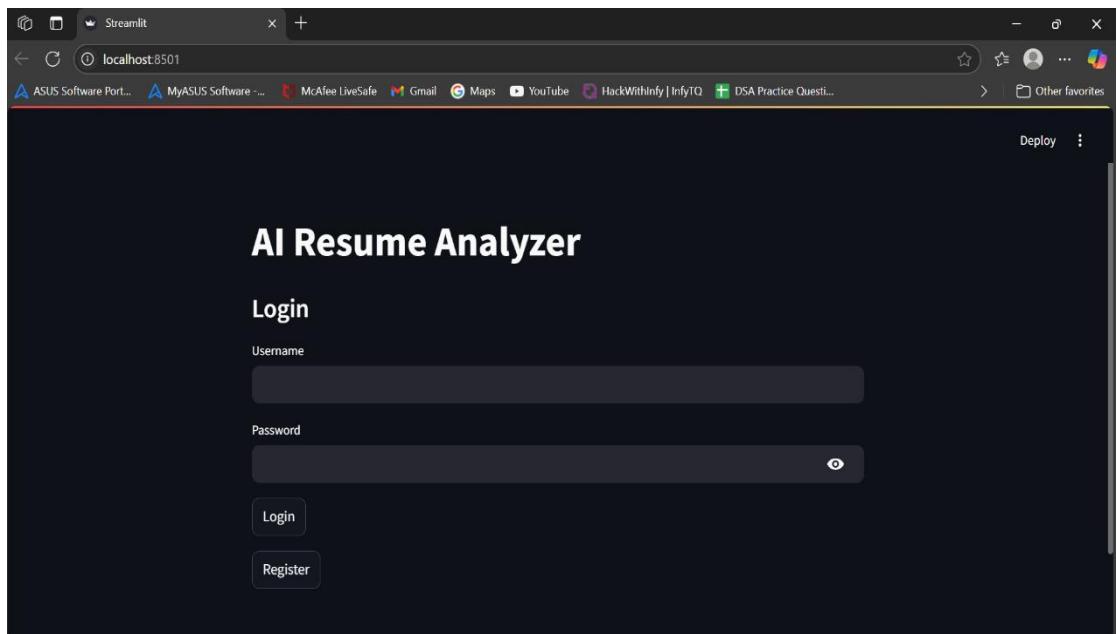
```
                st.write(f" **Fake Check:** {fake_status}")
```

```
                st.write(f" **Improvement Suggestions:** {suggestion}")
```

```
            st.markdown("---")
```

### 4.6. OUTPUT SCREENSHOT

This is the login interface of the AI Resume Analyzer web application, which was designed and implemented using Streamlit. It serves as the secure access gateway for users who wish to use the resume analysis features of the system.



**Figure 4.5.1: Login Page**

This page allows the user (typically an HR professional or recruiter) to input a job description, specify the job role, and upload one or more candidate resumes. The system then analyzes the resumes and matches them against the job requirements using AI-based techniques.



**Figure 4.5.2: Home Page**

This photo shows that the job description has been successfully entered, and multiple resumes have been uploaded for analysis. The system is prepared to evaluate the resumes and provide insights based on the job requirements.

The screenshot displays the user interface of the AI Resume Analyzer. At the top, there is a 'Logout' button. Below it, the title 'Enter Job Description' is centered. Under this title, there is a text input field containing the job description: 'Bachelor's degree in Computer Science, Software Engineering, or a related field. Basic knowledge of programming languages such as Python, Java, or JavaScript. Understanding of version control systems, such as Git. Strong problem-solving skills and attention to detail. Willingness to learn and adapt to new technologies.' Below the job description input, there is a 'Enter Job Role' label followed by a dropdown menu where 'Software Developer' is selected. The next section is titled 'Upload Resumes'. It features a 'Drag and drop files here' area with a cloud icon, which currently contains three resume files: 'Resume CTS word[1].docx' (2.9MB), 'Adarsh Resume.pdf' (0.6MB), and 'bhanu\_resume.docx' (24.6KB). To the right of the file list is a 'Browse files' button. At the bottom of the resume upload section is a 'Match Resumes' button.

Figure 4.5.3: Giving inputs to the Analyzer

### Final Output:

This photo displays the final output of the resume analysis process. It showcases the top matching resumes along with their prediction scores, fake check status, and personalized improvement suggestions. This provides a clear overview of how well each resume aligns with the entered job description, helping in better decision making for candidate selection.

The screenshot displays the final output of an AI resume analyzer. At the top, a green header bar contains the text "Top Matching Resumes:". Below this, three resume entries are listed, each with a file icon, the resume name, a prediction score, a fake check status, and improvement suggestions:

- Resume CTS word[1].docx**  
Prediction Score: 62.8%  
Fake Check: Pass  
Improvement Suggestions: Improve skills related to Software Developer, focus on relevant technologies, and gain more experience.
- Adarsh Resume.pdf**  
Prediction Score: 60.98%  
Fake Check: Pass  
Improvement Suggestions: Improve skills related to Software Developer, focus on relevant technologies, and gain more experience.
- bhanu\_resume.docx**  
Prediction Score: 58.47%  
Fake Check: Pass

**Figure 4.5.3: Final output**

## CHAPTER 5

### CONCLUSION

#### 5.1. CONCLUSION

The AI Resume Analyzer successfully automates the process of analyzing and matching resumes with job descriptions using advanced Natural Language Processing (NLP) techniques. By leveraging TF-IDF vectorization and cosine similarity, the system accurately evaluates the alignment of resumes with specific job roles, providing prediction scores, fake check insights, and personalized improvement suggestions. The tool also integrates a secure login system and supports multiple resume formats, enhancing user experience and data integrity. This project demonstrates an efficient and scalable approach to streamline recruitment processes, offering valuable insights for both recruiters and candidates.

## CHAPTER 6

### FUTURE SCOPE

#### 6.1. FUTURE ENHANCEMENTS

While the current system delivers effective results, there is always room for future development to enhance its capabilities and user experience:

- Web Extension Integration: Transform the platform into a browser extension for seamless and quicker access to resume analysis tools.
- Advanced Fake Check Mechanism: Enhance the fake check system to perform deeper analysis, such as verifying credentials and detecting inconsistencies in provided data.
- Expanded File Format Support: Extend support to additional file formats like HTML and JSON for broader compatibility.
- AI-Driven Suggestions: Incorporate AI-driven suggestions for resume improvement, tailored to specific job roles and industry standards.
- Real-Time Analytics Dashboard: Develop an interactive dashboard for recruiters to view trends, candidate performance, and insights in real-time.
- Cloud Integration: Enable cloud storage for uploaded resumes and job descriptions, ensuring better scalability and access.
- Mobile Application: Create a mobile version of the application for enhanced accessibility and convenience.

These enhancements aim to further strengthen the system's utility, accuracy, and user engagement, making it a comprehensive tool for modern recruitment needs.

## **BIBLIOGRAPHY**

### **References**

1. Sharma, S., Sethi, A., & Shirbhate, A. (2022). Resume Builder & Analyzer. *International Journal of Emerging Technologies and Innovative Research*.
2. Bhatt, A., Uniyal, A., Jyala, D., Mittal, S., Tiwari, P., & Singh, D. (2024, March). Resume Analyzer based on MapReduce and Machine Learning. In *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)* (Vol. 2, pp. 1-5). IEEE.
3. Abhishek, K. L., Niranjanamurthy, M., Aric, S., Ansarullah, S. I., Sinha, A., Tejani, G., & Shah, M. A. (2025). Developing an Intelligent Resume Screening Tool With AI-Driven Analysis and Recommendation Features. *Applied AI Letters*, 6(2), e116.
4. Mirajkar, R. R., Shinde, G. R., Shelke, P. M., Yadav, S., Shinde, B., & Shinde, G. (2024, December). Skill Recommendation System and Resume Analysis using AI. In *2024 IEEE Pune Section International Conference (PuneCon)* (pp. 1-3). IEEE.
5. Prashanth, V. J., Gopinath, S., Udit, S., & Kavitha, C. R. (2024, July). Resume Analyzer and Skill Enhancement Recommender System. In *2024 Asia Pacific Conference on Innovation in Technology (APCIT)* (pp. 1-6). IEEE.
6. Geiger, R. S., O'Sullivan, F., Wang, E., & Lo, J. (2025). Asking an AI for salary negotiation advice is a matter of concern: Controlled experimental perturbation of ChatGPT for protected and non-protected group discrimination on a contextual task with no clear ground truth answers. *PloS one*, 20(2), e0318500.
7. Wiles, E. B. (2024). *Artificial Intelligence in Labor Market Matching* (Doctoral dissertation, Massachusetts Institute of Technology).
8. Revillod, G. (2024). Implementation of AI Recruitment Systems in Swiss HRM: The Importance of Technological and Organizational Factors. *Journal of Human Resource Management*, 27(2), 95-122.
9. Gangoda, N., Yasantha, K. P., Sewwandi, C., Induvara, N., Thelijjagoda, S., & Giguruwa, N. (2024, October). Resume

- Ranker: AI-Based Skill Analysis and Skill Matching System.  
In *2024 Sixth International Conference on Intelligent Computing in Data Sciences (ICDS)* (pp. 1-8). IEEE.
10. Ather, D., Singh, G., Chaudhary, N., Verma, P., Kler, R., & Gupta, A. (2024, November). Analyzing Trends, Skills Demand, and Salary Prediction in the AI and ML Job Market. In *2024 International Conference on Intelligent & Innovative Practices in Engineering & Management (IIPEM)* (pp. 1-8). IEEE.

# AI Resume Analyzer for improvement of Employability among graduates

by 24091D0504 YADIKI NEHA

**Submission date:** 25-Jun-2025 12:40PM (UTC+0530)

**Submission ID:** 2705751692

**File name:** Analyzer\_for\_improvement\_of\_Employability\_among\_graduates\_1.pdf (1.83M)

**Word count:** 11244

**Character count:** 68702

NH

25/06/2025

D. D. CATYANARAYANA, B.E., M.Tech., Ph.D.  
Controller of Examinations  
Rajeev Gandhi Memorial College of  
Engineering & Technology (Autonomous)  
NANDYAL-518 501, Nandyal(Dt.), A.P.)

# AI Resume Analyzer to Improve Employability Among Graduates

## ORIGINALITY REPORT

**23%**  
SIMILARITY INDEX

**20%**  
INTERNET SOURCES

**6%**  
PUBLICATIONS

**18%**  
STUDENT PAPERS

## PRIMARY SOURCES

- 1 Submitted to Arab Open University  
Student Paper 3%
- 2 ijiemr.org  
Internet Source 3%
- 3 www.ijraset.com  
Internet Source 1 %
- 4 Submitted to University of Wales Institute,  
Cardiff  
Student Paper 1 %
- 5 Submitted to University of Science and  
Technology of Fujairah  
Student Paper 1 %
- 6 answersdrive.com  
Internet Source 1 %
- 7 Submitted to University of North Texas  
Student Paper 1 %
- 8 huggingface.co  
Internet Source 1 %

*Dr. D. SATYANARAYANA, B.E., M.Tech., Ph.D.*  
*Controller of Examinations*  
*Rajiv Gandhi Memorial College of*  
*Engineering & Technology (Autonomous)*  
*Hyderabad, Andhra Pradesh, India (D)*

*25/26/25/25*

*SM*



A DIGITAL LIBRARY

WWW.IJISRT.COM

# INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

## AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, TITLED  
Resume Inflation System for Improvement of Employability  
Among Graduates

AUTHORED BY  
Dr. R. Kaviarasan

HAS BEEN PUBLISHED IN  
Volume 10 | Issue 6 | June -2025

ARTICLE DIGITAL NO.  
IJISRT25JUN284



EDITOR IN CHIEF IJISRT

This document certifies that the manuscript listed above was submitted by above said respected author  
To verify the submitted manuscript please visit our official website: [www.ijisrt.com](http://www.ijisrt.com)  
Or Email us: [editor@ijisrt.com](mailto:editor@ijisrt.com)



A DIGITAL LIBRARY

WWW.IJISRT.COM

# INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

## AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, TITLED  
Resume Inflation System for Improvement of Employability  
Among Graduates

AUTHORED BY  
Y. Neha

HAS BEEN PUBLISHED IN  
Volume 10 | Issue 6 | June -2025

ARTICLE DIGITAL NO.  
IJISRT25JUN284



EDITOR IN CHIEF IJISRT

This document certifies that the manuscript listed above was submitted by above said respected author  
To verify the submitted manuscript please visit our official website: [www.ijisrt.com](http://www.ijisrt.com)  
Or Email us: [editor@ijisrt.com](mailto:editor@ijisrt.com)

# Resume Inflation System for Improvement of Employability among Graduates

Dr. R. Kaviarasan<sup>1</sup>; Y. Neha<sup>2</sup>

<sup>1</sup>Associate Professor; <sup>2</sup>M.Tech Student

<sup>1</sup>Department of Computer Science Engineering,  
Rajeev Gandhi Memorial College of Engineering & Technology, Andhra Pradesh, India.  
<sup>2</sup>Department of Computer Science Engineering,  
Rajeev Gandhi Memorial College of Engineering & Technology, Andhra Pradesh, India.

Publication Date: 2025/06/13

**Abstract:** A Resume Inflation System proposed in this research is a progressive, computer-based framework engineered to optimise the personnel recruitment cycle by facilitating hands-free operation and performance review of job seekers' resumes. In the recruitment cycle, hiring managers are often flooded with numerous resumes from applicants. This large volume of applicants can lead to difficulties in selecting candidates for various positions. However, there is no open-source application that assists in shortlisting resumes for a particular position directly using the job description. This inspires us to introduce a methodology for a resume Inflation system. The resume Inflation system mitigates these issues by introducing the initial review activity, which reduces the time spent on manual screening by up to 80%. This Inflation system employs machine learning models and NLP to proficiently inspect resumes, extract relevant skills, experiences, and qualifications, and match them against specific job descriptions with maximum accuracy. The procedure ensures that the most promising applicants are prioritised for further rounds of evaluation, which enhances the standards of shortlisted applicants and decreases the risk of mismatch. The Resume Inflation System ensures a fairer recruiting procedure and nurtures diversity in the hiring process. This latest application not only saves time but also increases the efficiency and accuracy of candidate selection.

**Keywords:** ML Models, NLP, AI, Performance Review, Inflation System, Computer-Based Framework, Application Tracking System (ATS).

**How to Cite:** Dr. R. Kaviarasan; Y. Neha (2025) Resume Inflation System for Improvement of Employability among Graduates. *International Journal of Innovative Science and Research Technology*, 10(6), 426-433.  
<https://doi.org/10.38124/ijisrt/25jun284>

## I. INTRODUCTION

The project is initiated to minimise the problems occurring in the recruitment process by hiring managers in large organisations like Microsoft. Such organisations are substantial, often resulting in a considerable number of job openings across various job profiles. Labelling the resumes received from job seekers for these profiles is a significant responsibility for hiring managers. Shortlisting these resumes for further review poses an additional challenge for recruiters.

Both recruiters and job seekers face challenges in these processes, as job seekers aim to optimise their resumes for a better chance of being shortlisted for the next rounds. Each applicant must write their resume to align with the job description for which they are applying. Job seekers also need an application that will guide them in creating their resumes and examining them against the job description corresponding to the job profile they are applying for.

Some hiring managers are looking through various platforms, such as "LinkedIn" and "Unstop," to find new, talented job seekers, but they find a lot of applications. This could make it time-consuming for the recruiters to manually review the resumes for the next steps of the recruitment process. As an applicant, he or she wants to know if the resume is sufficient. As a result, creating a solution that enhances resume quality is essential.

While some websites assess resumes for job seekers, they often lack detailed guidance on the modifications necessary for better results and shortlisting. Given the inadequacy of existing platforms to conduct thorough analyses on resume datasets, developing an effective and contemporary application that employs the latest advancements in artificial intelligence is essential. Taking this as a core problem, we have developed this resume inflation system that accurately shortlists the resumes that address the skills required in today's market.

## II. LITERATURE SURVEY

### ➤ Existing System:

AI-based resume analysis systems usually automate screening through rule-based filtering and keyword matching, generally limited by pre-established templates, using machine learning and natural language processing techniques. Although sophisticated algorithms improve data extraction by analyzing resumes for education, experience, and abilities, they have trouble understanding semantic context and managing a variety of resume formats. Although deep learning methods require a lot of computer power and huge labelled datasets, they provide better semantic matching. Furthermore, real-time feedback features that would assist candidates in optimizing their resumes for applicant tracking systems (ATS) are absent from the majority of current systems.

### ➤ Disadvantages of the Existing System:

- Context Misinterpretation: Qualified candidates may be rejected as a result of keyword-based filtering that misinterprets context.
- Inflexibility: Unconventional profiles and various resume formats frequently cause problems for current systems.
- High Resource Demand: Small-scale applications of deep learning models are restricted due to their high processing power and large dataset requirements.
- Absence of Real-Time Feedback: Applicants do not get real-time feedback on how to improve their resumes for greater interoperability with applicant tracking systems.

### ➤ Proposed System:

By combining NLP approaches with machine learning models, the proposed AI Resume Analyzer aims to overcome the shortcomings of the existing resume screening systems. Its main function is to give candidates real-time feedback so they may successfully optimize their resumes for ATS compatibility. In contrast to conventional keyword-based techniques, the analyzer uses semantic analysis to precisely extract and comprehend important data from a range of resume formats, including education, experience, and abilities. This lowers the possibility of missing competent people with unusual backgrounds by ensuring a

more comprehensive and context-aware assessment of applicant profiles.

The system makes use of machine learning models that have been trained on a variety of industry-relevant datasets to increase its efficacy and enable it to adjust to various job positions and industries. The analyzer helps candidates customize their resumes for particular job descriptions by offering actionable information in addition to resume parsing, such as formatting and keyword optimization tips. The ultimate objective is to create a user-friendly, scalable platform that enhances the relationship between recruiters and job seekers by improving the efficiency, inclusivity, and accuracy of the resume evaluation process.

### ➤ Advantages of the Proposed System:

The system has been analyzed in terms of cost, processing speed, requirements, level of expertise, and availability of resources. Parameter Identified:

- Accurate extraction of skills, experience, and education details
- Semantic analysis for better matching with job descriptions
- Real-time feedback for resume optimization
- ATS compatibility analysis
- Handling of diverse and unconventional resume formats
- Identification of redundant or missing information
- Contextual keyword analysis
- Formatting recommendations for clarity and readability
- Scalability for large-scale resume analysis
- Secure handling of personal data and privacy compliance

## III. METHODOLOGY

The system architecture of the Resume Inflation System is developed to address the problems of young graduates. As the primary round of the recruitment cycle is a resume shortlisting, they have to make effective resumes, for which they need guidance and suggestions. This architecture uses ML and NLP methods, which help the system produce efficient results. The system architecture can be seen in the following image:

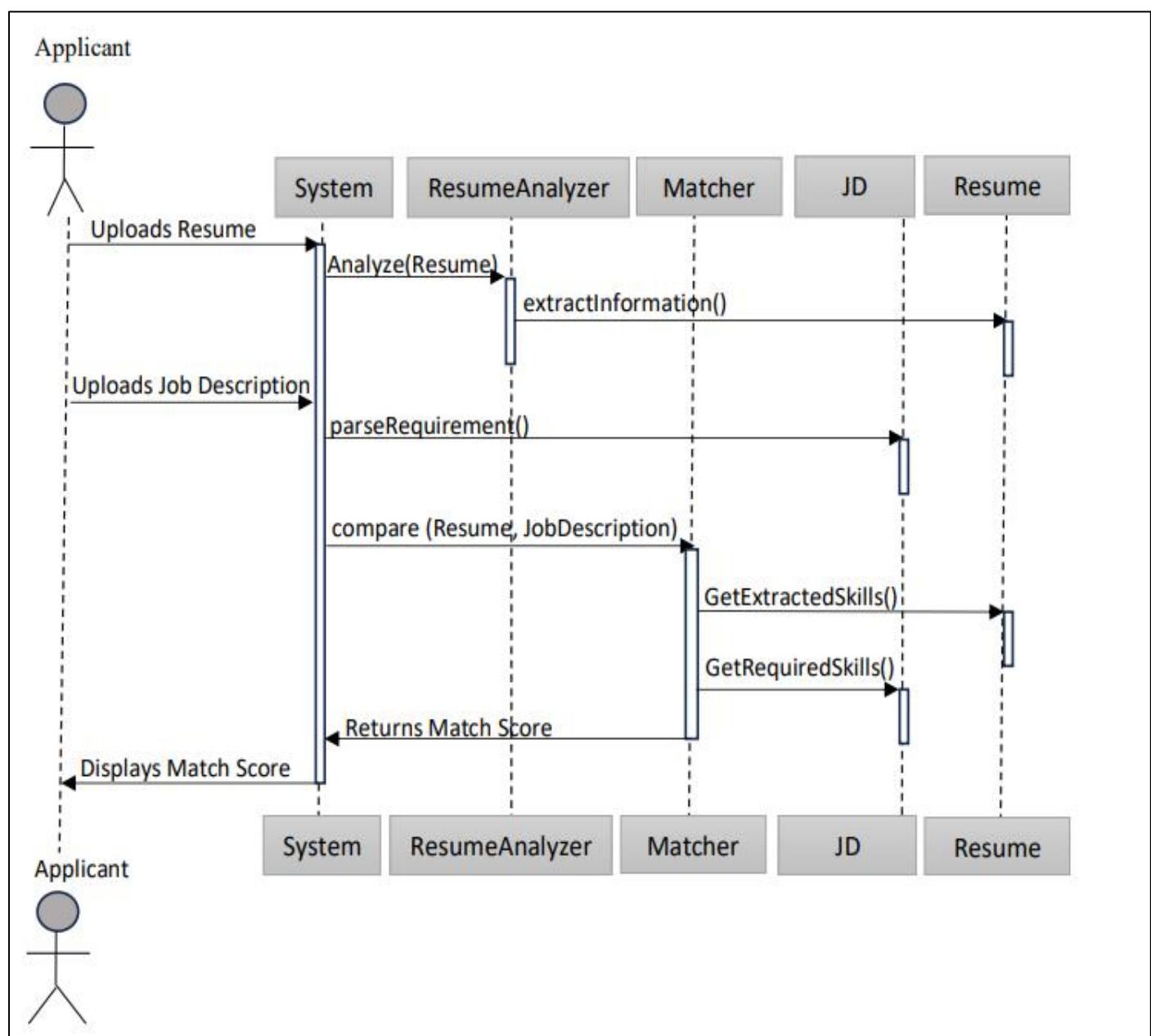


Fig 1 System Architecture of the Resume Inflation System

The system architecture has a few components which involved in developing our Resume Inflation System. These components have made it possible to go from having a simple resume and a job description to identifying which resumes match which job description and what drawbacks the resumes lack in fulfilling the demands of the job description. The final result of this architecture is a match score for each resume, which is displayed as the output on our application along with the suggestions provided to the resumes that lack scoring maximum score because of inefficient skills present in that particular resume. Each function present in the system architecture is explained in detail as follows:

#### ➤ Application

The primary level is the application level in the architecture, which gives the input and receives the output. Here, the input can be a single resume or a group of resumes

and a job description. This level receives match scores of the resumes as the output and displays the match score of each resume corresponding to the job description, along with suggestions for each resume. The invalid document formats are also detected using our system, and the output for this would be “invalid document format”.

#### ➤ System

The System component receives resumes and job descriptions from the application level. Here, the resume is parsed to extract important features using the `parseRequirements()` function. Extraction of these features is essential, as in the next step, we need to compare them with the job description to test if these extracted features match the job description or not. Using the `compare(Resume, JobDescription)` function, we have tested the resume against the job description.

➤ *Resume Analyzer*

The resume analyzer performs only the analysis of the resume using the analyze(resume) function and produces the important features. These features can be returned using the extract information function by using the extractinformation() function.

➤ *Matcher*

The Matcher component receives the results of the compare(Resume, JobDescription) function. Here, we have included two functions, such as GetExtractedSkills() and GetRequiredSkills(). The GetExtractedSkills() function is used to get the skills extracted from the resumes, and the GetRequiredSkills() function is used to get the required skills that are extracted from the job description. These skills are matched in this matcher component to generate the results.

➤ *Job Description (JD)*

The job description component is included in the architecture to provide the job description to the other components. These components require the job description to perform various functions. As we can observe, some of the functions like parseRequirements(), compare(Resume, JobDescription), and GetRequiredSkills() need a job description to be executed.

➤ *Resume*

The resume component of the architecture is involved as it is the main part of the architecture. The resumes are the components with which we are working. This component gives access to use the resumes in a few other components to execute their functions. The functions that require the resumes are Analyze(Resume), extractinformation(), compare(Resume, JobDescription), and GetExtractedSkills.

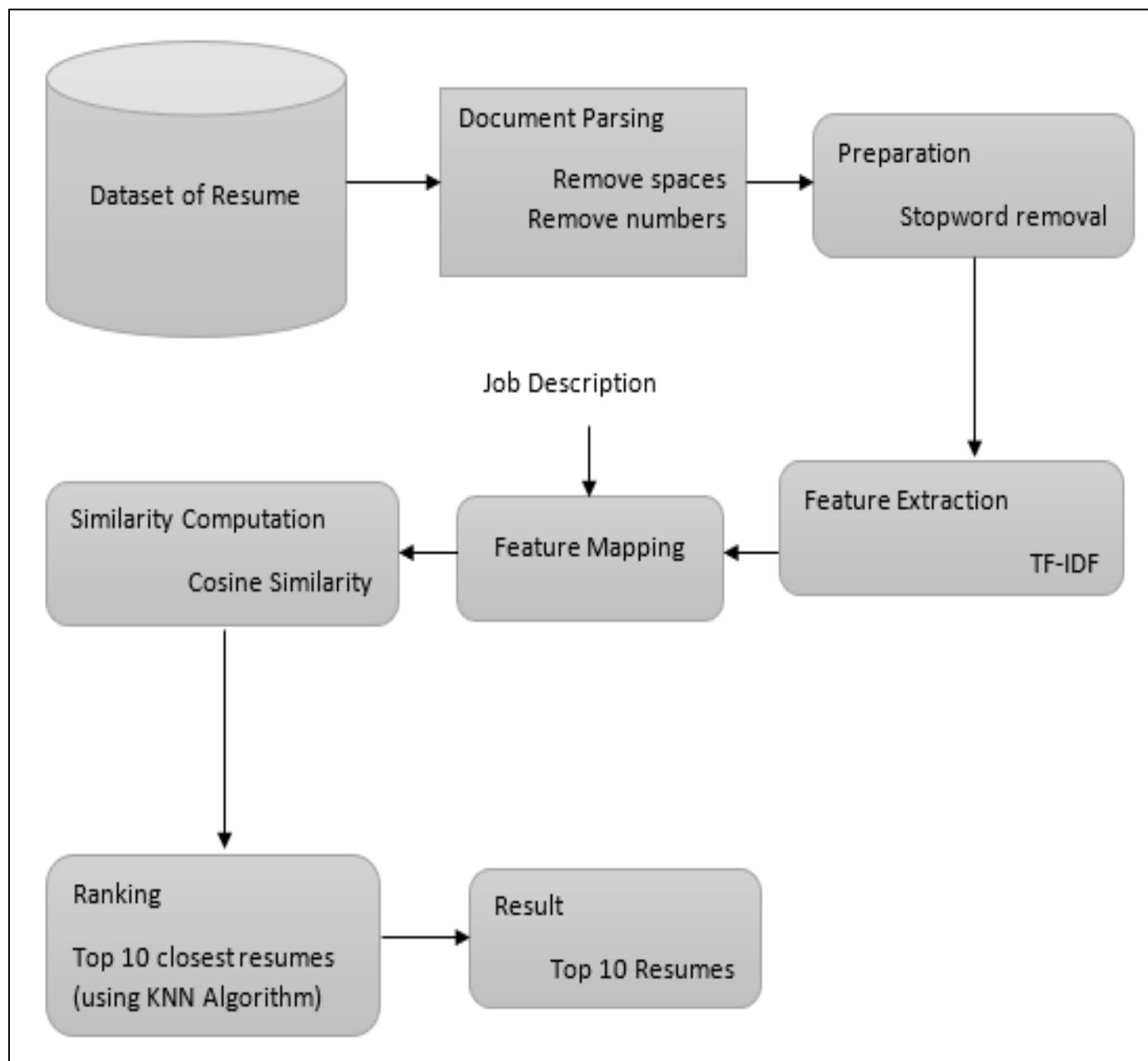


Fig 2 Activity Workflow of Resume Inflation System

Our project's workflow can be presented in the flowchart format, where each step in this flowchart explains the process of designing our project. Each step introduces various new technologies that help in finding effective results. We can express the details of our workflow by detailing the activities of this flowchart in the form of an algorithm. The actual algorithm is developed based on this activity workflow. The algorithm of our project can be represented as follows:

#### ➤ Activity Workflow/Algorithm

- **Stage 1: Input Data**

The Input of our system would be a single resume or a group of resumes and a job description to match with.

✓ Case 1: A single resume is given when the user (job seeker) wants to test his/her resume against the job description for guidance in creating a better resume, as our application will provide suggestions.

✓ Case 2: A group of resumes is given when the user (recruiter) wants to find the best resumes that match the job descriptions.

- **Stage 2: Document Parsing**

The resume is parsed to extract important features like skills and the qualifications of the applicant, and remove irrelevant data.

- **Stage 3: Preparation**

In any document, there will be stop words. Here, we don't have any use for these stop words. So we will remove them in this stage to avoid spending time on this unnecessary data.

- **Stage 4: Feature Extraction**

To produce the match score of each resume, we need to match the features of the resumes with the job description. So, we have extracted the important features of each resume that represent the strength of the job seeker. Here, we have used one of the NLP measures to perform feature extraction.

✓ Functionality: NLP is a commonly used technology to extract important features from various types of documents.

Here, we have to extract features of a resume, so we have used the TD-IDF measure to perform this function.

This measure can be represented as follows:

$$TF - IDF(t, d) = TD(t, d) * IDF(t)$$

Where:

$$TF(t, d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

$$IDF(t) = \log \frac{\text{Total number of documents}}{\text{Number of documents containing term } t}$$

- **Stage 5: Feature Mapping**

The term itself says that it is going to map the extracted feature of the resume with the job description to fine the match score.

- **Stage 6: Similarity Computation**

To generate the result, we need to check the similarity of the resume with the job description. This can be done using the cosine similarity measure.

✓ Functionality: Cosine similarity measure used to find the similarity between two documents. Here, we consider one document as a resume and another as a job description. The cosine similarity can be represented as follows:

$$\text{Cosine Similarity} = \cos(\theta) = \frac{A \cdot B}{\| A \| \| B \|}$$

- **Stage 7: Ranking**

✓ Activity: Using the KNN (K-Nearest Neighbour) algorithm, rank the resumes according to the skill set listed in the job description. The top ten resumes most closely fitting the job description are selected here.

✓ Functionality: The KNN algorithm groups the resumes using a distance measure known as Euclidean distance. In this case, the resumes are viewed as data points, the distance between them is computed, and all of the resumes are grouped based on shared characteristics that correspond to the job description. The following is a representation of the Euclidean distance formula.

- **Euclidean Distance:**

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

- **Phase 8: Outcome/Result**

✓ Activity: The top ten resumes that match are listed as the output.

#### IV. PERFORMANCE EVALUATION

➤ In the Following Steps, you can Observe the Performance of the Resume Inflation System:

- The Login Page will Look like this once our Software has Run:

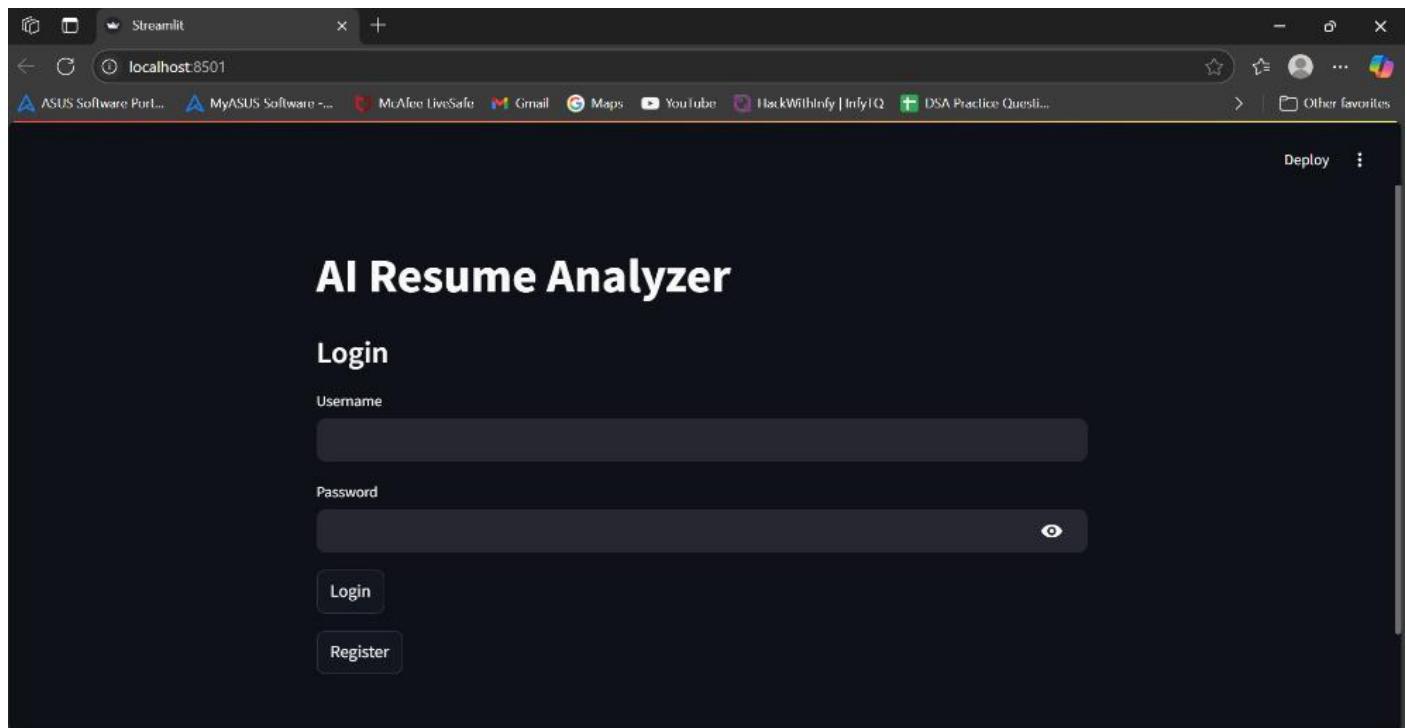


Fig 3 Login Page

- To begin, we will click 'Register' on this login screen to create a user account, as we do not yet have one. The page for registration will then be shown.

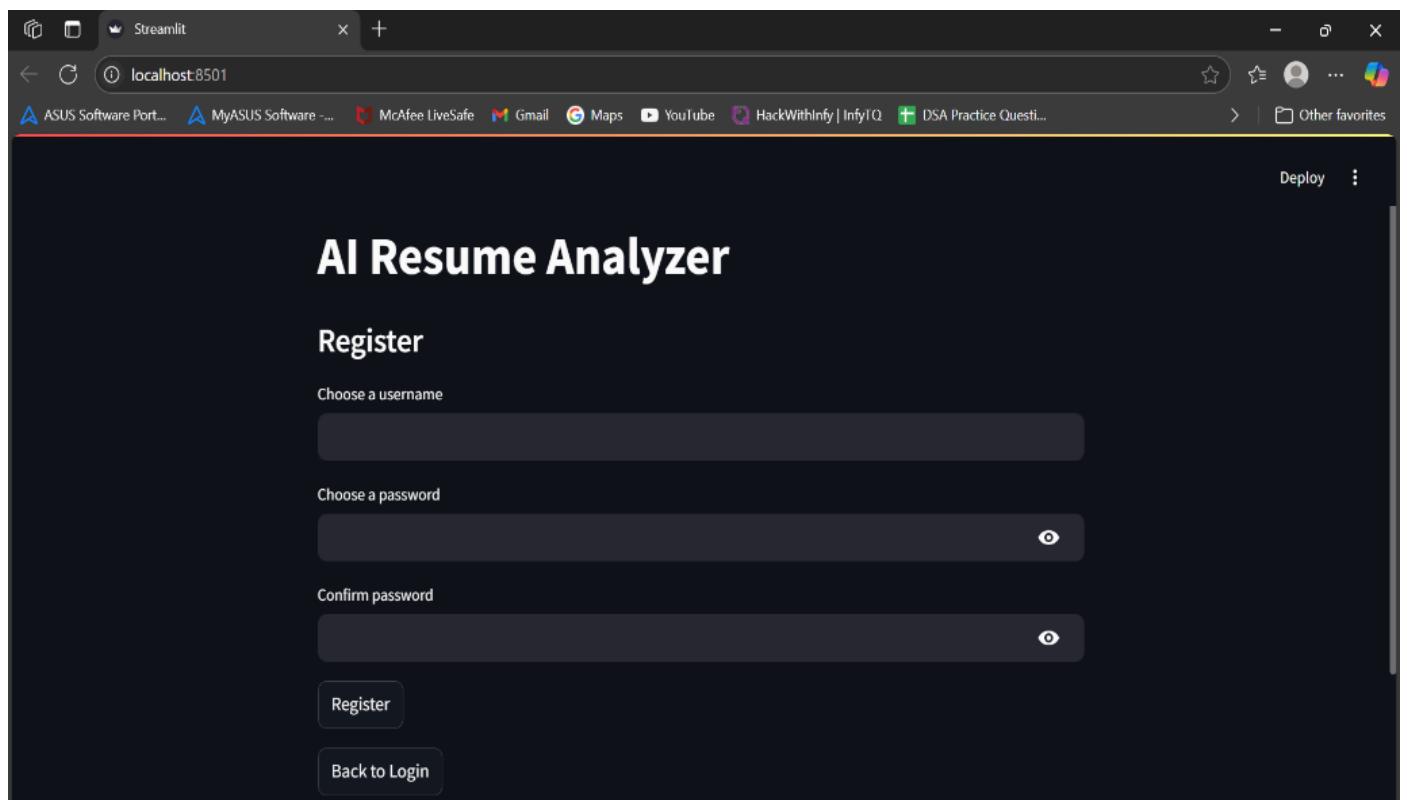


Fig 4 Registration Page

- After Registering Successfully, we must Log in and Supply the Necessary input Files.

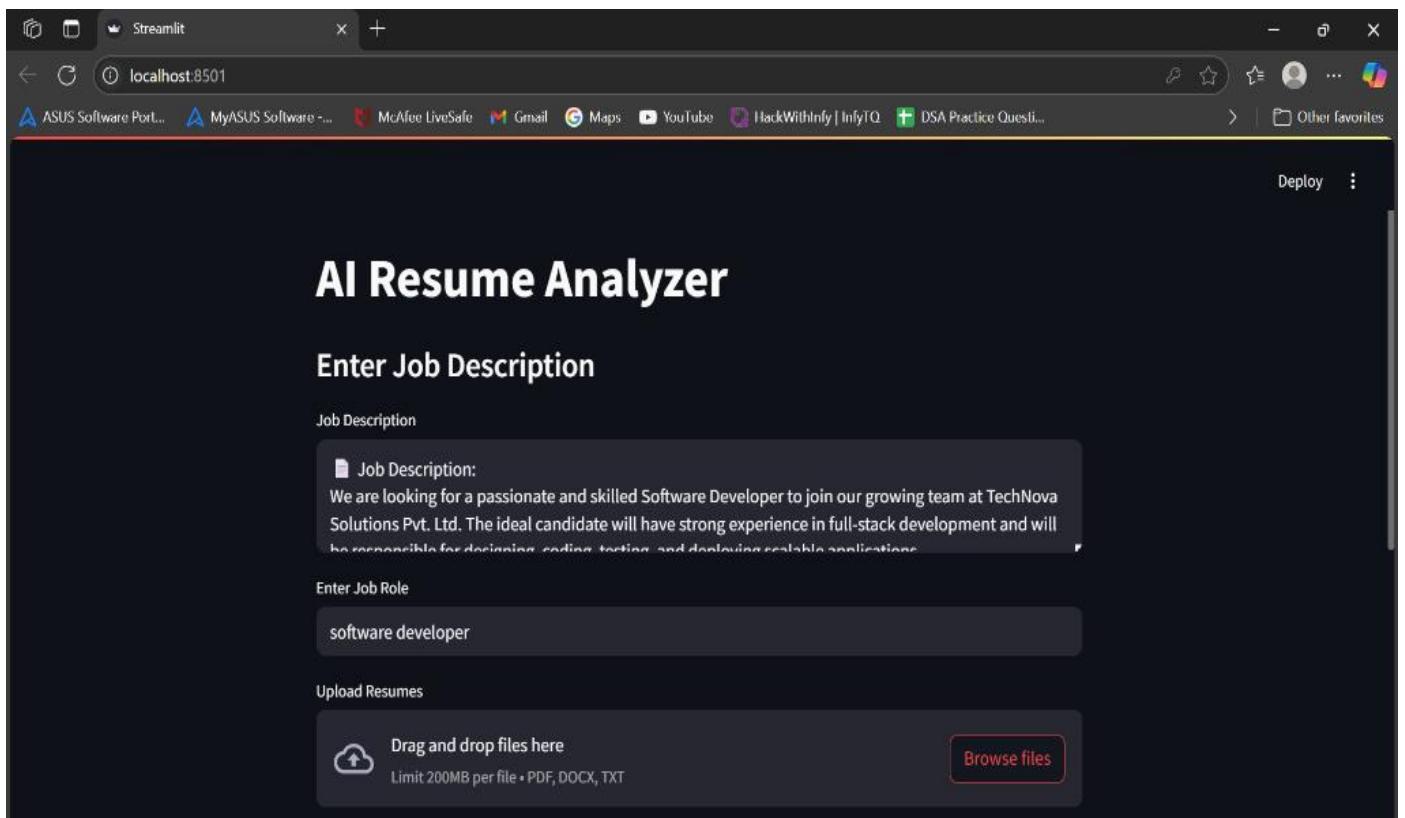


Fig 5 Giving Inputs to the Analyzer

- Once the Input Files have been analysed, our System will Show the output along with Recommendations.

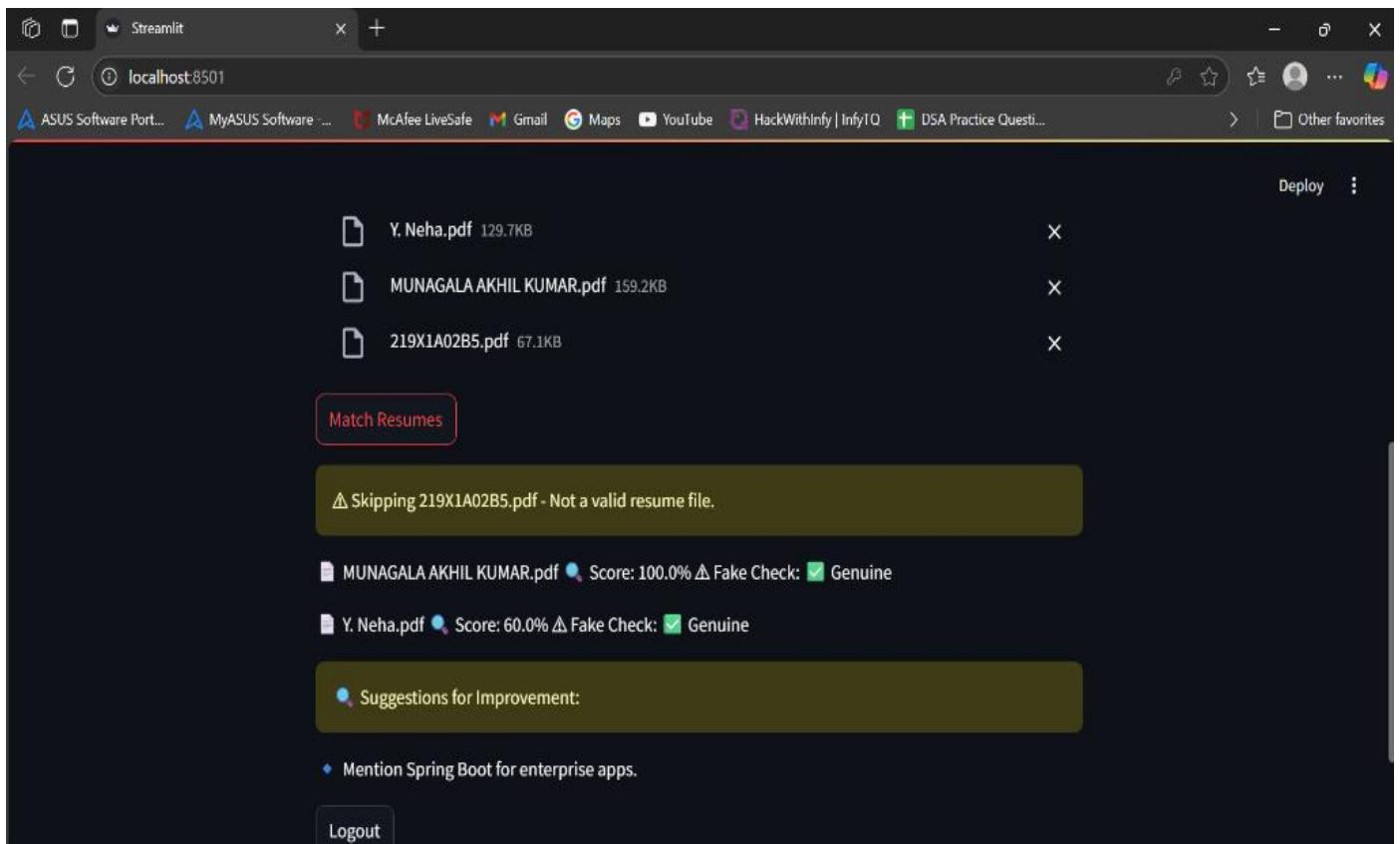


Fig 6 Output Page

## V. CONCLUSION AND FUTURE SCOPE

The challenge of identifying the best employee through resume analysis has become increasingly difficult. Conversely, job seekers are encountering obstacles in finding a suitable job. The core issue lies in the initial stage of recruitment, which relies on resumes. At this primary recruitment stage, hiring managers select candidates by shortlisting resumes; thus, job seekers must focus on crafting effective resumes. This Resume Inflation system simplifies the process for job seekers by analysing their resumes against the job descriptions they are applying for and offering suggestions to enhance their resumes for better chances of being shortlisted. This system is also beneficial for hiring managers, as it can shortlist resumes based on the skillset outlined in the job description.

## REFERENCES

- [1]. Elham Albaroudi, Taha Mansouri, "A Comprehensive Review of AI Techniques for Addressing Algorithmic Bias in Job Hiring", in AI journal Volume 5, Issue 1, MDPI 2024.
- [2]. Ashvini Chavan, Nikita Tatewar, "AI Resume Analyzer", in the International Journal of Creative Research Thoughts (IJCRT), December 2023.
- [3]. Yiou Lin, Hang Lei, "Machine learned resume-job matching solution", in International Conference on Artificial Intelligence and Data Engineering, 2021.
- [4]. Hameed, K.; Arshed, N.; Yazdani, N.; Munir, M. On globalization and business competitiveness: A panel data country classification. Stud. Appl. Econ. 2021, 39, 1–27. [CrossRef]
- [5]. Farida, I.; Setiawan, D. Business strategies and competitive advantage: The role of performance and innovation. J. Open Innov. Technol. Mark. Complex. 2022, 8, 163. [CrossRef]
- [6]. Yi-Chi Chou, Han-Yen Yu, " Based on the application of AI technology in resume analysis and job recommendation " in IEEE International Conference on Computational Electromagnetics, 2020, pp. 293-296.
- [7]. J. Singh and S. Singh, "A natural language processing approach to extracting information from resumes," in 5th International Conference on Industrial Engineering and Applications (ICIEA), 2018.
- [8]. Rimitha, S. R., Abburu, V., Kiranmai, A., & Chandrasekaran, K "Ontologies to Model User Profiles in Personalized Job Recommendation, " In 2018 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), pp. 98-103, 2018.
- [9]. Kim, S., et al. "Personalized Recommendation Systems for Job Vacancies." IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 6, 2019, pp. 1175-1187.
- [10]. Amato F., Boselli R., Cesarini M., et al. Challenge: Processing web texts for classifying job offers[C]/Semantic Computing (ICSC), 2015 IEEE International Conference on. IEEE, 2015: 460-463.