**Resume Screening App**

**A Main Project submitted**

**in partial fulfillment of the requirements**

**for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**submitted by**

**1. M. Satya Kalyani (15PA1A0533) 2. Y. Mary Sujatha (16PA5A0506)**

**3. V. Swathi (15PA1A0558) 4. K. Kiran Kumar (15PA1A0532)**

**Under the esteemed guidance**

**of CH. Lakshmi**

**Veenadhari Assistant**

**Professor**

****

**DEPARTMENT OF COMPUTER SCIENCE AND**

**ENGINEERING VISHNU INSTITUTE OF TECHNOLOGY**

**(Autonomous)**

**(Approved by AICTE, Accredited by NBA & NAAC and permanently affiliated to JNTU Kakinada) BHIMAVARAM – 534 202**

**VISHNU INSTITUTE OF TECHNOLOGY**

**(Autonomous)**

**(Approved by AICTE, accredited by NBA & NAAC, and permanently affiliated to JNTU Kakinada)**

**BHIMAVARAM-534202**

**2023-2024**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

**This is to certify that the project entitled “EVENT MANAGEMENT SYSTEM”, is being submitted by M.SATYA KALYANI,Y.MARY SUJATHA,V.SWATHI AND K.KIRAN KUMAR, bearing the REGD.NOS: 15PA1A0533, 16PA5A0506, 15PA1A0558 and 15PA1A0532 submitted in fulfillment for the award of the degree of “BACHELOR OF TECHNOLOGY” in “COMPUTER SCIENCE AND ENGINEERING” is a record of work carried out by them under my guidance and supervision during the academic year 2023-2024 and it has been found worthy of acceptance according to the requirements of university**

**Internal Guide Head of the Department**

CH. Lakshmi VeenadhariDr. Sumit Gupta

External Examiner

**ACKNOWLEDGEMENT**

It is natural and inevitable that the thoughts and ideas of other people tend to drift into the subconscious due to various human parameters, where one feels the need to acknowledge the help and guidance derived from others. We express our gratitude to those who have contributed to the fulfillment of this project.

We take the opportunity to extend our sincere thanks to **Dr. D. Suryanarayana**, the director of VIT, Bhimavaram, whose guidance from time to time helped us complete this project successfully.

We also express our sincere gratitude to **Dr. M. Venu**, the principal of VIT, Bhimavaram, for his continuous support in helping us complete the project on time.

We are thankful to **Dr. Sumit Gupta**, Head of the Department of Computer Science and Engineering, for his continuous and unwavering support and guidance. We acknowledge our gratitude for his valuable guidance and support extended to us from the conception of the idea to the completion of this project.

We are very thankful to **CH. Lakshmi Veenadhari**, Assistant Professor, our internal guide, whose guidance from time to time helped us complete this project successfully.

**Project Associates**

**CONTENTS**

**ABSTRACT**

**CHAPTERS**

1. **INTRODUCTION**

**1.2 PROJECT INTRODUCTION**

**1.2 MOTIVATION**

**1.3 OBJECTIVE**

**1.4 SCOPE**

1. **SYSTEM ANALYSIS**

**2.1 SOFTWARE REQUIREMENTS**

**2.2 HARDWARE REQUIREMENTS**

**2.3 PYTHON ENVIRONMENT**

**2.4 EXISTING SYSTEM**

**2.5 PROPOSED SYSTEM**

1. **SYSTEM DESIGN**

**3.1 DATA FLOW DIAGRAM**

**3.2 USE CASE DIAGRAM**

**3.3 CLASS DIAGRAM**

**3.4 SEQUENCE DIAGRAM**

1. **MODULE DESCRIPTIONS**

**4.1 TEXT EMBEDDING MODULE**

**4.2 COSINE SIMILARITY MODULE**

**4.3 GENERATIVE AI INTEGRATION MODULE**

**4.4 USER INTERFACE MODULE**

**4.5 PDF PARSING MODULE**

1. **IMPLEMENTATION**

**5.1 TECHNOLOGIES USED**

**5.2 SAMPLE CODE**

**5.3 SCREENSHOTS OF WEBPAGES**

1. **TESTING**

**6.1 TESTING STRATEGIES USED**

**6.2 TEST CASE REPORTS**

1. **CONCLUSION & FUTURE WORK**

**6.1 CONCLUSION**

**6.2 FUTURE WORK**

1. **REFERENCES**

**Abstract**

A Resume Screening Application is a software tool designed to automate and streamline the process of evaluating candidate resumes against job descriptions during recruitment processes. It utilizes advanced technologies to analyze textual data from resumes and job descriptions, allowing recruiters to efficiently identify qualified candidates based on their skills, experiences, and qualifications. By automating the initial screening process, these applications save time and resources for hiring teams while ensuring a more objective and consistent evaluation of candidates.

Our Resume Screening Application harnesses the power of cutting-edge technologies to optimize the candidate evaluation process. Leveraging models such as BERT, Doc2Vec, and the Gemini model, the application extracts and analyzes textual data from resumes and job descriptions, generating embeddings that represent the semantic meanings of the text. These embeddings are then compared using cosine similarity calculations to quantify the alignment between candidate qualifications and job requirements. With a user-friendly interface built on Streamlit, recruiters can easily upload resumes, input job descriptions, and view real-time results. Additionally, the application is designed for scalability, allowing it to efficiently process large volumes of data while maintaining optimal performance. By automating and enhancing the screening process, our application empowers recruiters to make informed decisions and select the best candidates for their organizations.

**Chapter 1**

**Introduction**

**1.1 Project Introduction**

In today's competitive job market, the task of identifying the most suitable candidates for job roles can be daunting for recruiters and hiring managers. The Resume Screening Application emerges as a powerful solution to streamline and optimize this critical aspect of the recruitment process. By leveraging advanced technologies in natural language processing (NLP) and machine learning, this application revolutionizes the way resumes are evaluated against job descriptions. With its intelligent algorithms and user-friendly interface, the Resume Screening Application offers a seamless experience for recruiters, allowing them to efficiently sift through a large pool of candidates and identify the most qualified individuals. In this document, we delve into the core functionalities, underlying technologies, and operational mechanisms of the Resume Screening Application, showcasing how it transforms the recruitment landscape and empowers organizations to make informed hiring decisions.

**1.3 Objective**

The primary objective of the Resume Screening App is to revolutionize the candidate evaluation process in recruitment and hiring by leveraging advanced technologies to automate and optimize resume screening. The app aims to achieve the following objectives:

**1. Efficiency :** Streamline the resume screening process to save time and resources for HR professionals and hiring managers. By automating repetitive tasks and employing advanced algorithms, the app aims to expedite the screening process and enable faster decision-making.

**2. Accuracy:** Enhance the accuracy and reliability of candidate evaluations by leveraging machine learning and natural language processing techniques. The app aims to accurately match candidate qualifications with job requirements, ensuring that only the most suitable candidates are shortlisted for further consideration.

**3. Objectivity:** Reduce the influence of unconscious biases in the screening process by focusing on objective criteria and data-driven insights. By standardizing the screening process and minimizing human intervention, the app promotes fairness and equality in candidate evaluations.

**4. Scalability:** Enable organizations to handle large volumes of resumes efficiently and effectively, regardless of the size of the applicant pool. The app aims to scale seamlessly to accommodate varying levels of demand, ensuring consistent and reliable performance even during peak hiring periods.

**5. User-Friendly Experience:** Provide a user-friendly interface that is intuitive and easy to navigate for both recruiters and job seekers. The app aims to simplify the resume submission and evaluation process, making it accessible to users with varying levels of technical expertise.

**6. Insights and Analytics:** Generate valuable insights and analytics about candidate qualifications, skills, and job fit to inform hiring decisions. The app aims to provide recruiters and hiring managers with actionable data and trends, empowering them to make informed decisions and drive organizational success.

**7. Compliance:** Ensure compliance with legal and regulatory requirements related to recruitment and hiring practices. The app aims to adhere to industry standards and best practices, helping organizations mitigate risks and avoid potential legal liabilities.

Overall, the objective of the Resume Screening App is to modernize and optimize the candidate evaluation process, ultimately facilitating more efficient, fair, and data-driven hiring decisions.

**1.4 Scope**

The scope of the Resume Screening Application encompasses various aspects of the candidate evaluation process in recruitment. Key components and functionalities within the scope of the application include:

1. **Resume Parsing:** The application parses and extracts relevant information from candidate resumes, including skills, experiences, education, and certifications.
2. **Job Description Analysis**: It analyzes job descriptions to identify key qualifications, skills, and requirements for specific job roles.
3. **Candidate Matching**: Using advanced natural language processing (NLP) techniques and machine learning models, the application matches candidate profiles with job descriptions to determine suitability and alignment.
4. **Cosine Similarity Calculation**: The application calculates cosine similarity scores to quantify the degree of match between candidate qualifications and job requirements.
5. **User Interface**: It provides an intuitive and user-friendly interface for recruiters to upload resumes, input job descriptions, and view screening results.
6. **Performance Optimization**: The application is designed for scalability and performance optimization, capable of handling large volumes of resumes and job descriptions efficiently.
7. **Automation**: It automates repetitive tasks in the candidate screening process, allowing recruiters to streamline workflow processes and focus on more strategic aspects of recruitment.
8. **Integration**: The application can be integrated with existing recruitment systems and workflows, enhancing overall efficiency and productivity.

**Chapter 2**

**System Analysis**

Requirement analysis for the Resume Screening App encompasses software, hardware, and Python environment requirements to ensure the successful development, deployment, and operation of the application.

**2.1. Software Requirements**

**Python**: The application is developed using Python programming language, requiring a compatible Python interpreter installed on the user's device for execution.

1. **Streamlit**: The user interface of the application is built using the Streamlit library, necessitating the installation of Streamlit framework for running the application.
2. **Transformers Library**: The app utilizes the Transformers library from Hugging Face for natural language processing tasks, requiring installation and integration of the library for generating contextual embeddings.
3. **Gensim Library**: Additionally, the Gensim library is used for Doc2Vec embedding generation, necessitating installation and integration for document similarity calculations.
4. **PyPDF2**: For parsing resume PDF files, the PyPDF2 library is employed, requiring installation to extract text data from PDF documents.
5. **NLTK**: The NLTK library is utilized for tokenization and text processing tasks, requiring installation to support various natural language processing functionalities.
6. **PyTorch**: The application utilizes PyTorch for deep learning functionalities, particularly in utilizing BERT model, requiring installation to support model execution.
7. **Google Generative AI** (Gemini Model): Optionally, the application can utilize the Google Generative AI model for text generation tasks, requiring installation and integration for enhanced functionality.

**2.2. Hardware Requirements**

**- Development Environment:** A computer or workstation with sufficient processing power, memory, and storage capacity to support software development activities, including coding, testing, and debugging.

**- Deployment Environment**: A server or cloud-based infrastructure capable of hosting the Streamlit application, with adequate resources to handle user requests, data processing, and storage requirements.

**- Scalability:** The deployment environment should be scalable to accommodate potential increases in user traffic and data volume, ensuring optimal performance and responsiveness of the application.

**2.3 Python Environment**

**- Python Version**: The application is compatible with Python 3.x versions, with Python 3.7 or higher recommended for optimal performance and compatibility with the required libraries and frameworks.

**- Virtual Environment**: It is recommended to set up a virtual environment for the application to manage dependencies and ensure isolation from other Python projects or system-level packages.

**- Package Management:** A package manager such as pip or conda is used to install and manage Python packages and dependencies required by the application, ensuring consistent and reproducible environment configurations.

By identifying and documenting the software, hardware, and Python environment requirements, the development team can ensure that all necessary resources and dependencies are available to support the development, deployment, and operation of the Resume Screening App. This comprehensive requirement analysis lays the foundation for successful project execution and delivery.

**2.4 Existing system**

In the domain of candidate evaluation and recruitment, existing systems encompass a variety of tools and platforms designed to streamline and optimize the hiring process. These systems typically offer functionalities such as resume parsing, candidate matching, interview scheduling, and performance analytics. One common type of existing system is the Applicant Tracking System (ATS), which serves as a centralized platform for managing job postings, receiving applications, and tracking candidate progress throughout the recruitment pipeline.

ATS platforms typically feature resume parsing capabilities, allowing recruiters to extract key information from resumes and store it in a structured format for easy retrieval and analysis. Additionally, these systems often include keyword-based searching and filtering functionalities, enabling recruiters to quickly identify candidates with the desired skills and qualifications.

Another key component of existing systems is candidate matching and ranking. These systems leverage algorithms and machine learning techniques to analyze resumes and job descriptions, identifying the most suitable candidates for specific roles based on factors such as skills, experience, and education. Candidate ranking algorithms prioritize candidates based on their relevance to the job requirements, facilitating more efficient screening and selection processes.

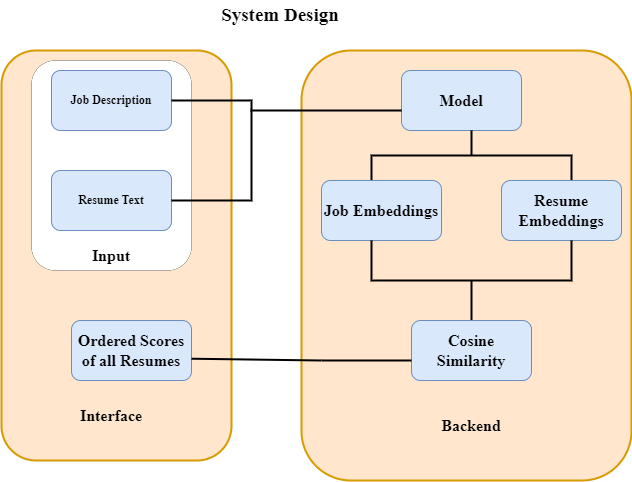
Interview scheduling and coordination are also important features of existing systems, allowing recruiters to seamlessly schedule interviews, communicate with candidates, and manage interview logistics. Some systems offer automated scheduling capabilities, using AI-driven algorithms to identify optimal interview times based on the availability of both recruiters and candidates.

Performance analytics and reporting functionalities are often integrated into existing systems to provide recruiters and hiring managers with insights into recruitment metrics such as time-to-fill, source of hire, and candidate conversion rates. These analytics help organizations track the effectiveness of their recruitment efforts and identify areas for improvement.

Overall, existing systems play a crucial role in streamlining and optimizing the recruitment process, providing recruiters with the tools and insights they need to identify, evaluate, and hire top talent effectively. However, there is ongoing innovation in this space, with new technologies and approaches continuously emerging to further enhance the efficiency and effectiveness of candidate evaluation and recruitment.

**2.5 Proposed System**

The proposed system is a Smart Applicant Tracking System designed to streamline the process of screening resumes and matching them with job descriptions. It leverages advanced natural language processing (NLP) techniques and machine learning models to analyze the content of resumes and job descriptions, providing insights into the suitability of candidates for specific roles.



At its core, the system employs state-of-the-art NLP models such as Hugging Face's BERT and Gensim's Doc2Vec for generating semantic embeddings of text data. These embeddings capture the contextual meaning of words and phrases, enabling the system to understand the content of resumes and job descriptions in a more nuanced manner.

The system's functionality can be summarized as follows:

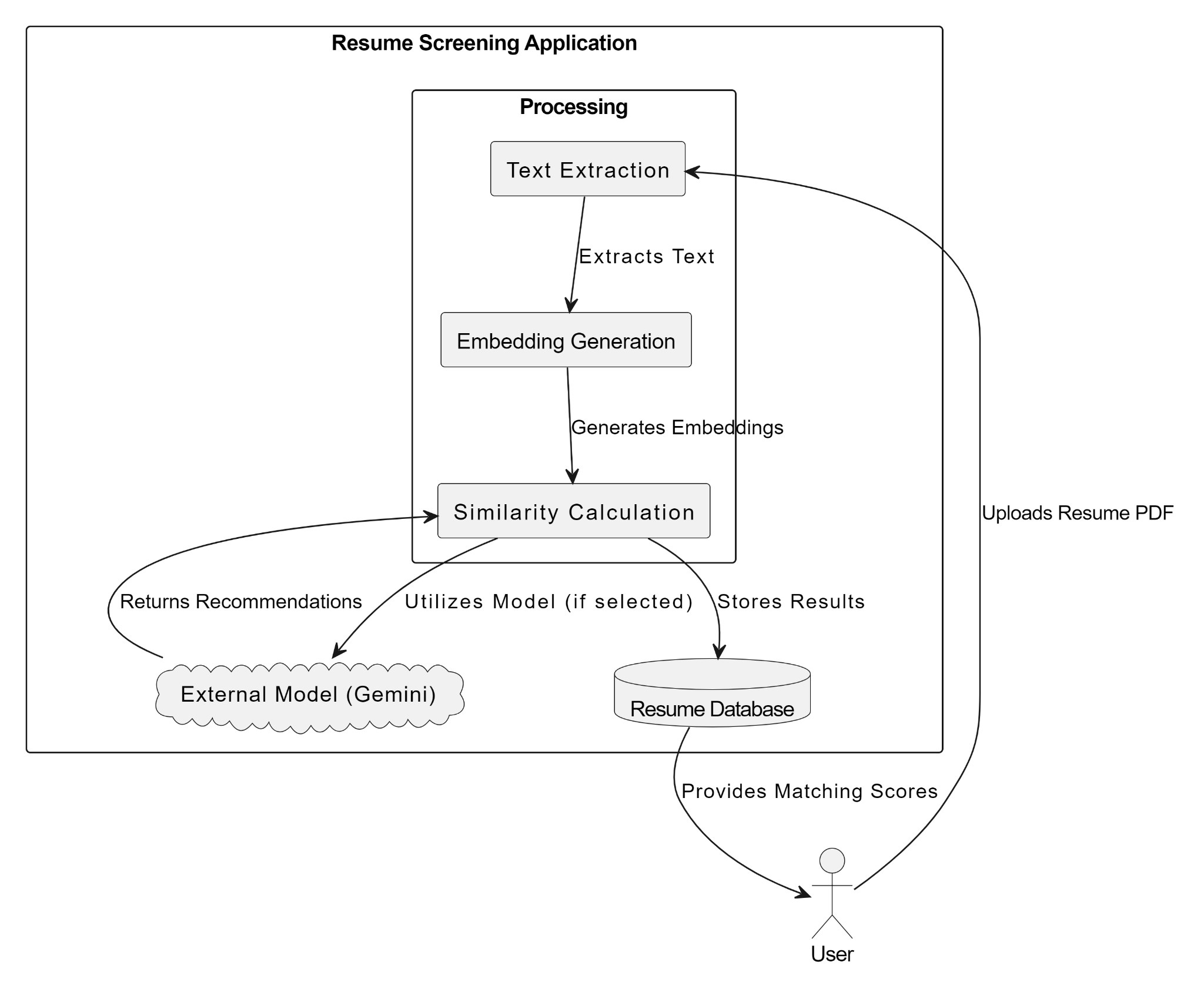
* **Resume Analysis:** Users can upload resumes into the system. The system then processes this textual data, extracting relevant information and generating embeddings to represent the semantic meaning of the text.
* **Semantic Matching:** The system computes the similarity between the embeddings of resumes and job descriptions using techniques like cosine similarity. This allows the system to quantitatively measure the degree of alignment between a candidate's qualifications and the requirements outlined in the job description.
* **Feedback and Recommendations:** Based on the similarity scores and other analysis, the system provides feedback to users, indicating the percentage match between a resume and a job description. Additionally, it may suggest areas for improvement in the resume or recommend relevant courses or resources to enhance the candidate's qualifications**.**
* **Integration with Generative AI Model:** Optionally, the system can integrate with a Generative AI model (Gemini Model) to provide more comprehensive feedback and recommendations. This model can generate responses based on predefined prompts, offering insights into missing keywords, profile summaries, and recommended courses/resources.
* **User-friendly Interface:** The system is implemented as a web application using Streamlit, providing users with an intuitive interface for uploading resumes, entering job descriptions, and initiating comparisons. Users can easily interact with the system and obtain actionable insights regarding candidate evaluation and selection.

Overall, the proposed system aims to automate and optimize the resume screening process, enabling recruiters to efficiently identify qualified candidates and make informed hiring decisions. By leveraging advanced NLP techniques and machine learning models, the system offers a powerful tool for talent acquisition and recruitment management.

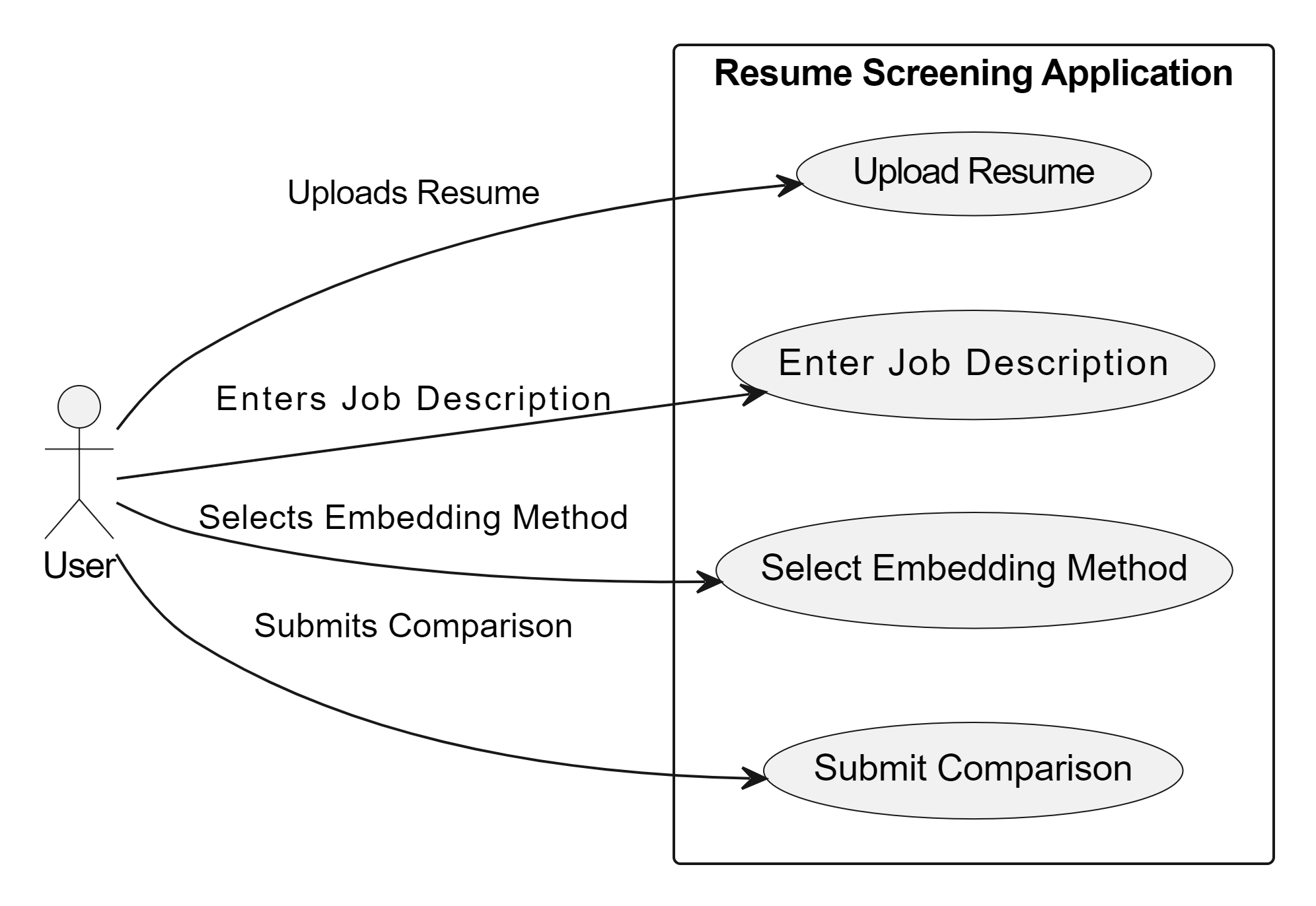
**Chapter 3**

**System Design**

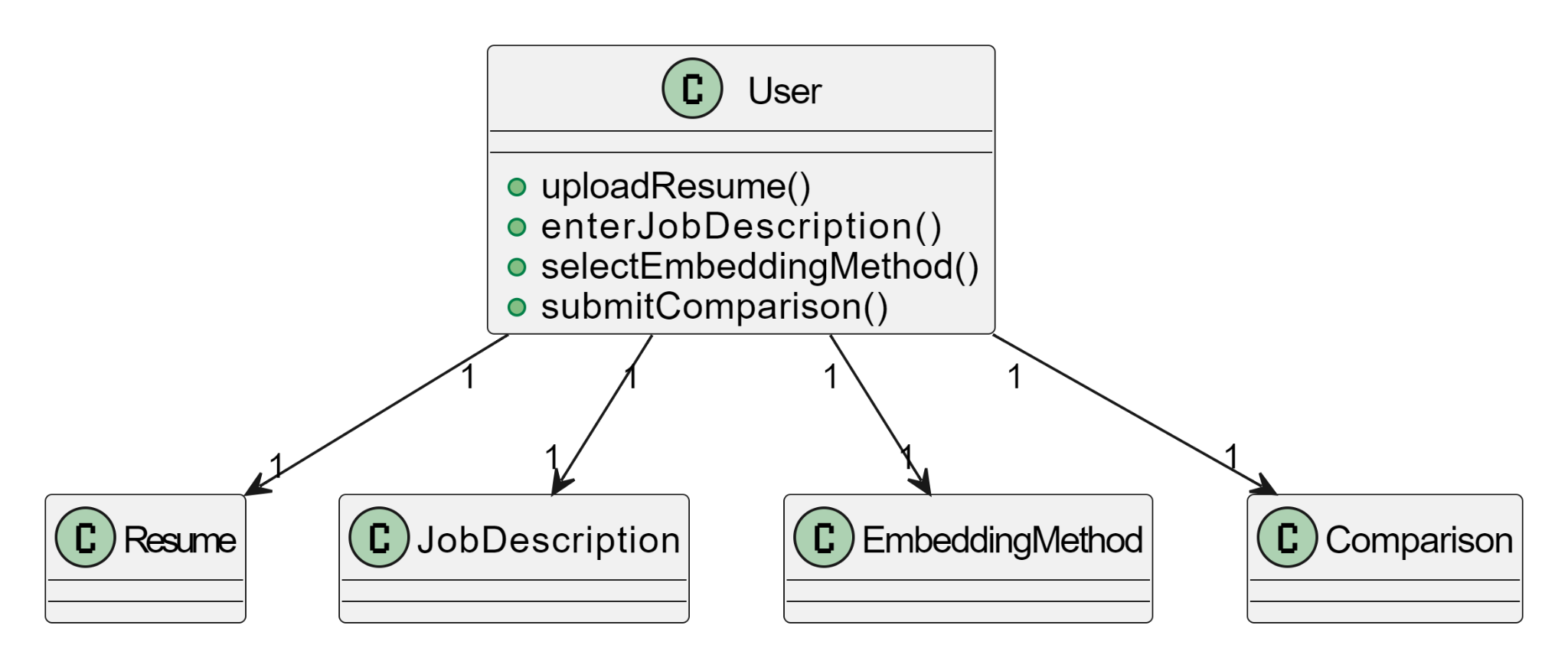
**3.1 Data flow diagram**

****

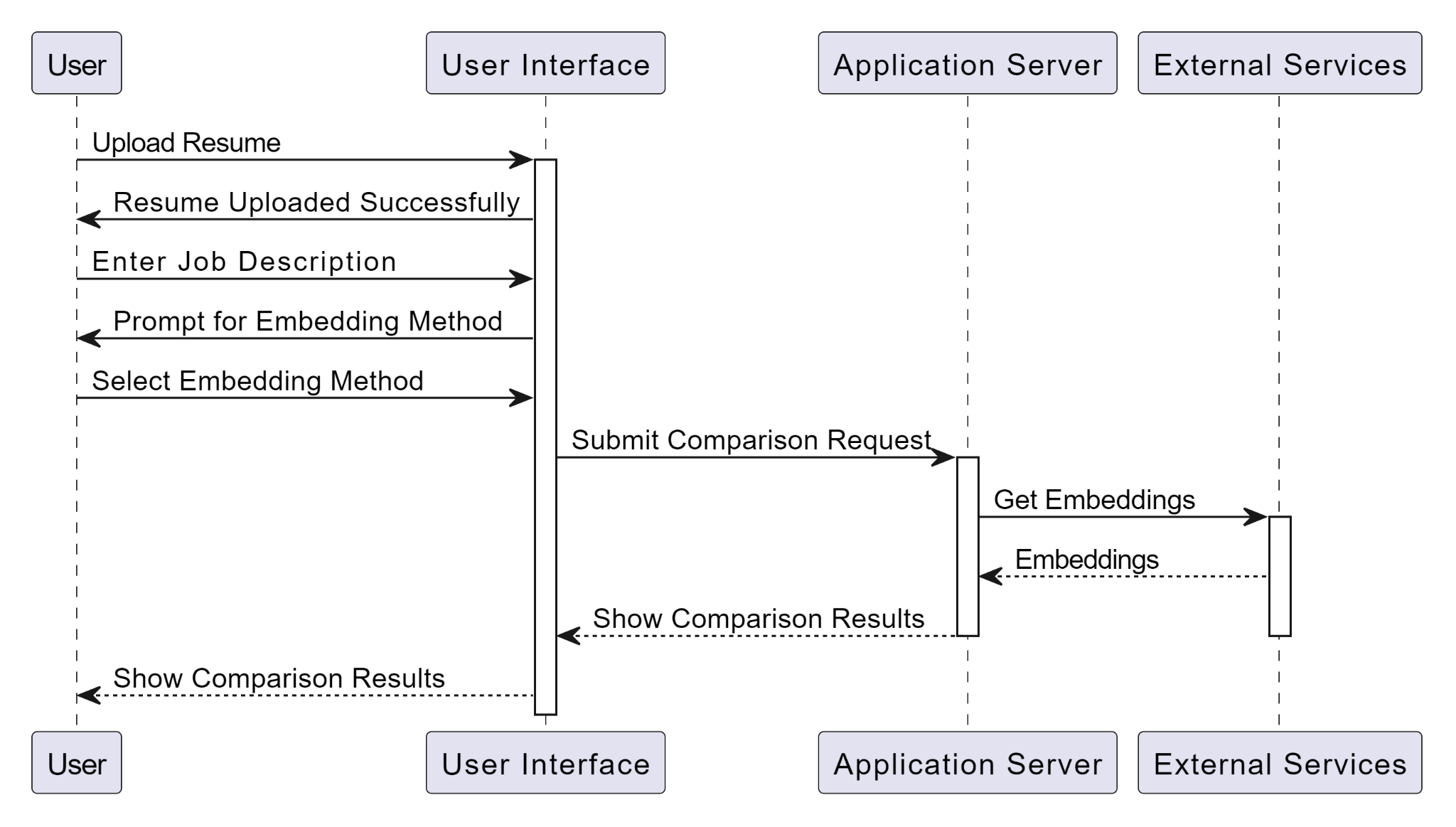
**3.2 Use case diagram**

****

**3.3 Class diagram**

****

**3.4 Sequence diagram**

****

**Chapter 4.**

**Module Descriptions**

**4.1 Text Embedding Module**:

This module is responsible for transforming textual data into numerical representations, facilitating analysis and comparison. It leverages state-of-the-art NLP models such as BERT and Doc2Vec to convert raw text into high-dimensional embeddings. BERT, a transformer-based model, captures contextualized embeddings that encode the semantic meaning of words and sentences. On the other hand, Doc2Vec generates fixed-size vectors representing entire documents, enabling efficient comparison of resumes and job descriptions. By utilizing these techniques, the Text Embedding Module ensures that textual information is translated into a format suitable for further analysis and evaluation within the application.

**4.2 Cosine Similarity Module:**

The Cosine Similarity Module computes the similarity between pairs of embeddings using the cosine similarity metric. This metric measures the cosine of the angle between two vectors, providing a measure of their alignment in a high-dimensional space. By comparing the embeddings of resumes and job descriptions, this module determines the degree of match between a candidate's qualifications and the requirements of a job. Higher cosine similarity scores indicate a stronger alignment, suggesting a better fit between the candidate and the position. This module plays a crucial role in the screening process by quantifying the similarity between resumes and job descriptions, aiding recruiters in identifying suitable candidates efficiently.

**4.3 Generative AI Integration Module:**

The integration of the Gemini model adds an advanced capability to the application, enabling it to generate content based on a given prompt. This model, powered by generative AI technology, produces responses in natural language, enhancing the application's ability to provide personalized feedback and recommendations to users. By incorporating the Gemini model, the application can offer tailored suggestions for improving resumes, recommending relevant courses or resources, and providing insights into missing keywords or profile summaries. This integration enriches the user experience and elevates the functionality of the application by leveraging cutting-edge AI technology for content generation and analysis.

**4.4 User Interface Module** (Streamlit):

The User Interface (UI) Module serves as the front-end component of the application, providing an intuitive and interactive interface for users to interact with the system. Developed using the Streamlit framework, the UI module offers a user-friendly environment where users can upload their resumes, input job descriptions, and initiate the comparison process. Through Streamlit, various UI elements such as file uploaders, text areas, and buttons are seamlessly integrated, facilitating smooth user interactions. Additionally, the module incorporates dynamic elements that update in real-time based on user inputs, enhancing the responsiveness and usability of the interface. Overall, the UI module plays a crucial role in facilitating user engagement and streamlining the resume screening process by presenting a visually appealing and functional interface for users to interact with.

**4.5 PDF Parsing Module**:

The PDF Parsing Module is responsible for extracting text data from uploaded resume files in PDF format. Leveraging the PDFPlumber library, this module reads the contents of PDF documents and converts them into plaintext format, making the textual information accessible for further processing. By parsing PDF files, the module ensures compatibility with common resume formats and enables seamless integration of resume data into the application. This functionality enhances the user experience by simplifying the process of uploading and analyzing resumes, allowing users to submit their documents with ease and efficiency.

**Chapter 5**

**Implementation**

**Technologies used**

The application leverages a combination of cutting-edge technologies to fulfill its functionalities effectively:

1. **Streamlit:** The UI is built using Streamlit, a Python library that simplifies the creation of web applications for data science and machine learning projects. Streamlit allows for the seamless integration of various UI elements and enables real-time updates based on user interactions, enhancing the overall user experience.
2. **Hugging Face Transformers Library:** The application utilizes the Transformers library from Hugging Face for natural language processing tasks. Specifically, it employs pre-trained models such as BERT for generating embeddings of job descriptions and resumes, enabling the computation of similarity scores between them.
3. **Gensim Library:** The Gensim library is employed for Doc2Vec embedding generation, which facilitates document similarity calculations. By utilizing Doc2Vec models, the application can capture semantic similarities between job descriptions and resumes, providing more nuanced insights during the screening process.
4. **PyPDF2 and PDFPlumber:** These libraries are used for parsing resume PDF files and extracting text data from them. PyPDF2 and PDFPlumber enable the application to process resume documents uploaded by users and extract relevant textual information for analysis.
5. **NLTK (Natural Language Toolkit):** NLTK is utilized for various text processing tasks, including tokenization and other preprocessing steps. By leveraging NLTK functionalities, the application can prepare text data for embedding generation and similarity calculations effectively.
6. **Generative AI (Gemini):** The application integrates with Generative AI models, specifically the Gemini model, to provide additional insights and recommendations for resume improvement. Through the Gemini model, the application can generate content and recommendations based on user inputs, enhancing the overall utility of the system.

**Sample code:**

The application's functionality revolves around three main components: the user interface, backend processing, and external services integration.

The user interface allows users to upload a resume PDF file and input a job description easily. Behind the scenes, the backend processing modules handle the core functionalities.

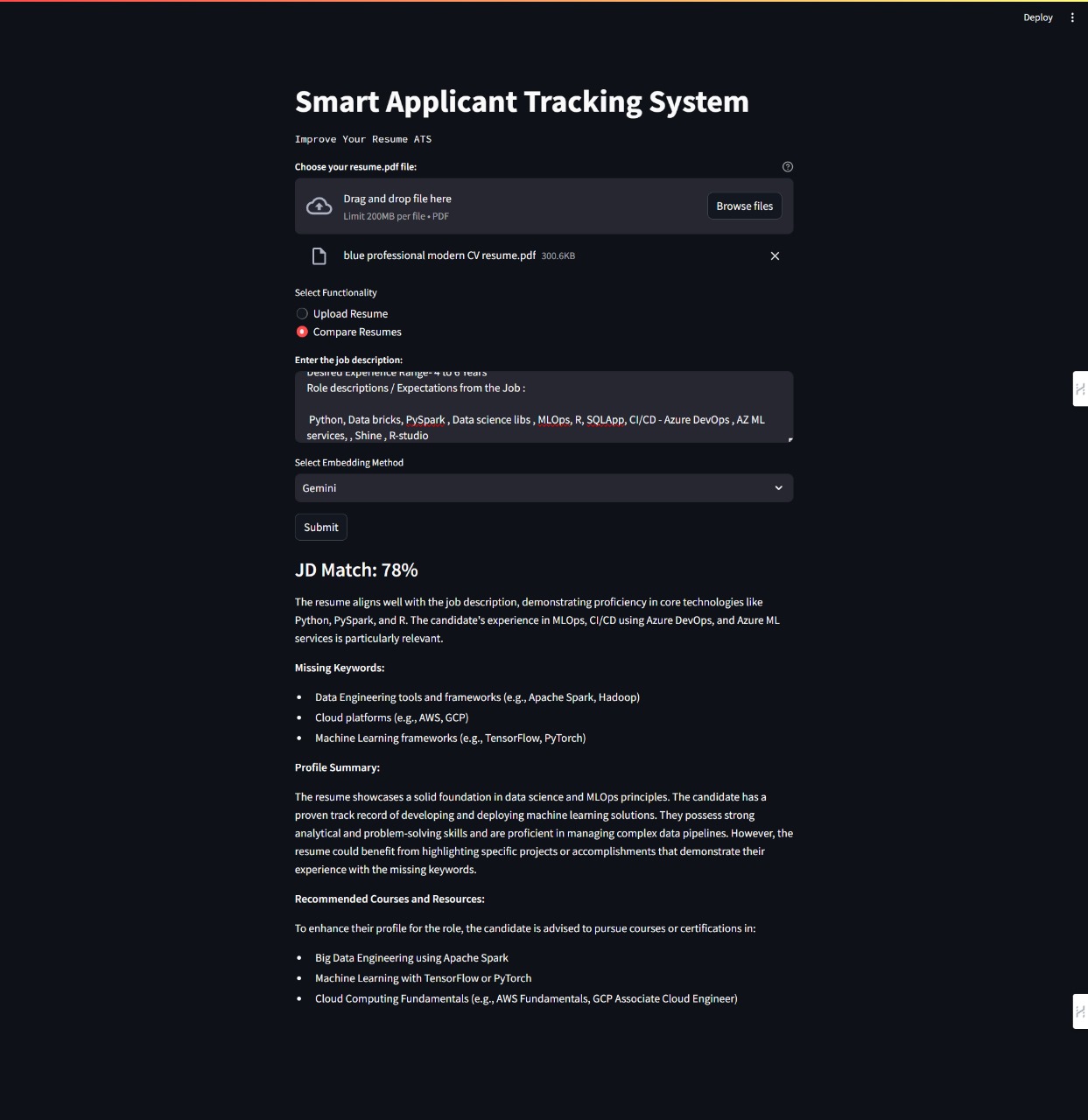
Using natural language processing techniques, such as BERT and Doc2Vec, the backend generates semantic representations (embeddings) for both the resume and job description.

Then, it computes similarity scores between these embeddings to quantify the alignment between the candidate's qualifications and the job requirements.

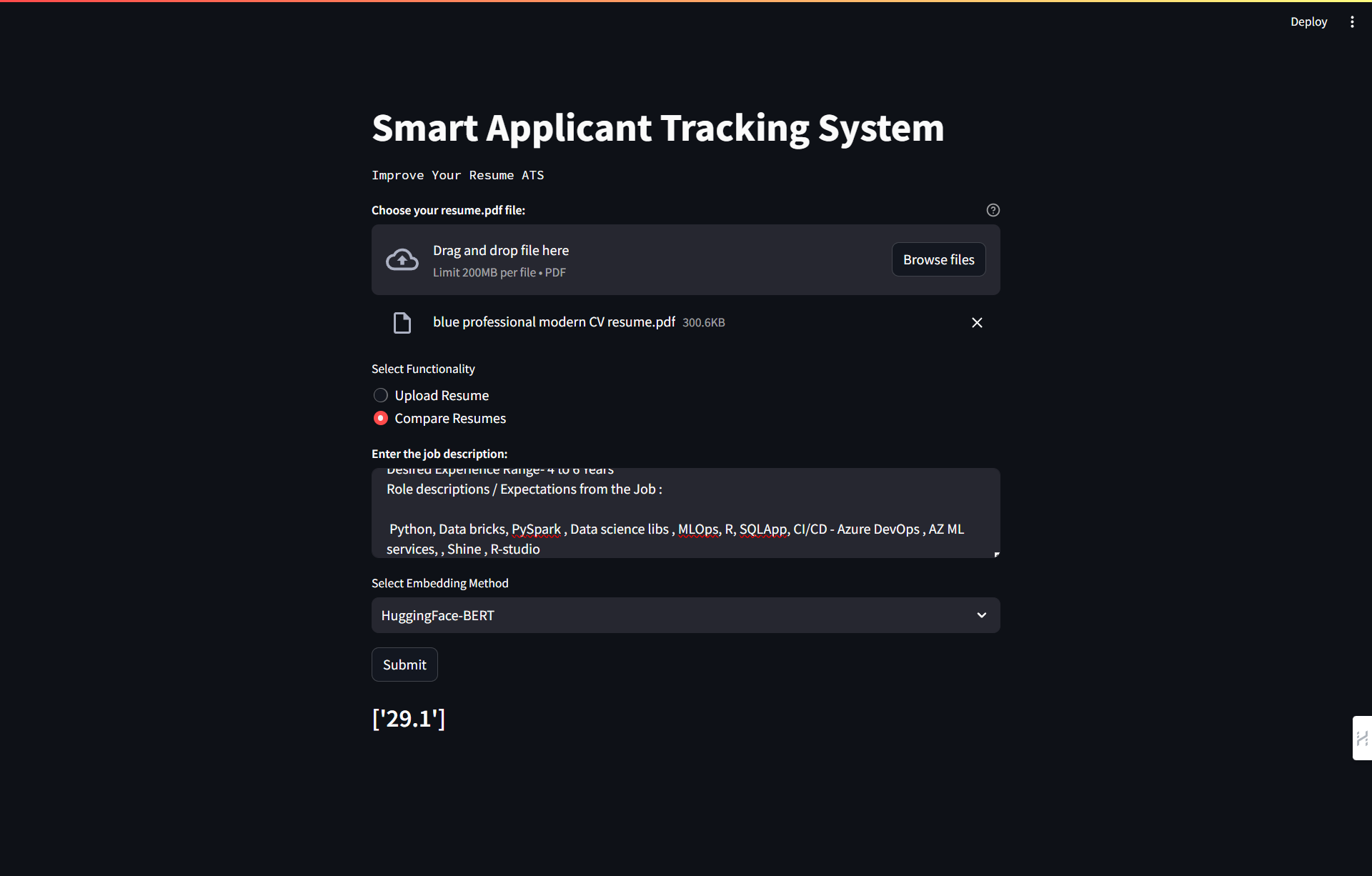
Integration with external services like Google's Generative AI model (Gemini) enhances the application's functionality by providing additional insights or suggestions for resume improvement.

Overall, the application combines a simple user interface with sophisticated backend processing to offer users a powerful tool for evaluating resume-job description alignment efficiently.

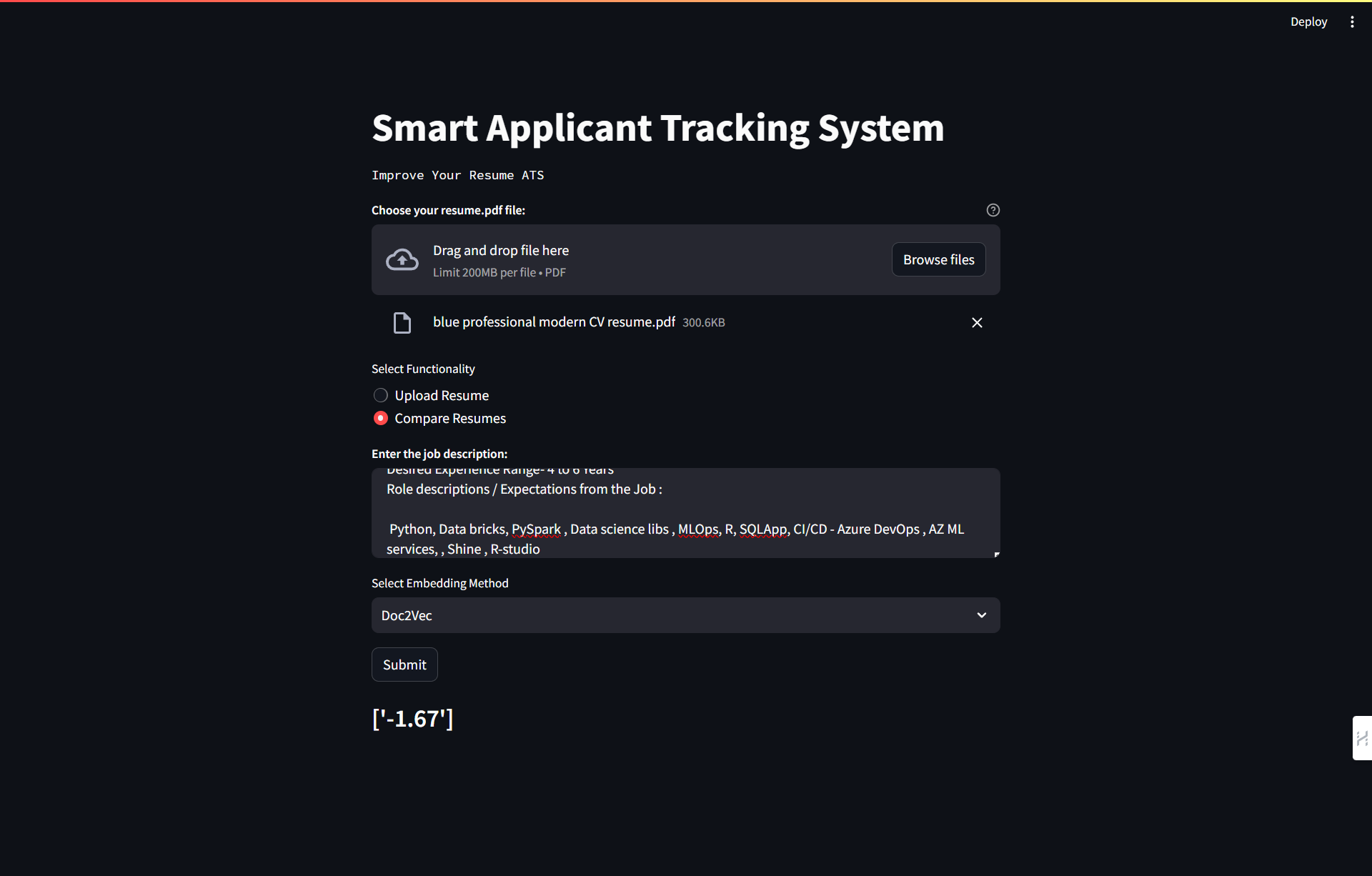
**Screenshots of webpages**

****

**Gemini**

****

**Bert**

****

**Doc2Vec**

**Chapter 6**

**Testing**

**6.1 Testing strategies**

To ensure the reliability and effectiveness of the Resume Screening App, rigorous testing strategies were employed. The testing process encompassed various phases, including:

1. **Unit Testing:** Individual components of the application, such as data preprocessing functions, BERT integration, and cosine similarity calculation, were subjected to unit tests to verify their functionality in isolation.
2. **Integration Testing:** The integration of different modules within the backend processing pipeline, including BERT model integration and cosine similarity calculation, was thoroughly tested to ensure seamless interaction and data flow between components.
3. **End-to-End Testing:** The entire application, from user interaction in the Streamlit interface to backend processing and result presentation, underwent end-to-end testing to validate its functionality in real-world usage scenarios.

### **6.2 Test Case Reports:**

During the testing process, various test cases were designed and executed to evaluate the app's performance across different scenarios. Some key test case reports include:

* **Input Validation Test Case**: Providing invalid input, such as incorrect file formats or empty fields.
  + **Expected Outcome**: Application should prompt users for correct input and provide informative error messages.
* **Accuracy Test Case**: Inputting resumes and job descriptions with known similarity scores.
  + **Expected Outcome**: Application should accurately compute similarity scores matching the predefined ground truth.
* **Scalability Test Case**: Processing a large volume of resumes and job descriptions simultaneously.
  + **Expected Outcome:** Application should maintain real-time or near-real-time performance without crashing or significant slowdown.
* **Customization Test Case**: Adjusting similarity score thresholds or weighting factors.
  + **Expected Outcome:** Application should reflect user-customized parameters accurately in the similarity score computation.
* **User Interface Test Case**: Interacting with the Streamlit interface to upload resumes, enter job descriptions, and view results.
  + **Expected Outcome:** Application should provide a user-friendly experience with intuitive navigation and clear presentation of results.

**Chapter 7**

**Conclusion and Future work**

**7.1 Conclusion :**

In conclusion, the resume screening application presents a comprehensive solution for efficiently matching resumes to job descriptions, streamlining the recruitment process. By harnessing advanced technologies such as Streamlit for user interface development, Hugging Face Transformers and Gensim for natural language processing tasks, and PyPDF2, PDFPlumber, and NLTK for document parsing and text processing, the application offers robust functionality and accurate results. Additionally, the integration of Generative AI models like Gemini enhances the application's capabilities by providing tailored recommendations for resume improvement. Overall, the application's innovative approach and utilization of state-of-the-art technologies contribute to its effectiveness in optimizing candidate evaluation and selection processes, ultimately benefiting both recruiters and job seekers alike.

**7.2 Future Work:**

Moving forward, there are several avenues for enhancing the capabilities and usability of the Resume Screening App:

**1. Enhanced Feature Set :** Expand the application's feature set to include additional functionalities such as resume parsing, sentiment analysis, and skill matching. These features can provide deeper insights into candidate qualifications and compatibility with job requirements.

**2. Integration with External APIs :** Integrate the application with external APIs or databases to access additional data sources, such as job boards, professional networking platforms, or applicant tracking systems. This integration can enrich the evaluation process with more comprehensive candidate profiles and job listings.

**3. Advanced Analytics :** Implement advanced analytics and visualization techniques to provide users with deeper insights into candidate-job matches. This may include predictive analytics, trend analysis, and cohort comparisons to identify patterns and optimize hiring strategies.

**4. User Feedback Mechanism :**  Establish a user feedback mechanism within the application to gather input from recruiters and hiring managers regarding the effectiveness and usability of the app. This feedback can inform future updates and improvements to better meet user needs.

**5. Machine Learning Model Optimization :**  Continuously optimize the machine learning models used in the application, including fine-tuning hyperparameters, updating training data, and exploring alternative model architectures. This ensures the application remains state-of-the-art and adapts to evolving recruitment trends and requirements.

By pursuing these avenues for future work, the Resume Screening App can further enhance its effectiveness, usability, and impact on the recruitment process, ultimately empowering organizations to make more informed hiring decisions and build high-performing teams.

**Chapter 8**

**References**

1 . Priyanka, J. Himabindu, and Nikhat Parveen. "An efficient resume skill extraction using deep feature-based AGT optimized K means clustering." *Multimedia Tools and Applications* (2024): 1-20.

2. Chen, Qiqi, Dexu Kong, Yongchi Zhu, Zitao Shen, Chengyue Lu, Yang Li, and Lin Zhang. "CareerMiner: Automatic extraction of professional network from large Chinese resume data." *Franklin Open* 6 (2024): 100065.

3.Gan, Chengguang, Qinghao Zhang, and Tatsunori Mori. "Application of LLM Agents in Recruitment: A Novel Framework for Resume Screening." *arXiv preprint arXiv:2401.08315* (2024).

4.Bounab, Yazid, Mourad Oussalah, Nabil Arhab, and Salah Bekhouche. "Towards job screening and personality traits estimation from video transcriptions." *Expert Systems with Applications* 238 (2024): 122016.

5.Werner, M., & Laber, E. (2024). Extracting section structure from resumes in Brazilian Portuguese. *Expert Systems with Applications*, *242*, 122495.

6. Ujlayan, Anshul, Sanjay Bhattacharya, and Sonakshi. "A Machine Learning-Based AI Framework to Optimize the Recruitment Screening Process." *International Journal of Global Business and Competitiveness* (2024): 1-16.

7. Zhou, Nai, Nianmin Yao, Nannan Hu, Jian Zhao, and Yanan Zhang. "CDGAN-BERT: Adversarial constraint and diversity discriminator for semi-supervised text classification." *Knowledge-Based Systems* 284 (2024): 111291.

8.Li, Qiuchi, and Christina Lioma. "Joint Extraction and Classification of Danish Competences for Job Matching." In *European Conference on Information Retrieval*, pp. 475-483. Cham: Springer Nature Switzerland, 2023.