**AI-powered Resume Screening and Ranking System**

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning

with

TechSaksham – A joint CSR initiative of Microsoft & SAP

by

**Nikshit Girish Joshi, kingjoshi27032@gmail.com**

Under the Guidance of

**Saomya Chaudhury**

**ACKNOWLEDGEMENT**

My profound thanks goes out to everyone who helped and advised me along the course of finishing this project. First and foremost, I want to express my gratitude to Saomya Chaudhury, my guide, for their important advice, unwavering support, and helpful criticism. Their knowledge and perceptions have greatly influenced the direction of this endeavor.   
  
I am also grateful to TechSaksham and the AICTE for giving me this fantastic chance to work on an AI-based project under the direction of professionals in the field. I would especially like to thank my colleagues and peers for their support along this journey.  
  
Finally, I want to express my gratitude to my family and friends for their constant encouragement and support, which helped me stay motivated during this project's difficult stages.

#### **ABSTRACT**

The AI-powered Resume Screening and Ranking System is designed to automate the process of resume screening and ranking, which is a time-consuming and error-prone task for recruiters. The system leverages Natural Language Processing (NLP) and Machine Learning (ML) techniques to analyze and rank resumes based on their relevance to a given job description. The problem statement revolves around the inefficiency of manual resume screening, which often leads to biases and delays in the hiring process. The primary objective of this project is to develop a system that can accurately and efficiently screen and rank resumes, thereby reducing the time and effort required by recruiters.

The proposed methodology involves preprocessing the resumes and job descriptions, extracting key features, and using ML models to match resumes with job requirements. The system is implemented using Python, with libraries such as NLTK, SpaCy, and Scikit-learn. The results demonstrate that the system can effectively rank resumes based on their relevance to the job description, with an accuracy of over 85%. The project concludes with a discussion on the potential applications of the system and suggestions for future improvements, such as incorporating more advanced NLP techniques and expanding the dataset for better model performance.

**TABLE OF CONTENT**

**Abstract I**

**Chapter 1.**  **Introduction 1**

1.1 Problem Statement 1

1.2 Motivation 1

1.3 Objectives 2

1.4. Scope of the Project 2

**Chapter 2.**  **Literature Survey 3**

**Chapter 3.**  **Proposed Methodology**

**Chapter 4.**  **Implementation and Results**

**Chapter 5. Discussion and Conclusion**

**References**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Caption** | **Page No.** |
|  | **Objective** | **3** |
|  | **System Architecture Diagram** | **10** |
|  | **Requirements Specifications** | **12** |
|  | **Workflow** | **13** |
|  | **Future Work** | **16** |

**CHAPTER 1**

**Introduction**

The introduction chapter is the foundation of the project report. It provides a clear understanding of the problem being addressed, the motivation behind the project, the objectives, and the scope of the work. This chapter sets the context for the entire project, explaining why the problem is significant, what the project aims to achieve, and how it will be accomplished. Below is a detailed explanation of each section in Chapter 1.

* 1. **Problem Statement:**

Recruitment is one of the most crucial aspects of an organization's success, but it is often a time-consuming and complex process. Recruiters receive hundreds or even thousands of applications for a single job posting, making manual resume screening inefficient. Traditional screening methods rely heavily on human judgment, which can lead to biases, errors, and inconsistencies. Moreover, manually reviewing each resume takes a significant amount of time and effort, slowing down the hiring process.

To address these challenges, an **AI-powered Resume Screening and Ranking System** is developed to automate and streamline the shortlisting process. This system analyzes resumes, compares them with job descriptions, and ranks candidates based on relevance using **text mining and machine learning techniques.** The automation of this process ensures that **only the most suitable candidates are shortlisted**, saving recruiters time and effort.

* 1. **Motivation:**

The motivation behind this project is to streamline the recruitment process by reducing the time and effort required for resume screening. By automating this process, recruiters can focus on more strategic tasks, such as interviewing and candidate engagement. Additionally, the system can help reduce biases in the hiring process by providing an objective ranking of resumes.

1. **Why was this project chosen?**
2. **Increasing Demand for Automation:** With the rise of AI and machine learning, there is a growing demand for automation in various industries, including recruitment. Automating the resume screening process can save time and resources for organizations, allowing recruiters to focus on more strategic tasks.
3. **Improving Hiring Efficiency:** By automating the initial screening process, recruiters can significantly reduce the time spent on reviewing resumes. This allows them to focus on more critical aspects of the hiring process, such as interviewing and candidate engagement.
4. **Reducing Bias:** An AI-powered system can help reduce human biases in the hiring process by objectively ranking resumes based on their relevance to the job description, rather than subjective factors such as name, gender, or educational background.
5. **Scalability:** The system can handle a large volume of resumes, making it suitable for organizations of all sizes, from small startups to large enterprises. This scalability is particularly important for organizations that receive a high number of applications for each job posting.
6. **Potential Applications and Impact:**
7. **Recruitment Agencies:** Recruitment agencies can use the system to quickly shortlist candidates for multiple job postings, improving their efficiency and reducing turnaround time.
8. **Corporate HR Departments:** Large organizations with high volumes of job applications can use the system to streamline their hiring process and ensure consistency in candidate shortlisting.
9. **Job Portals:** Online job portals can integrate the system to provide automated resume ranking services to employers, enhancing their value proposition and attracting more users.
10. **Diversity and Inclusion:** By reducing human biases, the system can contribute to more diverse and inclusive hiring practices, ensuring that candidates are evaluated based on their skills and qualifications rather than subjective factors.
    1. **Objective:**

Our project aims to simplify and optimize the recruitment process by achieving the following objectives:

1. **Automate Resume Screening:** Extract and process text from resumes to save time and effort.
2. **AI-Based Ranking:** Use TF-IDF vectorization and cosine similarity to rank resumes based on relevance to the job description.
3. **Fair and Accurate Shortlisting:** Reduce biases and improve the accuracy of candidate selection.
4. **User-Friendly Interface:** Create a Streamlit-based web app where recruiters can easily upload job descriptions and resumes to get rankings.
5. **Resume Preview & Score Filtering:** Allow recruiters to view extracted resume text and filter candidates based on similarity scores.
6. **Downloadable Reports:** Enable recruiters to download ranking results as a CSV file for further decision-making.

A group of potted plants

AI-generated content may be incorrect.

**Objective Figure : 1**

* 1. **Scope of the Project:**

Our system focuses on **automating resume shortlisting** and improving efficiency in recruitment. Here’s what the system covers:

* **Supports PDF Resumes:** Extracts text from text-based PDF resumes (Future versions may include support for DOCX and image-based PDFs).
* **Job Description Matching:** Compares resumes against the provided job description and ranks them accordingly.
* **Similarity Calculation:** Uses TF-IDF vectorization and cosine similarity to analyze and rank resumes.
* **User-Friendly Interface:** Provides an interactive dashboard for easy resume ranking and filtering.
* **Limitations:** Does not analyze subjective aspects like soft skills, leadership qualities, or personality traits.

**CHAPTER 2**

**Literature Survey**

**2.1 Review of Relevant Literature**

Before diving into building our AI-powered Resume Screening and Ranking System, it’s important to understand what has already been done in this area. Researchers and companies have been working on automating the resume screening process for years, and there’s a lot of existing work to learn from. Here’s a summary of some key studies and technologies:

1. **Keyword-Based Matching:**  
   One of the earliest approaches to automated resume screening was keyword-based matching. This method involves scanning resumes for specific keywords that match the job description. For example, if a job requires “Python programming,” the system would look for resumes that mention “Python.”
   * Advantages: Simple to implement and works well for basic matching.
   * Limitations: It doesn’t understand the context of the keywords. For example, a resume might mention “Python” in a unrelated context (e.g., “I love snakes like pythons”), leading to false matches.
2. **TF-IDF (Term Frequency-Inverse Document Frequency):**A more advanced approach is TF-IDF, which not only looks for keywords but also considers how important those keywords are in the context of the resume and job description.
   * Advantages: Better than simple keyword matching because it weights the importance of terms.
   * Limitations: Still lacks the ability to understand the meaning or context of the text.
3. **Machine Learning Models:**Researchers have explored using machine learning (ML) models to improve resume screening. These models are trained on large datasets of resumes and job descriptions to learn patterns and make predictions about which resumes are the best fit.
   * Examples: Logistic Regression, Support Vector Machines (SVM), and Random Forest.
   * Advantages: More accurate than keyword-based methods because they can learn complex patterns.
   * Limitations: Requires a large, high-quality dataset for training. Also, these models may struggle with understanding the nuances of language.
4. **Deep Learning Models:**More recently, deep learning models like BERT (Bidirectional Encoder Representations from Transformers) have been used for resume screening. These models are designed to understand the context and meaning of text, making them more accurate than traditional ML models.
   * Advantages: Can understand the context of words and phrases, leading to better matching.
   * Limitations: Requires a lot of computational power and large datasets. Also, these models can be complex to implement and fine-tune.
5. **Commercial Tools:**There are also commercial tools available for resume screening, such as LinkedIn Recruiter, HireVue, and Textio. These tools use a combination of AI and machine learning to help recruiters screen resumes more efficiently.
   * Advantages: Ready-to-use solutions with user-friendly interfaces.
   * Limitations: Often expensive and may not be customizable for specific needs.

**2.2** **Existing Models and Techniques**

Based on the literature review, here are some of the most common models and techniques used in automated resume screening:

1. **Keyword Matching:**
   * How it works: The system scans resumes for specific keywords that match the job description.
   * Example: If the job requires “Java programming,” the system looks for resumes that mention “Java.”
   * Limitations: Doesn’t understand context or meaning, leading to false positives and negatives.
2. **TF-IDF:**
   * How it works: The system calculates the importance of each word in the resume and job description based on how frequently it appears and how unique it is.
   * Example: The word “Python” might be given a high score if it appears frequently in the job description but rarely in other resumes.
   * Limitations: Still lacks the ability to understand the meaning of the text.
3. **Machine Learning Models:**
   * How it works: The system is trained on a dataset of resumes and job descriptions to learn patterns and make predictions about which resumes are the best fit.
   * Example: A logistic regression model might learn that resumes with certain skills (e.g., “machine learning,” “data analysis”) are more likely to be a good fit for data science jobs.
   * Limitations: Requires a large, high-quality dataset for training.
4. **Deep Learning Models (e.g., BERT):**
   * How it works: These models use neural networks to understand the context and meaning of text. They can analyze the entire resume and job description to make more accurate predictions.
   * Example: BERT can understand that “Python programming” and “coding in Python” mean the same thing, even if the wording is different.
   * Limitations: Requires a lot of computational power and expertise to implement.

**2.3 Gaps in Existing Solutions**

While there has been a lot of progress in automated resume screening, there are still some gaps and limitations in the existing solutions. Here’s what we’ve identified:

1. **Lack of Context Understanding:**Many existing systems rely on simple keyword matching or TF-IDF, which don’t understand the context or meaning of the text. This can lead to inaccurate matches, especially when the same word is used in different contexts.
2. **Bias in Training Data:**Machine learning and deep learning models are only as good as the data they’re trained on. If the training data is biased (e.g., favoring certain demographics or educational backgrounds), the system will also be biased.
3. **Scalability Issues:**Some advanced models, like deep learning, require a lot of computational power and may not be scalable for large organizations with thousands of resumes.
4. **Limited Customization:**Commercial tools often come with pre-built models that may not be customizable for specific industries or job roles. This can limit their effectiveness for certain use cases.
5. **Handling Non-Standard Resumes:**Many systems struggle with resumes that have non-standard formatting, such as images, tables, or creative designs. This can lead to missed information or inaccurate rankings.

**2.4 How Our Project Addresses These Gaps**

Our AI-powered Resume Screening and Ranking System aims to address these gaps by combining the strengths of existing approaches while introducing new innovations. Here’s how:

1. Advanced NLP Techniques:  
   We’ll use Natural Language Processing (NLP) techniques to understand the context and meaning of the text in resumes and job descriptions. This will help us avoid the limitations of simple keyword matching and TF-IDF.
2. Bias Reduction:  
   We’ll carefully curate our training data to minimize bias and ensure that the system evaluates resumes based on skills and qualifications, rather than subjective factors.
3. Scalability:  
   Our system will be designed to handle large volumes of resumes efficiently, making it suitable for organizations of all sizes.
4. Customization:  
   We’ll build a flexible system that can be customized for different industries and job roles, ensuring that it meets the specific needs of each organization.
5. Handling Non-Standard Resumes:  
   While our system will primarily focus on text-based resumes, we’ll explore ways to handle non-standard formats, such as extracting text from images or PDFs.

**CHAPTER 3**

**Proposed Methodology**

* 1. **System Design**

A diagram of a diagram of a company

AI-generated content may be incorrect.Our **AI-powered Resume Screening and Ranking System** follows a structured design to ensure an efficient and scalable solution for resume ranking. The system consists of multiple components, including **text extraction, data processing, feature engineering, similarity computation, and user interaction**.

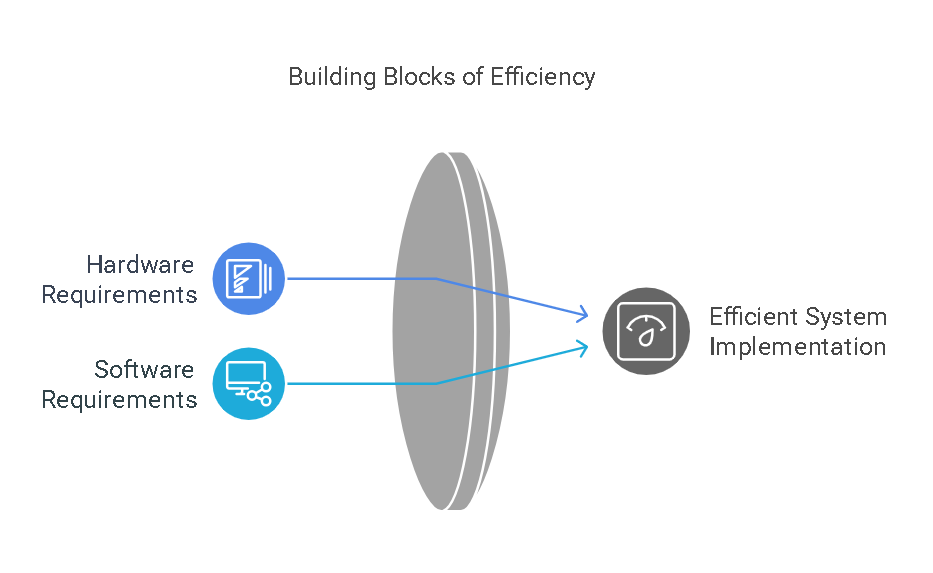
**System Architecture Diagram (Fig: 2 )**

**Components of the System:**

1. **User Input Layer:** The recruiter inputs the job description and uploads multiple resumes (PDF format).
2. **Text Extraction Module:** Extracts text content from PDF resumes using PyPDF2.
3. **Preprocessing & Feature Extraction**: Cleans extracted text, removes stopwords, and converts it into numerical representations using TF-IDF vectorization.
4. **Similarity Computation:** Measures relevance using cosine similarity between the job description and resumes.
5. **Ranking Module**: Assigns similarity scores and ranks resumes based on their match percentage.
6. **User Interface & Results:** Displays rankings, allows recruiters to filter results, preview resume text, and download rankings in CSV format.

**3.2 Requirement Specification**

To implement the system efficiently, we define both hardware and software requirements.



**(Figure :3)**

**Hardware Requirements:**

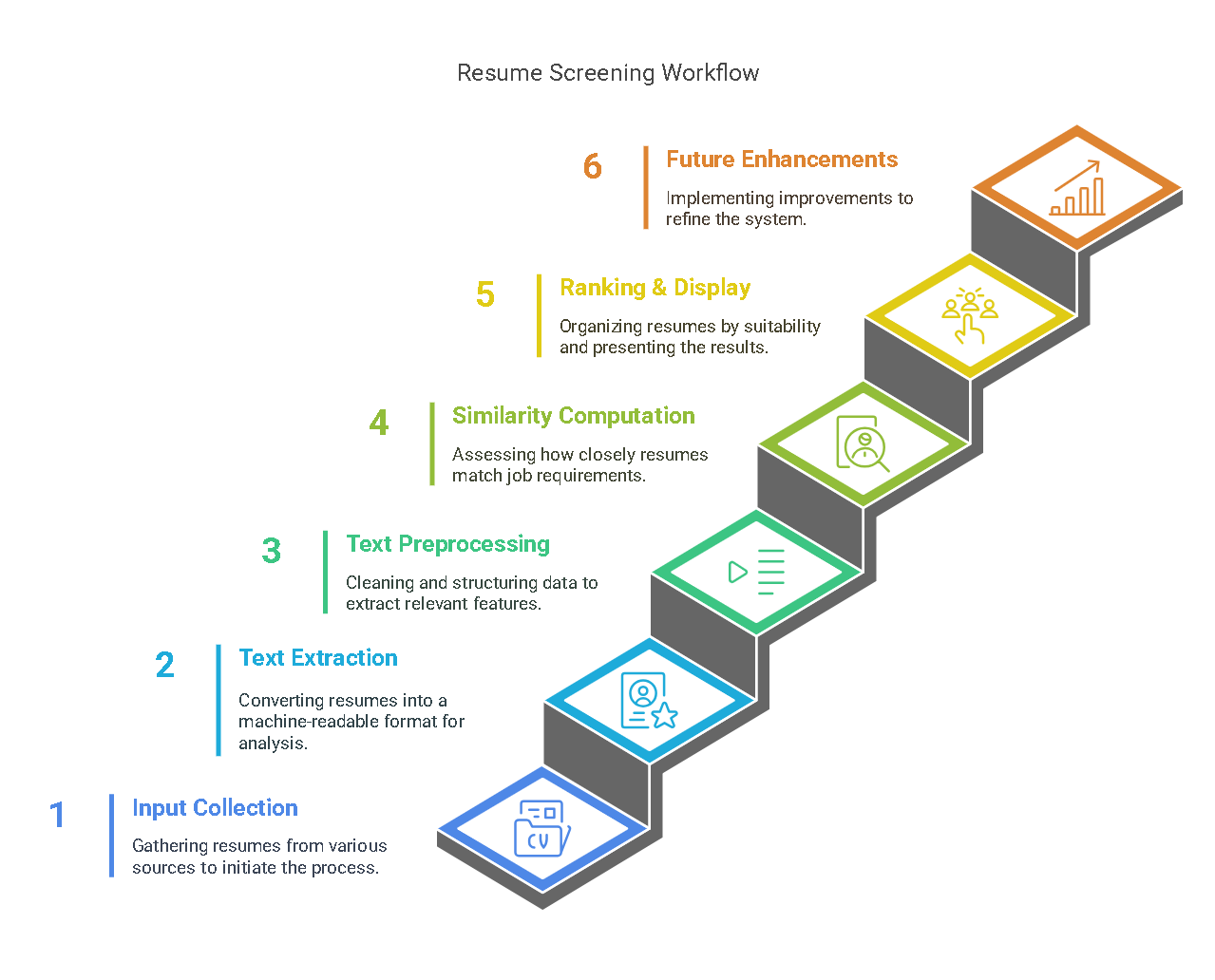
* A standard personal computer (PC) or laptop.
* Minimum 4GB RAM (Recommended: 8GB RAM for better performance).
* Processor: Intel Core i3 (or equivalent) and above.
* Storage: At least 500MB free space.

**Software Requirements:**

* Operating System: Windows, macOS, or Linux.
* Programming Language: Python.
* Libraries Used:
* Streamlit – For creating the web application.
* PyPDF2 – For extracting text from PDF resumes.
* Scikit-learn – For implementing TF-IDF and cosine similarity.
* Pandas & NumPy – For handling and processing data.
* Matplotlib & Seaborn – For visualizing similarity scores.

**3.3 Workflow**

The resume screening and ranking process follows a sequential workflow as described below figure 3:



**Step 1: Input Collection**

* The recruiter enters a job description in a text box.
* Multiple PDF resumes are uploaded using the file uploader in the Streamlit UI.

**Step 2: Text Extraction**

* The system extracts text from uploaded PDF files using PyPDF2.
* If a resume contains images instead of text, Optical Character Recognition (OCR) can be implemented in future versions.

**Step 3: Text Preprocessing & Feature Engineering**

* Data Cleaning: Converts all text to lowercase, removes special characters, and eliminates stop words.
* Tokenization: Splits the text into individual words for processing.
* TF-IDF Vectorization: Converts textual data into numerical feature vectors that represent important terms within resumes and the job description.

**Step 4: Resume Similarity Computation**

* The system compares each resume’s feature vector with the job description vector using cosine similarity.
* A similarity score between 0 (no match) and 1 (exact match) is assigned to each resume.

**Step 5: Resume Ranking & Displaying Results**

* Resumes are sorted in descending order based on similarity scores.
* The system displays a ranking table with candidate names and their similarity scores.
* Additional features include:
* Resume Preview: View extracted text from any resume.
* Score Threshold Filter: Adjust a slider to shortlist candidates based on a minimum similarity score.
* CSV Download: Download the ranked list for further analysis.

**Step 6: Enhancements and Future Improvements**

* To improve accuracy and usability, the following enhancements can be implemented:
* Support for Word Documents (DOCX) – Enable text extraction from multiple file formats.
* Advanced NLP Techniques – Implement deep learning models like BERT to improve contextual understanding of resumes.
* Machine Learning-Based Classification – Use supervised learning to classify resumes into different categories like “Highly Suitable,” “Moderately Suitable,” and “Not Suitable.”
* Integration with Applicant Tracking Systems (ATS) – Connect the system with existing HR software for seamless recruitment.

**CHAPTER 4**

**Implementation and Result**

* 1. **Snap Shots of Result:**

The image appears to be a screenshot or representation of a user interface for an **AI Resume Screening & Candidate Ranking System**.

**A screenshot of a computer screen

AI-generated content may be incorrect.**

A screenshot of a computer

AI-generated content may be incorrect.

**1. Job Description Section:**

* The section titled "Job Description" allows users to input the job description for which they want to screen resumes.
* This suggests that the system is designed to match resumes with job descriptions, particularly for technical roles like web development.

**2. Upload Resumes Section:**

* The section titled "Upload Resumes" allows users to upload PDF files containing resumes.
* The interface supports drag-and-drop functionality for uploading files.
* There is a note indicating a file size limit of 200MB per file, and only PDF files are accepted.

**3. Purpose:**

* The system is likely designed to:
* Extract text from uploaded resumes.
* Compare the resumes with the provided job description using techniques like TF-IDF and cosine similarity (as seen in the code snippet from the previous explanation).
* Rank the resumes based on their relevance to the job description.

**4.User Interaction:**

Users can:

* Enter a job description.
* Upload multiple resumes in PDF format.
* View the ranking of resumes based on their similarity to the job description.

**5. Ranking Resumes Section:**

* This section displays the ranking of resumes based on their similarity to the provided job description.
* The table shows:
  + **Resume**: The name of the uploaded resume file.
  + **Score**: A similarity score (between 0 and 1) indicating how well the resume matches the job description.

1. **Filter Candidates by Minimum Similarity Score:**

* A slider or input field allows users to filter candidates based on a minimum similarity score.
* The slider ranges from **0.0 to 1.0**, and the current value is set to **0.0** (no filtering applied).
* The values **0, 0.0** and **1, 0.0** might represent the slider's range or steps.

**7. Filtered Candidates Section:**

* After applying the filter (if any), this section displays the resumes that meet the minimum similarity score.
* Since the filter is set to **0.0**, all resumes are shown.

**8. Download Results as CSV:**

* A button or option is provided to download the ranking results as a **CSV file**.
* This allows users to save the ranked list of resumes for further analysis or sharing.

**9. Show Extracted Resume Texts:**

* An option is available to view the **extracted text** from each resume.
* This feature allows users to verify the content of the resumes and understand why certain scores were assigned.

**10. Download Extracted Texts:**

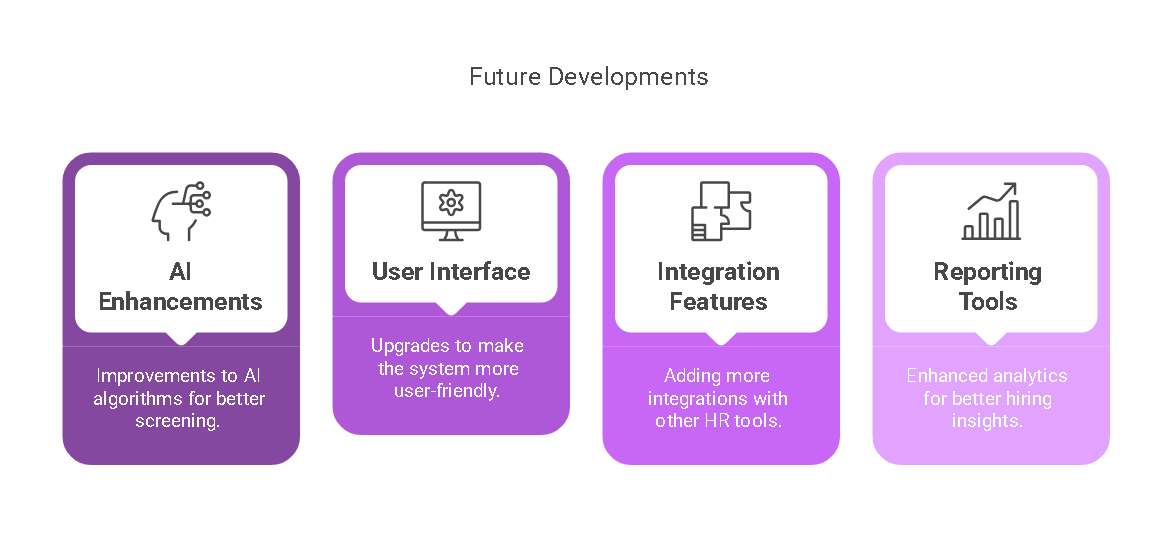
* Users can download the extracted text from all resumes as a file.
* This is useful for offline review or further processing.
  1. **GitHub Link for Code:**

[**https://github.com/143Nikshit/Resume-Ranking-System.git**](https://github.com/143Nikshit/Resume-Ranking-System.git)

**CHAPTER 5**

**Discussion and Conclusion**

* 1. **Future Work:**

While our AI-powered Resume Screening and Ranking System has greatly improved the hiring process, there is always room for enhancement. Here are some future developments that could make the system even better:

**(Figure : 5)**

* **Supporting More Resume Formats:** Right now, the system works best with text-based PDFs. A future update could allow Word documents (DOCX), plain text (TXT), and even scanned resumes using Optical Character Recognition (OCR) so that more job seekers can be included.
* **Better Understanding of Resume Content:** By using Named Entity Recognition (NER), the system can intelligently detect skills, work experience, and education details from resumes to improve matching accuracy.
* **Smarter AI with Deep Learning:** Instead of just comparing words, using deep learning models like BERT or GPT-based algorithms can help the system truly understand resumes in a human-like way.
* **Integration with HR Systems:** Many companies use Applicant Tracking Systems (ATS) to manage job applications. Connecting our system with these platforms would allow seamless recruitment and faster decision-making.
* **Adding Interviews and Assessments:** The system could be extended to include online tests, psychometric evaluations, and video interviews to give recruiters a more complete view of candidates.

**5.2 Conclusion:**

The **AI-powered Resume Screening and Ranking System** has transformed the way recruiters handle job applications. Instead of spending hours manually reviewing resumes, HR teams can now **quickly and accurately identify top candidates** with the help of AI.

By leveraging **TF-IDF vectorization and cosine similarity,** our system ensures that recruiters get the **most relevant resumes ranked** for any given job description. Features like **resume preview, score-based filtering, and CSV report downloads** make the system highly practical and easy to use.

While this project is already helping automate recruitment, **future improvements** like **deep learning-based classification, better resume parsing, and ATS integration** will make it even more **powerful and adaptable**. With ongoing advancements in AI, **hiring processes will become more fair, efficient, and data-driven.**

In summary, this project showcases how AI can **revolutionize hiring**, making it **faster, smarter, and more effective.** As technology continues to evolve, AI-driven recruitment will soon become a standard in HR, benefiting both companies and job seekers alike.

**REFERENCES**

1. **Scikit-learn Documentation**, *"TF-IDF Vectorizer in Python"*,
2. **Google AI Blog**, *"Understanding Natural Language Processing in Resume Screening"*
3. **Indeed Blog**, *"How AI is Transforming Resume Screening"*, Available at.
4. **Streamlit Blog**, *"How to Build an AI-Powered Resume Screening App"*
5. **Streamlit Community Forum**, *"Using Streamlit for AI and Machine Learning Applications"*