A WEB BASED APPLICATION FOR AUTOMATED RESUME SCREENING AND ANALYSIS

A PROJECT REPORT

Submitted by

GANESH KUMAR TANGUTURI[RA2011033010118]
AMITHS SAIBABA[RA2011026010432]

Under the Guidance of

(DR.PG.OM PRAKASH)

(Assistant Professor, CINTEL)

in partial fulfillment of the requirementsfor the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING
with specialization in (SOFTWARE ENGINEERING)

DEPARTMENT OF COMPUTATIONAL INTELLIGENCE
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE ANDTECHNOLOGY
KATTANKULATHUR- 603 203

MAY 2024

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

Department of Computational Intelligence

SRM Institute of Science & Technology

Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course **:BTECH/CSE**

Student Name **:GANESH KUMAR TANGUTURI , AMITH SAIBABA**

Registration Number **:RA2011033010118 , RA2011026010432**

Title of Work **: A WEB BASED APPLICATION FOR AUTOMATED RESUME SCREENING AND ANALYSIS**

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with theUniversity policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that 18CSP109L/18CSP111L project report [titled "**A WEB BASED APPLICATION FOR AUTOMATED RESUME SCREENING AND ANALYSIS** " is the bonafide work of "**GANESH KUMAR TANGUTURI [RA2011033010118], AMITH SAIBABA [RA2011026010432]**" who carried out the project work[internship] under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	SIGNATURE
DR.PG.OM PRAKASH	**DR.R.ANNIE UTHRA**
ASSISTANT PROFESSOR	HEAD OF DEPARTMENT
DEPARTMENT OF	DEPARTMENT OF
COMPUTATIONAL INTELLIGENCE	COMPUTATIONAL INTELLIGENCE

SIGNATURE

DR.S.SADAGOPAN
PANEL HEAD
DEPARTMEBT OF
COMPUTATIONAL INTELLIGENCE

Examiner I

Examiner II

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support. We would like to express our warmth of gratitude to our Registrar **Dr. S. Ponnusamy**, for his
encouragement.

We wish to thank **Dr. T.V. Gopal**, Professor & Dean, College of Engineering and Technology, SRM Institute of Science and Technology, for his valuable support and encouragement. We would like to express our heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman**, for imparting confidence to complete my course project. We extend my gratitude to Dr. **R. Annie Uthra** , Professor and Head, Department of Computational Intelligence.

We wish to express our sincere thanks to Panel Head**, Dr. S. Sadagopan,** Associate Professor, Department of Computational Intelligence, for his constant encouragement and support. We are highly thankful to our Project Guide, **Dr. P.G.Om prakash,** Associate Professor, Department of Computational Intelligence for his assistance, timely suggestion, and for providing us an opportunity to pursue our project under her mentorship. She provided us the freedom and support to explore the research topics of our interest. Her passion for solving real problems and making a difference in the world has always been inspiring.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

ABSTRACT

This project develops an innovative web-based application designed to streamline the resume screening process for recruiters and hiring managers. Utilizing cutting-edge natural language processing (NLP) techniques, the application analyzes resumes in depth, comparing them against job descriptions to identify the most promising candidates. The core of the application is built on Python, leveraging powerful libraries such as Genism for topic modeling, Pandas for data manipulation, and Streamlit for creating an interactive user interface.

At its heart, the application performs similarity scoring between the text of each resume and the job description, ranking the resumes to help recruiters quickly identify the top candidates. It employs Latent Dirichlet Allocation (LDA) for topic modeling, extracting meaningful themes from the resumes, which enables a deeper understanding of the candidates' expertise and areas of interest. Visualization tools like Plotly and Matplotlib are integrated to present the analysis results through interactive graphs and word clouds, enhancing the user experience and providing insightful data representations.

Furthermore, the application supports file uploads, allowing users to easily submit resumes and job descriptions. It includes functionality for handling multiple job descriptions, enabling recruiters to select the most relevant ones for their current hiring needs. The application is designed with a focus on user experience, featuring an intuitive interface that guides the user through the process of uploading documents, selecting job descriptions, and viewing the analysis results, including ranked resumes and topic distributions.

This project represents a significant advancement in recruitment technology, offering a practical solution to the challenges of resume screening. By automating the initial stages of the recruitment process, it saves time, reduces manual effort, and increases the accuracy of candidate selection, ultimately aiding organizations in finding the right talent efficiently.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Keywords:

NLP, ML, MySQL, Elastic Search, Ranking, Scrapy, Scikit learn, SQL

Glossary:

A

ATS: Applicant Tracking System

D

DES: Data Envelopment Analysis

F

Free Grammer: Set of rules and grammar used analyzing of programming languages, parsing and manage the structure of the document.

G

GUI: Graphical User Interface, is a type of interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation.

H

History log: A snapshot of file history is just a click away with the File History Log.

M

Metaphor: A figure of speech in which a word or phrase is applied to an object or action to which it is not literally applicable.

Metonymy: The substitution of the name of an attribute or adjunct for that of the thing meant.

Morphoms: It is a basic unit of meaning.

MySQL: SQL language.

N

NLIDB: Natural Language Interface for Database, interface to interact with the database system.

NLTK: Natural Language Toolkit, python library used to process the natural language. **NLP:** Natural Language Processing, field of Artificial Intelligence used to process human natural language.

p

Presuppositions: Presuppositions in general are beliefs underlying a system. The presuppositions of NLP are beliefs that guide and have guided the development of NLP. **Php:** Hypertext Preprocessor, is a server-side scripting language designed for web development but also used as a general-purpose programming language.

Pattern: Web mining module for Python, with tools for scraping, natural language processing, machine learning, network analysis and visualization.

Phonology: The system of contrastive relationships among the speech sounds that constitute the fundamental components of a language.

Parser: A parser is a program, usually part of a compiler.

Q

Quantifications: Use of an indicator of quantity in linguistic term.

S

Speech tagging: Process of marking up a word in a text (corpus) as corresponding to a particular part of speech.

SQL: Structured Query Language is a special-purpose programming language designed for managing data held in a database management system.

Shallow: Things that aren't very deep, thus Deep linguistic processing is a natural language processing framework.

Software as Service (SaaS): SaaS is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network.

Scrapy: Python tool for Web-Crawlers

Scikit-learn: It is open-source python library which contains optimized implementation of machine learning algorithm.

T

Token: Each separate word of the sentence.

TK: Tk/Tcl has long been an integral part of Python. It provides a robust and platform independent windowing toolkit, that is available to Python programmers using the tkinter package.

CHAPTER 1

INTRODUCTION

1.1 Statement of Project

1.1.1 Need of Resume Ranking System:

The need for a resume ranking system arises from the challenges faced by recruiters in efficiently identifying the most suitable candidates from a large pool of applicants. Here's a breakdown of the key points:

Volume of Applicants:

With the increasing prevalence of online job postings and applications, recruiters often receive a high volume of resumes for each job opening. Manual screening of each resume is time-consuming and labor-intensive, leading to delays in the hiring process.

Diverse Formats and Content:

Resumes come in various formats and styles, making it challenging to compare candidates objectively. Additionally, the content of resumes may vary widely, making it difficult to assess the relevance of skills and experiences to the job requirements.

Objective Evaluation:

Human bias can influence the screening process, leading to subjective judgments and potential discrimination. A resume ranking system aims to minimize bias by objectively evaluating candidates based on predefined criteria and job requirements.

Efficiency and Time Savings:

By automating the screening process and ranking resumes based on their relevance to the job description, recruiters can save time and focus their efforts on reviewing the most promising candidates. This efficiency gains significant importance, especially in fast-paced hiring environments.

Improved Candidate Experience:

A streamlined screening process ensures that qualified candidates are promptly identified and contacted, enhancing their experience with the recruitment process. Conversely, prolonged screening times or lack of feedback can lead to candidate dissatisfaction and negative perceptions of the employer brand.

Enhanced Decision-making:

By providing recruiters with a prioritized list of candidates based on their suitability for the role, a resume ranking system enables more informed decision-making. Recruiters can focus on candidates who are the best fit for the job, increasing the likelihood of successful hires.

1.1.2 Problems and Solution:

in the context of a resume ranking system, several problems may arise during the recruitment process, along with corresponding solutions:

Problems:

High Volume of Resumes:

Recruiters often receive a large number of resumes for each job opening, making manual screening time-consuming and inefficient.

Subjective Evaluation:

Human bias can influence the screening process, leading to inconsistent judgments and potential discrimination against certain candidates.

Diverse Resume Formats and Content:

Resumes come in various formats and styles, making it challenging to compare candidates objectively. Additionally, the content of resumes may vary widely, making it difficult to assess the relevance of skills and experiences to the job requirements.

Inefficient Use of Resources:

Spending excessive time on resume screening can divert resources away from other critical recruitment tasks, such as interviewing and candidate engagement.

Solutions:

Automated Resume Screening:

Implement an automated system that leverages natural language processing (NLP) and machine learning algorithms to analyze and rank resumes based on their relevance to the job description. This significantly reduces the time and effort required for manual screening.

Objective Evaluation Criteria:

Define clear and objective evaluation criteria based on the job requirements, such as specific skills, qualifications, and experience levels. Use these criteria to standardize the screening process and minimize bias.

Normalization of Resume Data:

Preprocess resumes to standardize the format and structure, ensuring consistency in the analysis. Techniques such as text normalization, tokenization, and removal of irrelevant information can help streamline the comparison process.

Prioritization of Resumes:

Develop a scoring mechanism that ranks resumes based on their similarity to the job description, considering factors such as keyword relevance, skills match, and experience level. This prioritizes candidates who are the best fit for the role, allowing recruiters to focus their attention on top candidates.

Integration with Applicant Tracking Systems (ATS):

Integrate the resume ranking system with existing ATS platforms to streamline the recruitment workflow. This ensures seamless data exchange between screening, interviewing, and hiring stages, improving overall efficiency and collaboration among recruiters.

Continuous Improvement:

Regularly review and refine the resume ranking algorithms based on feedback from recruiters and hiring managers. Incorporate new data sources, update evaluation criteria, and fine-tune the scoring models to adapt to changing hiring needs and industry trends.

By addressing these problems and implementing corresponding solutions, organizations can streamline their recruitment processes, improve the quality of hires, and enhance the overall candidate experience. Additionally, automated resume screening systems enable recruiters to make data-driven decisions and focus their efforts on engaging with top candidates, ultimately driving better recruitment outcomes..

1.2 Motivation

The motivation behind the development of a resume ranking system stems from several key factors:

Efficiency Improvement:

High Volume of Applicants: With the increasing accessibility of job postings through online platforms, recruiters are inundated with a large volume of resumes for each job opening. Manual screening of these resumes is time-consuming and resource-intensive, leading to delays in the recruitment process.

Time Constraints:

Recruiters often face tight deadlines to fill job vacancies, especially in fast-paced industries where talent acquisition is competitive. The need to review numerous resumes within a short timeframe necessitates a more efficient screening process.

Quality Enhancement:

Identifying Top Candidates: Traditional resume screening methods may overlook qualified candida tes or fail to prioritize those who are the best fit for the job. A resume ranking system aims to identify top candidates based on their skills, experiences, and alignment with job requirements.

Objective Evaluation:

Human biases in the screening process can inadvertently exclude qualified candidates or favor certain demographics over others. By implementing an automated ranking system, organizations can ensure more objective evaluation criteria and minimize bias.

Cost Reduction:

Resource Optimization: Manual resume screening requires significant time and effort from recruiters, which could be better utilized for other strategic recruitment activities such as interviewing and candidate engagement. By automating the screening process, organizations can optimize resource allocation and reduce recruitment costs.

Increased Productivity:

Streamlining the resume screening process allows recruiters to focus onhigh-value tasks that require human intervention, such as assessing cultural fit, conducting interviews, and negotiating job offers. This improves overall productivity and efficiency within the recruitment team.

Competitiveness:

Talent Acquisition Advantage: In today's competitive job market, organizations need to attract and retain top talent efficiently. A resume ranking system provides a competitive advantage by enabling faster identification of qualified candidates and expediting the hiring process.

Enhanced Candidate Experience:

A streamlined recruitment process, facilitated by a resume ranking system, enhances the candidate experience by reducing waiting times and providing timely feedback. This positive experience can improve employer branding and attract top talent to the organization.

Technological Advancements:

Advances in AI and NLP: Recent advancements in artificial intelligence (AI) and natural language processing (NLP) have made it possible to develop sophisticated algorithms for analyzing and ranking resumes. These technologies enable organizations to leverage data-driven insights for more effective talent acquisition.

Availability of Tools and Libraries:

The availability of open-source libraries and tools for NLP and machine learning, along with the proliferation of cloud computing resources, has lowered the barrier to entry for developing and implementing resume ranking systems.

1.3 Objective and scope

1.3.1 Objective:

The primary objective of implementing a resume ranking system is to streamline the recruitment process and improve the efficiency and effectiveness of talent acquisition efforts. This objective encompasses several specific goals:

Automated Screening: Develop a system capable of automatically analyzing and evaluating resumes to identify qualified candidates based on predefined criteria and job requirements. By automating the initial screening process, recruiters can focus their time and resources on interacting with top candidates and conducting in-depth assessments.

Objective Evaluation: Ensure that the resume ranking system employs objective criteria for evaluating candidate suitability, minimizing the influence of human biases and subjective judgments. By using data-driven algorithms and predefined metrics, the system aims to provide fair and consistent evaluations of candidate qualifications.

Efficiency Enhancement: Expedite the screening process by efficiently processing a large volume of resumes and providing recruiters with a prioritized list of top candidates. By ranking resumes based on their relevance to the job description and desired qualifications, the system accelerates the identification of suitable candidates and reduces time-to-hire.

Quality Improvement: Enhance the quality of candidate selection by identifying top candidates who closely match the requirements of the job role. By leveraging advanced technologies such as natural language processing (NLP) and machine learning, the system can extract relevant information from resumes and assess candidate suitability with greater accuracy.

Cost Reduction: Optimize resource allocation and minimize recruitment costs by automating repetitive tasks and reducing reliance on manual screening processes. By streamlining the recruitment workflow, the system enables recruiters to achieve more with fewer resources, leading to cost savings for the organization.

Competitive Advantage: Gain a competitive edge in talent acquisition by attracting and retaining top talent more efficiently. By providing a seamless and user-friendly candidate experience, the system enhances the organization's employer brand and positions it as an attractive destination for skilled professionals.

Scalability and Adaptability: Design the resume ranking system to be scalable and adaptable to accommodate changes in recruitment needs and evolving job market dynamics. By leveraging cloud-based infrastructure and flexible architecture, the system can scale to handle varying volumes of resumes and adapt to new recruitment requirements.

Overall, the objective of the resume ranking system is to revolutionize the recruitment process by leveraging technology to identify, evaluate, and prioritize candidates effectively and efficiently. By aligning with these objectives, organizations can optimize their talent acquisition strategies and achieve better outcomes in hiring

1.3.2 Scope:

The scope of the resume ranking system project encompasses various aspects related to the development, implementation, and utilization of the system within the context of talent acquisition and recruitment processes. Here are the key components included within the scope:

Feature Development:

Define and develop the key features and functionalities of the resume ranking system, including resume and job description upload, text preprocessing and analysis, similarity scoring and ranking, topic modeling and insight generation, interactive data visualization, and user interaction and customization options.

Technology Stack Selection:

Choose appropriate technologies, frameworks, and libraries to support the development of the resume ranking system, considering factors such as scalability, performance, ease of integration, and compatibility with existing infrastructure.

Data Processing and Analysis:

Implement algorithms and techniques for processing and analyzing resume and job description data, including natural language processing (NLP), machine
learning, semantic similarity scoring, topic modeling, and data visualization.

Integration with HR Systems:

Explore options for integrating the resume ranking system with existing human resources (HR) management systems, applicant tracking systems (ATS), and recruitment software to facilitate seamless data exchange and workflow automation.

User Interface Design:

Design an intuitive and user-friendly interface for the resume ranking system, allowing recruiters to easily upload resumes and job descriptions, customize preferences, visualize insights, and interact with the system to make informed decisions.

Testing and Validation:

Conduct thorough testing and validation of the resume ranking system to ensure accuracy, reliability, and robustness across different use cases, data sources, and scenarios. Perform unit testing, integration testing, and user acceptance testing to identify and address any issues or discrepancies.

Deployment and Deployment:

Deploy the resume ranking system in a production environment, either on-premises or in the cloud, ensuring scalability, security, and performance. Implement monitoring and maintenance procedures to support ongoing operation and management of the system.

Training and Support:

Provide training and support resources to users and administrators of the resume ranking system, including documentation, tutorials, and helpdesk services. Address user queries, troubleshoot issues, and provide guidance on system usage and best practices.

Evaluation and Feedback:

Evaluate the effectiveness and impact of the resume ranking system through metrics such as time-to-hire, candidate quality, recruiter satisfaction, and cost savings. Gather feedback from users and stakeholders to identify areas for improvement and inform future enhancements.

Continuous Improvement:

Continuously iterate and improve the resume ranking system based on user feedback, technological advancements, and changes in recruitment practices and requirements. Incorporate new features, updates, and optimizations to enhance the system's functionality, performance, and usability over time.

top talent.

1.4 System Architecture

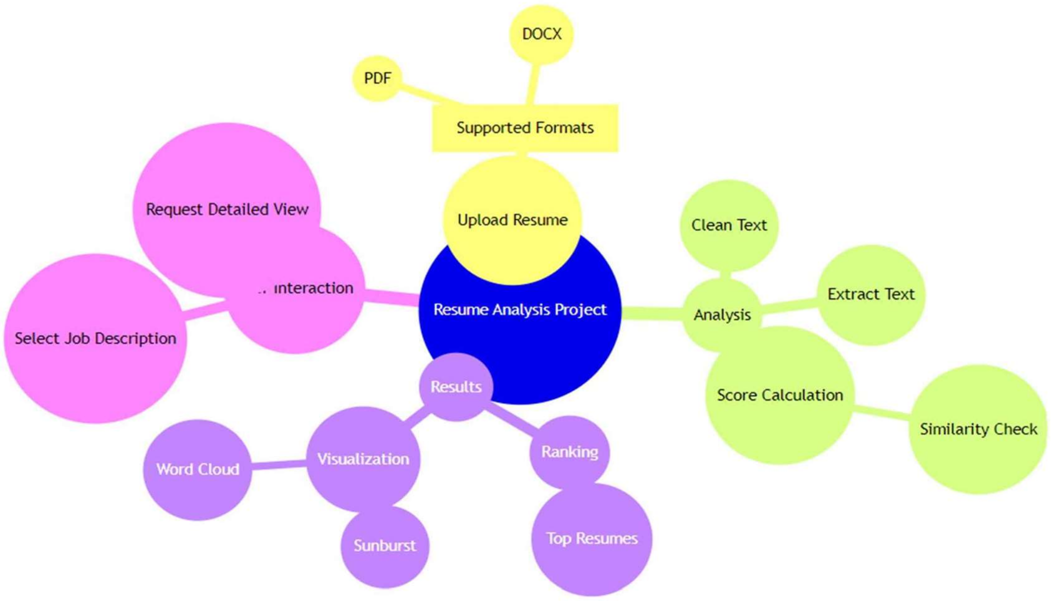


fig1.4.1. System Architecture

1. **User Interface (UI) Layer:**

- Responsible for interacting with users, including recruiters and administrators.
- Provides functionalities for uploading resumes and job descriptions, customizing preferences, visualizing insights, and interacting with the system.
- Implements an intuitive and user-friendly interface using web technologies or desktop applications.
- Subcomponents:
- Upload Interface: Allows users to upload resumes and job descriptions in various formats.
- Customization Options: Provides settings for customizing similarity thresholds, topic preferences, and visualization preferences.
- Visualization Widgets: Displays interactive charts, graphs, and word clouds for presenting insights and analysis results.

2. **Application Layer:**

- Manages the core logic and functionalities of the resume ranking system.
- Implements algorithms for text preprocessing, similarity scoring, topic modeling, and data visualization.
- Orchestrates communication between the UI layer, data layer, and external services.
- Subcomponents:
- Text Preprocessing Module: Handles tokenization, stop word removal,

normalization, and other preprocessing tasks for resumes and job descriptions.

- Similarity Scoring Engine: Calculates semantic similarity scores between resumes and job descriptions using NLP techniques and machine learning models.

3. **Data Layer:**

- Stores and manages resume and job description data, as well as metadata generated during analysis.
- Provides access to structured and unstructured data for processing and analysis.
- Supports efficient retrieval, storage, indexing, and querying of large datasets.
- Subcomponents:
- Resume Database: Stores resumes in a structured format, allowing for efficient storage and retrieval based on candidate attributes.
- Job Description Repository: Stores job descriptions and associated metadata, enabling quick access and comparison with resumes.
- Metadata Store: Stores analysis results, similarity scores, topic distributions, and other metadata generated during the analysis process.

4. **Integration Layer:**

- Facilitates integration with external systems, services, and APIs.
- Enables communication with HR management systems, applicant tracking systems, and other recruitment software.
- Supports data exchange, workflow automation, and real-time updates.
- Subcomponents:
- API Gateway: Acts as a single entry point for external integrations, providing APIs for data exchange and system interaction.
- Webhooks and Event Handlers: Listens for events and triggers from external systems, initiating actions or updates within the resume ranking system.

5. **Security Layer:**

- Ensures the security, confidentiality, and integrity of data and interactions within the system.
- Implements authentication, authorization, encryption, and access control mechanisms.
- Monitors and detects potential security threats and vulnerabilities.
- Subcomponents:
- Authentication Service: Handles user authentication and authorization, verifying user identities and permissions before granting access to system functionalities.
- Encryption Mechanisms: Encrypts sensitive data at rest and in transit, preventing unauthorized access or interception.

6. **Scalability and Performance Layer:**

- Addresses scalability and performance requirements to support large-scale deployments and high volumes of data and users.
- Implements mechanisms for horizontal scaling, load balancing, and resource optimization*

- Monitors system performance metrics and adjusts resources dynamically to meet demand.
- Subcomponents:
- Load Balancer: Distributes incoming traffic across multiple instances or nodes of the application, ensuring optimal resource utilization and performance.
- Autoscaling Mechanism: Automatically adjusts the number of application instances or resources based on workload and performance metrics, scaling up or down as needed.
- Performance Monitoring Tools: Collects and analyzes performance metrics, such as response times, throughput, and resource utilization, to identify bottlenecks and optimize system performance.

7. **Infrastructure Layer:**

- Provides the underlying infrastructure and resources for hosting and running the resume ranking system.
- Includes servers, virtual machines, containers, networking components, and cloud services.
- Ensures reliability, availability, and fault tolerance of the system.
- Subcomponents:
- Cloud Infrastructure: Utilizes cloud computing services (e.g., AWS, Azure, GCP) for scalable and flexible deployment of system components.
- Containerization Platform: Uses container orchestration tools (e.g., Kubernetes, Docker Swarm) to manage and deploy application containers across multiple nodes or clusters.
- Networking Infrastructure: Establishes secure and reliable network connections between system components, users, and external services, ensuring data transmission and communication.

8. **Monitoring and Logging Layer:**

- Monitors system health, performance, and availability in real-time.
- Collects and aggregates logs, metrics, and events for analysis and troubleshooting.
- Provides visibility into system operations and detects anomalies or issues.
- Subcomponents:
- Logging Framework: Captures and stores log messages from different system components, applications, and services, facilitating troubleshooting and auditing.
- Monitoring Tools: Monitors system metrics, resource usage, and application performance, generating alerts or notifications for abnormal conditions or thresholds.
- Dashboard and Reporting Tools: Visualizes monitoring data and performance metrics in dashboards and reports, providing insights into system behavior and trends.

9. **Backup and Recovery Layer:**

- Implements backup, disaster recovery, and data protection mechanisms to safeguard against data loss or system failures.
- Performs regular backups of critical data, configurations, and system state.
- Establishes recovery procedures and contingency plans for restoring operations in case of disruptions.
- Subcomponents:
- Backup Systems: Automatically backs up data to redundant storage systems or backup repositories, ensuring data durability and availability in the event of hardware or software failures.

10.**Governance and Compliance Layer:**

- Ensures compliancewith regulatory requirements, industry standards, and organizational policies.
- Implements controls, audits, and safeguards to protect sensitive data and ensure ethical use of the system.
- Facilitates adherence to privacy regulations, data protection laws, and security best practices.
- Subcomponents:
- Compliance Framework: Establishes policies, procedures, and controls for managing data privacy, security, and compliance risks, aligning with relevant regulations and standards (e.g., GDPR, HIPAA, ISO 27001).
- Auditing and Reporting Tools: Conducts regular audits and assessments of the system's security posture, governance practices, and compliance status, generating reports and documentation for internal and external stakeholders.
- Training and Awareness Programs: Provides training and awareness initiatives to educate users and stakeholders about their roles and responsibilities in ensuring compliance with applicable laws, regulations, and policies

Semantic Analysis:

Semantic analysis, also known as semantic understanding or natural language understanding (NLU), is a crucial component of the resume ranking system. It involves interpreting and extracting meaning from unstructured text data, such as resumes and job descriptions, to identify relevant information, skills, experiences, and qualifications. Semantic analysis enables the system to understand the context, intent, and semantics of the text, facilitating accurate matching and ranking of candidates based on their suitability for specific job roles. Below is a detailed overview of semantic analysis within the system:

Text Preprocessing:

Tokenization: Breaks down the text into individual words or tokens, capturing the basic semantic units for analysis.
Stop Word Removal: Eliminates common words (e.g., "the," "and," "is") that do not carry significant semantic meaning and may introduce noise in the analysis.
Normalization: Standardizes text by converting it to lowercase, removing punctuation, and handling special characters or symbols.

Named Entity Recognition (NER):

Identifies and classifies named entities, such as person names, organizations, locations, dates, and numerical expressions, within the text.
Helps extract relevant information, such as candidate names, company names, job titles, and educational institutions, for further analysis.

Part-of-Speech (POS) Tagging:

Assigns grammatical tags to each word in the text, indicating its syntactic role and grammatical category (e.g., noun, verb, adjective).
Provides contextual information about the relationships between words and their semantic roles within sentences.

Dependency Parsing:

Analyzes the grammatical structure of sentences to identify dependencies between words and their syntactic relationships.
Helps understand the semantic hierarchy and dependencies within text, facilitating deeper analysis of sentence meaning and context.

Applies unsupervised machine learning techniques, such as Latent Dirichlet Allocation (LDA) or Non- negative Matrix Factorization (NMF), to identify latent topics or themes within resumes.
Clusters similar words and phrases into coherent topics, providing insights into the thematic strengths and expertise areas of candidates.
Enables the system to understand the broader context and focus areas of candidate profiles, enhancing the accuracy of matching and ranking.

Sentiment Analysis:

Analyzes the sentiment or emotional tone expressed in text, such as positivity, negativity, or neutrality.
Helps assess the overall tone of resumes and job descriptions, identifying sentiment-rich keywords or phrases that may indicate candidate preferences, achievements, or attitudes.
Provides additional context for understanding candidate profiles and job requirements, especially in roles where soft skills and cultural fit are important factors.
Contextual Understanding:

Considers the broader context, domain-specific knowledge, and industry terminology relevant to job roles and requirements.
Incorporates domain-specific dictionaries, ontologies, or knowledge graphs to enrich semantic understanding and improve accuracy in matching candidates to specific domains or industries.
Adapts the semantic analysis process to account for variations in language use, terminology, and context across different domains, regions, or cultures.

CHAPTER 2
LITERATURE SURVEY

1.1 Case Study on talent acquisition

1.1.1 First Generation Hiring Systems

Context: Traditional resume screening is labor-intensive, requiring significant time and effort from human resources personnel to match resumes to job descriptions. With advances in technology, automated systems are increasingly adopted to streamline this process.

Key Studies and Findings:

Automated resume screening tools leverage algorithms to parse, analyze, and rank resumes based on their relevance to specific job criteria (Raj and Pahwa, 2020). These systems can significantly reduce the time to hire and improve the quality of candidate shortlisting.

A study by Bogen and Rieke (2018) emphasizes the importance of transparency and fairness in automated screening systems, noting that poorly designed algorithms can perpetuate biases.

1.1.2 Second Generation Hiring Systems

Context: Visualization tools are essential in HR analytics as they convert complex data sets into understandable and actionable insights, helping stakeholders make informed decisions.

Key Studies and Findings:

According to Kaur and Singh (2019), effective visualization assists HR managers in recognizing patterns and trends that would be obscure in raw data. For example, visual representations of candidate distribution by skills or experience can help in strategic planning and resource allocation.

Plotly and Matplotlib, used in your script, are highlighted in academic literature for their robust capabilities in creating dynamic and static visualizations respectively (Hunter, 2007; Sievert, 2020).

1.1.3 Third Generation Hiring Systems

Context: NLP is a branch of artificial intelligence that deals with the interaction between computers and humans through natural language. It plays a crucial role in extracting insights from text data in resumes.

Key Studies and Findings:

NLP techniques such as topic modeling and keyword extraction are widely used to analyze textual content and derive meaningful patterns and topics (Blei et al., 2003). LDA (Latent Dirichlet Allocation) is particularly noted for its effectiveness in identifying latent topics in large datasets.
Lopes et al. (2019) discuss the application of NLP in automated resume screening, where the technology is used to parse detailed job descriptions and candidate resumes to match skills and qualifications with job req

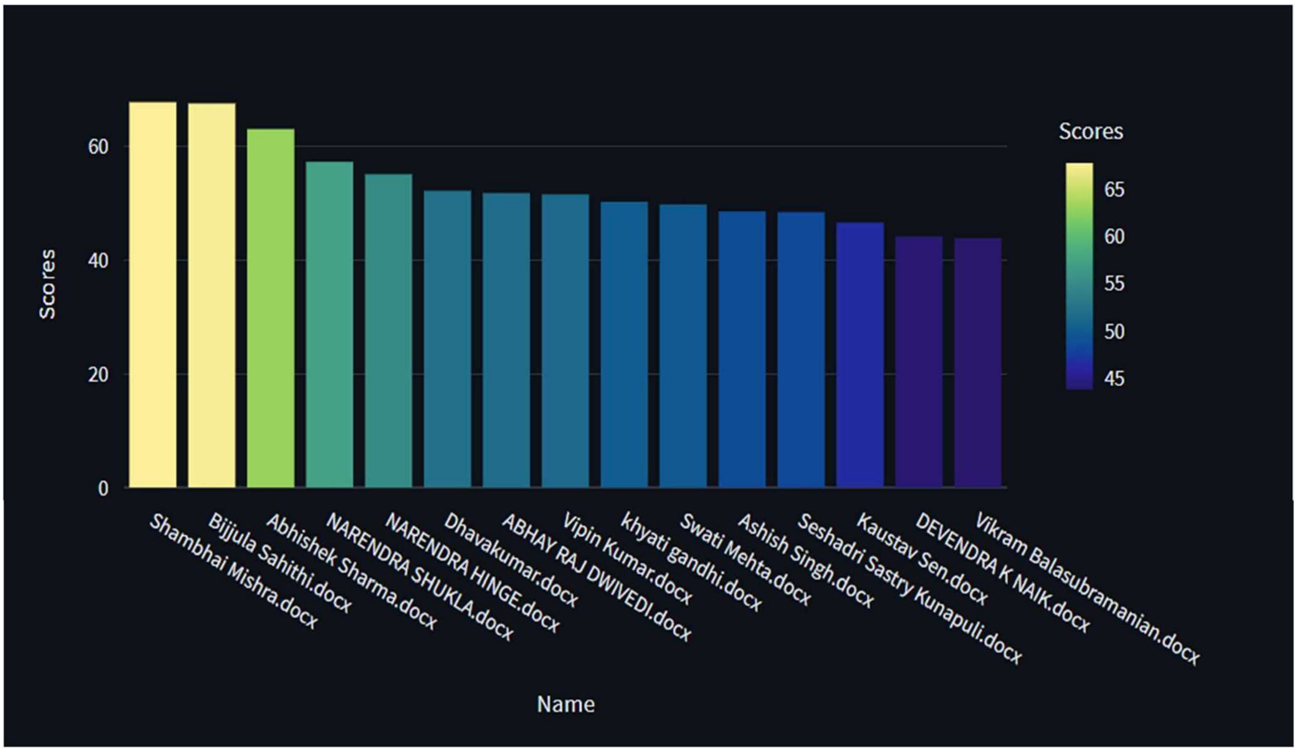


fig2.1. Desired Output

1.2 Intelligent searching

Put simply, Artificial Intelligence or "AI" is an add-on to system, complementing to provide the online recruitment solution . As the name suggests, AI enables a combination of an applicant-tracking system and an artificial intelligence resume parsing, searching and matching engine. The result is a supercharged tool giving incredibly accurate candidate matching to jobs, and 'talent pool' searching that makes other systems look like they're from the stone-age.

1.2.1 Identifying ''best'' applicants in recruiting using data envelopment analysis

Selecting the most promising candidates to fill an open position can be a difficult task when there are many applicants. Context: In the realm of recruitment and human resource management, identifying the "best" applicants from a pool of candidates is a critical challenge. Traditional methods rely heavily on subjective assessments and may not effectively capture the multidimensional aspects of job fitness. Data Envelopment Analysis (DEA), originally developed for assessing the efficiency of decision-making units in operations research, has been adapted to evaluate job candidates.

Key Concepts and Application

Data Envelopment Analysis (DEA): DEA is a non-parametric method used in operations research and economics for the efficiency measurement of decision-making units (DMUs) which convert multiple inputs into multiple outputs. In the context of HR, applicants are viewed as DMUs, where inputs can be qualifications, experience, skills, and outputs are their potential contributions to the organization (Cook and Zhu, 2007).

Application in Recruiting: By using DEA, recruiters can objectively measure and compare the efficiency of candidates based on various criteria, essentially identifying who would deliver the most value relative to the "cost" (e.g., salary demands, training needs) (Emrouznejad and De Witte, 2010). This approach is particularly useful when dealing with multifaceted roles requiring a balance of diverse skills and experiences.

Objective Assessment: DEA provides a quantitative and objective method to assess candidates, which helps in reducing personal biases that might affect the recruitment decisions.

Flexibility: It accommodates multiple inputs and outputs, making it suitable for complex job roles where different skills and experiences contribute to job performance in non- linear ways.

Benchmarking: It not only identifies the best candidates but also highlights areas where other candidates can improve to become more competitive, thereby supporting strategic HR development.

Objective Assessment: DEA provides a quantitative and objective method to assess candidates, which helps in reducing personal biases that might affect the recruitment decisions.

Flexibility: It accommodates multiple inputs and outputs, making it suitable for complex job roles where different skills and experiences contribute to job performance in non- linear ways.

Benchmarking: It not only identifies the best candidates but also highlights areas where other candidates can improve to become more competitive, thereby supporting strategic HR development.

1.3 A Short Introduction to Learning to Rank

Learning to rank refers to machine learning techniques for training the model in a ranking task. Learning to rank is useful for many applications in Information Retrieval, Natural Language Processing, and Data Mining. Intensive stud-ies have been conducted on the problem and significant progress has been made. This short paper gives an introduction to learning to rank, and it specifically explains the fundamental problems, existing approaches, and future work of learning to rank.

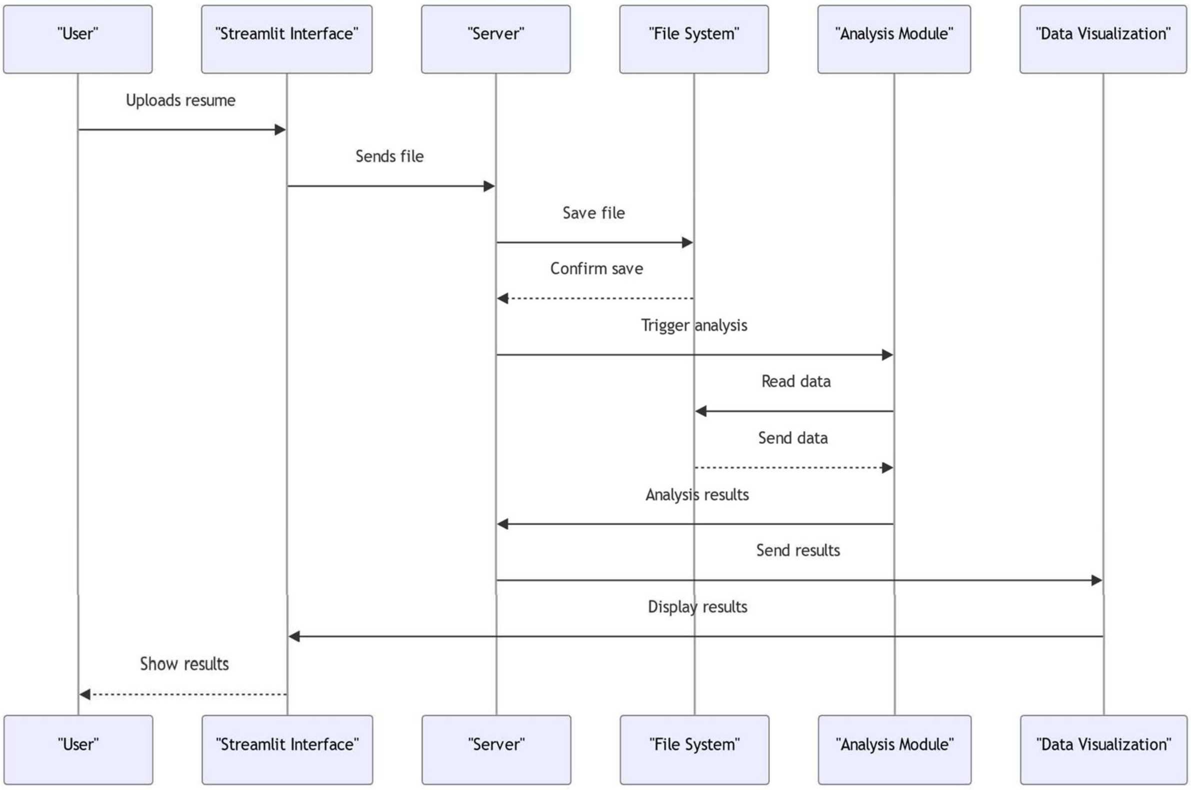


fig2.3. Overview of Ranking System

1.4 Weaknesses

1. Complex Data Requirements:

   DEA requires detailed and quantifiable data on numerous aspects of each candidate's qualifications and expected contributions, which can be difficult to gather comprehensively.

2. Complexity in Interpretation:

   The mathematical nature of DEA may pose interpretation challenges for HR professionals not familiar with quantitative analysis techniques.

3. Relative Evaluation:

   DEA evaluates candidates relative to others in the dataset, which means its effectiveness is contingent on having a diverse and representative pool of candidates.

4. Scalability Issues:

   Applying DEA in large-scale recruiting environments can be computationally intensive and may not be practical for quick decision-making processes.

1.5 How to overcome

Enhance Data Collection:

Invest in robust data collection processes to ensure all relevant candidate information (both qualitative and quantitative) is accurately captured and quantified.

Training and Tools:

Provide training for HR personnel on DEA and its applications or utilize user-friendly software that automates the DEA process and presents results in an easily interpretable format.

Broaden Candidate Pools:

Ensure the candidate pool is diverse and comprehensive to avoid biased results and enhance the effectiveness of relative evaluations.

Integration with Other Methods:

Combine DEA with other decision-making tools and recruitment methods (e.g., behavioral interviews, psychometric tests) to validate and supplement DEA findings.

Regular Review and Update:

Regularly review and update the DEA model and criteria to align with evolving job roles and market conditions, preventing overfitting and maintaining relevance.

Utilize Advanced Computational Resources:

Leverage cloud computing and efficient algorithmic improvements to manage large datasets and complex calculations, enhancing the scalability of DEA applications.

CHAPTER 3
REQUIREMENT ANALYSIS

the adoption of Data Envelopment Analysis (DEA) in recruitment processes requires a thorough assessment of organizational needs, technological infrastructure, data availability, and human resources capabilities. This detailed requirement analysis will help ensure that the implementation is feasible, effective, and aligned with the strategic goals of the organization.

1. Organizational Needs Assessment

Objective Identification: Clearly define what the organization aims to achieve by implementing DEA in recruitment—whether it's improving the efficiency of the hiring process, ensuring better job-candidate matches, or reducing biases.

Stakeholder Involvement: Engage key stakeholders from HR, IT, and executive leadership to align the DEA implementation with broader business objectives and to secure necessary buy-in and resources.

Change Management: Assess the readiness of the organization to adopt new technologies and methodologies in HR processes, including willingness to change and adapt to new workflows.

2. Technology Infrastructure Evaluation

Software and Tools: Identify or develop software that can handle DEA modeling and data analysis. This may involve custom software development or adapting existing tools to include DEA capabilities.

Hardware Requirements: Ensure that the organization has the necessary computational power and data storage capabilities to handle large datasets and complex analyses involved in DEA.

Integration Capabilities: Check the compatibility of the new system with existing HRIS (Human Resource Information Systems) and ATS (Applicant Tracking Systems) to ensure seamless data flow and functionality.

3. Data Management

Data Availability: Evaluate the availability of detailed and reliable data on candidates and job roles that will be required for DEA. This includes both historical and current data on candidate qualifications, experiences, and job performance outcomes.

Data Quality: Assess the quality of existing data and identify gaps. This includes ensuring that data is accurate, complete, and consistently formatted across sources.

Privacy and Security: Develop policies and systems to protect candidate data privacy and comply with relevant data protection regulations (e.g., GDPR, HIPAA).

4. Human Resource Capabilities

Training Needs: Identify training needs for HR staff on DEA concepts, model interpretation, and the use of any new software tools. Consider external consultants or in-house training programs.

Analytical Skills: Assess the current analytical skills within the HR team and the need for hiring new staff or upskilling existing personnel to handle advanced analytical tasks.

Ongoing Support: Plan for ongoing technical support and analytical guidance, either from an internal team or through a partnership with external experts.

5. Evaluation and Continuous Improvement

Performance Metrics: Define clear metrics to evaluate the success of DEA implementation in recruitment, such as improvements in hiring time, quality of hire, and satisfaction ratings from hiring managers.

Feedback Mechanisms: Establish mechanisms for collecting feedback from users (recruiters, hiring managers) and candidates to continually refine the approach.

Regular Review: Schedule regular reviews of the DEA model and its parameters to ensure they remain relevant and effective as job roles and market conditions evolve.

6. Legal and Ethical Considerations

Ethical Use: Ensure the ethical use of DEA, particularly in terms of transparency, fairness, and non-discrimination in candidate evaluation and selection.

Regulatory Compliance: Verify that the use of DEA in recruitment complies with employment laws and regulations specific to the jurisdictions in which the organization operates.

1.1 Software Requirements

The software requirements in this project include:

- Python
- nltk
- ML
- PostgreSQL
- Elasticsearch

Python:

Python is used for creating backbone structure. Python is intended to be a highly readable language. It is designed to have an uncluttered visual layout, it uses whitespace indentation, rather than curly braces or keywords. Python has a large standard library, commonly cited as one of Python's greatest strengths.

Functional Requirements:

Data Input and Management: Ability to input, store, and manage extensive candidate data, including qualifications, experience, and other relevant metrics.

DEA Analysis Engine: Core functionality to perform DEA calculations, generating efficiency scores based on the defined inputs and outputs.

User Interface (UI): Intuitive UI for HR personnel to input data, initiate DEA processes, and view results.

Reporting and Visualization Tools: Advanced reporting features to visualize DEA results in understandable formats such as charts, graphs, and tables.

Integration Capabilities: Seamless integration with existing HR systems like HRIS and ATS for data interchange without manual intervention.

Non-Functional Requirements:

Scalability: Capable of handling increases in data volume and user load without performance degradation.

Security: Robust security measures to ensure data integrity and confidentiality, including compliance with data protection laws.

Usability: High usability to accommodate users with varying levels of technical skills.

Maintainability: Designed for easy updates and maintenance without significant downtime or disruption.

2. **Technology Stack Selection**

Back-End Development: Choose a reliable and scalable back-end technology (e.g., Python for its strong support in data analysis and machine learning libraries).

Front-End Development: User-friendly front-end frameworks (e.g., React or Angular) that offer responsive design and interactive data visualization capabilities.

Database Management: A robust database system (e.g., PostgreSQL or MongoDB) that can handle complex queries and large volumes of data efficiently.

Cloud Infrastructure: Cloud platforms like AWS or Azure for scalable storage and

3.Software Development Lifecycle (SDLC)

Agile Methodology: Adopt an agile development approach to allow for iterative testing, development, and user feedback, ensuring the software meets user needs effectively.

Version Control: Implement version control systems like Git to manage changes and collaboration among development teams.

Continuous Integration/Continuous Deployment (CI/CD): Set up CI/CD pipelines to automate testing and deployment processes, minimizing errors and improving deployment efficiency.

4.Quality Assurance and Testing

Unit Testing: Conduct extensive unit tests to ensure each component of the software functions correctly.

Integration Testing: Test the integration points between the DEA software and existing HR systems to ensure smooth data flows.

Performance Testing: Evaluate the system's performance, especially its ability to process data and generate reports efficiently under load.

Security Audits: Regular security checks and compliance audits to safeguard against data breaches and ensure adherence to legal standards.

3.Training and Documentation

User Manuals: Comprehensive documentation covering all aspects of the software usage.

Training Sessions: Regular training for end-users focusing on how to use the software effectively, interpret DEA results, and integrate findings into the recruitment process.

Technical Support: Ongoing technical support for troubleshooting and assisting users with software issue

4.Compliance and Ethical Considerations

Data Compliance: Ensure the software complies with global data protection regulations such as GDPR.

Ethical Guidelines: Develop guidelines for the ethical use of DEA in recruitment to avoid potential misuse and bias.

Natural Language Processing Tool: Natural Language Toolkit (NLTK) (Python Package) NLTK was originally created in 2001 as part of a computational linguistics course in the Department of Computer and Information Science at the University of Pennsylvania. Since then, it has been developed and expanded with the help of dozens of contributors. It has now

been

adopted in courses in dozens of universities, and serves as the basis of many research projects. NLTK was designed with four primary goals in mind:

Simplicity

To provide an intuitive framework along with substantial building blocks, giving users a practical knowledge of NLP without getting bogged down in the tedious house-keeping usually associated with processing annotated language data.

Consistency

To provide a uniform framework with consistent interfaces and data structures, and easily guessable method names.

Extensibility

To provide a structure into which new software modules can be easily accommodated, including alternative implementations and competing approaches to the same task.

Modularity

To provide components that can be used independently without needing to understand the rest of the toolkit. A significant fraction of any NLP syllabus deals with algorithms and data structures. On their own these can be rather dry, but NLTK brings them to life with the help of interactive graphical user interfaces that make it possible to view algorithms step-by-step. Most NLTK components include a demonstration that performs an interesting task without requiring any special input from the user. An effective way to deliver the materials is through interactive presentation of the examples in this book, entering them in a Python session, observing what they do, and modifying them to explore some empirical or theoretical issue.

1.2 Hardware Requirements

Server Specifications

Processing Power: DEA calculations can be computationally intensive, especially with large datasets. High-performance servers with multi-core processors (e.g., Intel Xeon, AMD Ryzen) are recommended to handle parallel processing effectively.

Memory Requirements: Adequate RAM is crucial for speeding up data processing and analysis. Servers should be equipped with scalable memory options to ensure efficient handling of large- scale data operations without slowdowns.

Storage Solutions: High-speed storage solutions (like SSDs) are necessary for quick data retrieval and storage. The size will depend on the volume of data but should be scalable to accommodate data growth. RAID configurations may be considered for redundancy and data safety.

Network Infrastructure: High-bandwidth network infrastructure is needed to support fast data transfers, especially when integrating with cloud services or external data centers. Gigabit Ethernet or faster is advisable.

2. Client Hardware

Computers and Peripherals: End-user machines (workstations) should have sufficient specifications to run the DEA software efficiently. This includes modern processors, adequate RAM, and high-resolution monitors for detailed data visualization.

Accessibility Devices: Hardware that enhances accessibility for users with disabilities, such as screen readers, alternative input devices, and speech recognition systems, ensuring inclusivity.

3. Data Center Infrastructure

Power Supply Systems: Reliable power supply systems with backup solutions like uninterruptible power supplies (UPS) and generators to ensure continuous operation during power outages.

Cooling Systems: Proper cooling systems to maintain optimal operating temperatures and prevent hardware overheating, which is crucial for server health and efficiency.

Physical Security Systems: Secure access controls, surveillance cameras, and intrusion detection systems to protect hardware from unauthorized access and potential threats.

Virtualization Technology: Using virtualization to create multiple virtual servers on a single physical server can improve the efficiency of resource utilization, allowing for better scalability and isolated testing environments.

4. Integration Considerations

   Compatibility: Ensure hardware compatibility with existing systems, including HRIS and ATS platforms. This may require specific interface cards or adapters.

   Redundancy: Implement redundant hardware systems to ensure high availability and fault tolerance. This includes duplicate servers, network paths, and storage systems to ensure that there is no single point of failure.

1.3 Supportive Operating Systems:

The supported Operating Systems for client include:

- Windows xp onwards
- Linux any flavor.

Windows and Linux are two of the operating systems that will support comparative website.
Since Linux is an open source operating system, This system which is will use in this project is developed on the Linux platform but is made compatible with windows too.The comparative website will be tested on both Linux and windows. The supported Operating Systems for server include: The supported Operating Systems For server include Linux. Linux is used as server operating system. For web server we are using apache 2.0

CHAPTER 4

PROJECT DESIGN

4.1 Design Approach

1. **System Architecture Design High-Level Architecture Three-Tier Architecture:**

Presentation Layer: Manages UI and user interaction.

Application Layer: Houses the business logic, including the DEA com decision support functionalities.

Data Layer: Manages data storage and retrieval, interfacing with databases and external data sources.

Server and Infrastructure

Cloud-Based Solutions: Utilize cloud services (AWS, Azure) for scalable compute and storage resources, ensuring flexibility and high availability.

Load Balancers: Deploy load balancers to distribute user requests efficiently across servers, enhancing performance and reliability.

2.User Interface (UI) and User Experience (UX) Design UI Design

Responsive Design: Ensure the interface is responsive and accessible on various devices, including desktops, tablets, and smartphones.

Intuitive Layout: Create an intuitive layout with clear navigation paths, making it easy for users to upload documents, view results, and access various functionalities.

UX Design

User Feedback: Incorporate feedback mechanisms to capture user satisfaction and areas for improvement.

Progress Indicators: Provide clear progress indicators for lengthy processes, such as file uploads and DEA computations.

3.Data Management Data Modeling

Entity-Relationship Model: Design a comprehensive entity-relationship model that captures all necessary data elements, including candidate details, job specifications, and analysis results.

Normalization: Apply database normalization principles to ensure data integrity and reduce redundancy.

Security and Compliance

Data Encryption: Implement encryption both at rest and in transit to protect sensitive data.

Compliance Standards: Adhere to relevant legal and regulatory standards, such as GDPR for data protection.

4.Integration Strategy Internal Systems Integration

API Development:

Develop RESTful APIs to facilitate seamless integration with existing HR systems like HRIS and ATS.

Data Synchronization:

Implement mechanisms for real-time or batch data synchronization with other enterprise systems.

External Data Sources

Third-Party Services:

Integrate with external data services for additional data verification, enrichment, or analytics capabilities.

API Gateways:

Use API gateways to manage, authenticate, and monitor the APIs' usage and performance.

5.Performance and Scalability Caching Mechanisms

Data Caching: Implement data caching strategies to improve response times and reduce database load for frequently accessed data.

Load Testing: Regularly perform load testing to identify bottlenecks and optimize performance under different load conditions.

6.Security and Data Privacy Security Measures

Role-Based Access Control (RBAC): Implement RBAC to ensure users have access only to appropriate data and functionalities based on their roles.

Audit Trails: Maintain audit trails for all critical actions to enhance security and facilitate audits.

Privacy Protections

Anonymization: Where possible, anonymize sensitive data to protect candidate privacy during analysis.

Data Retention Policies: Establish and enforce data retention policies to comply with legal requirements and minimize risk.

7.Deployment and Maintenance

Continuous Integration/Continuous Deployment (CI/CD)

Automated Deployments: Use CI/CD pipelines for automated testing and deployment, ensuring quick and reliable updates to the application.

Monitoring and Maintenance

System Monitoring: Implement comprehensive monitoring solutions to track system health, usage patterns, and potential security incidents.

Regular Updates: Schedule regular updates and maintenance to ensure the system remains secure and efficient, incorporating the latest security patches and functionality

4.2 Software Architectural Designs

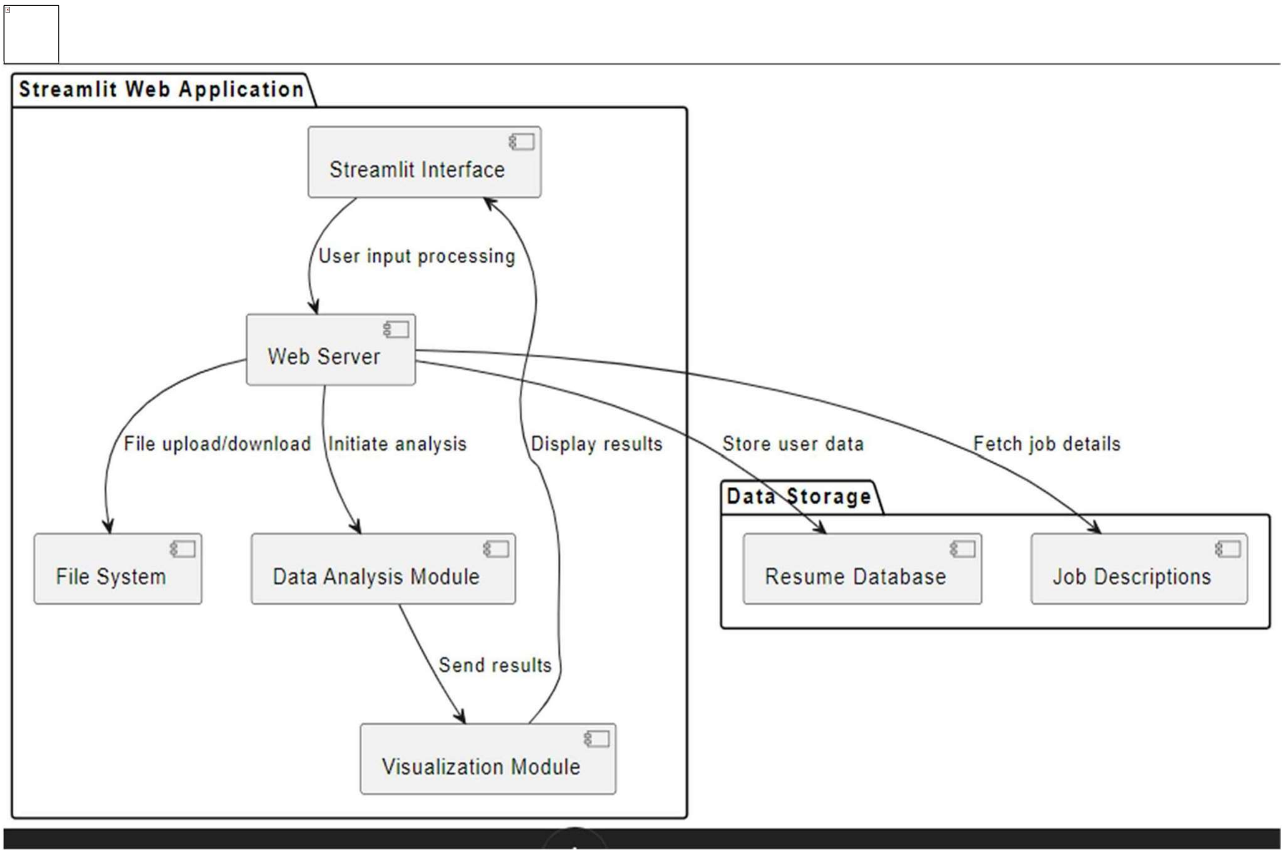
For effective deployment of Data Envelopment Analysis (DEA) in recruitment, a well-considered software architecture is essential. This architecture should support scalability, reliability, and maintainability. Below are succinct outlines of suitable architectural models:

1. Microservices Architecture
   Structure: Divide the application into small, independent services that communicate over a well- defined API.
   Benefits: Enhances scalability and fault isolation. Each service can be independently developed, deployed, and scaled.

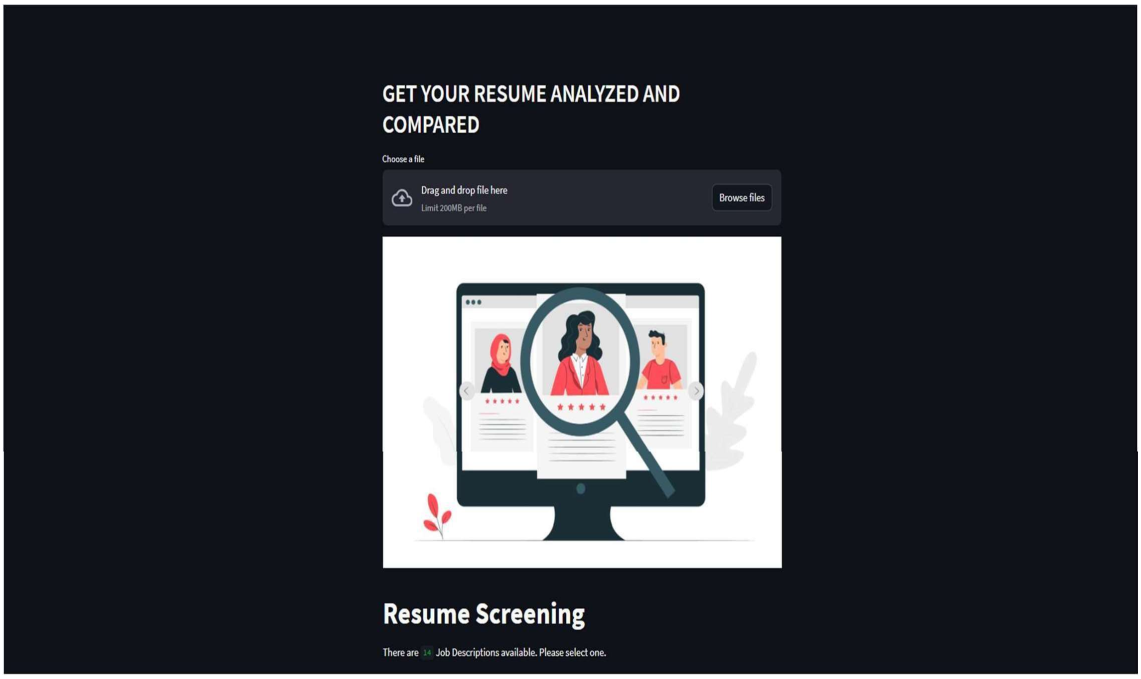
2. serverless Architecture
   Structure: Build and deploy applications as a collection of functions that are triggered by events. Benefits: Reduces operational costs and management overhead, as the cloud provider manages the execution environment. Automatically scales with the application load.
3. Event-Driven Architecture
   Structure: Components react to events (e.g., data inputs) and are loosely coupled through an event management system.
   Benefits: Improves responsiveness and flexibility by decoupling event producers from consumers, facilitating asynchronous processing and dynamic scaling.
4. Layered Architecture
   Structure: Organize the software into layers such as presentation, application, domain, and data access layers.
   Benefits: Promotes separation of concerns and encapsulation, simplifying maintenance and updates.

Software Architecture Design

Fig4.2.1 Software Architecture Design



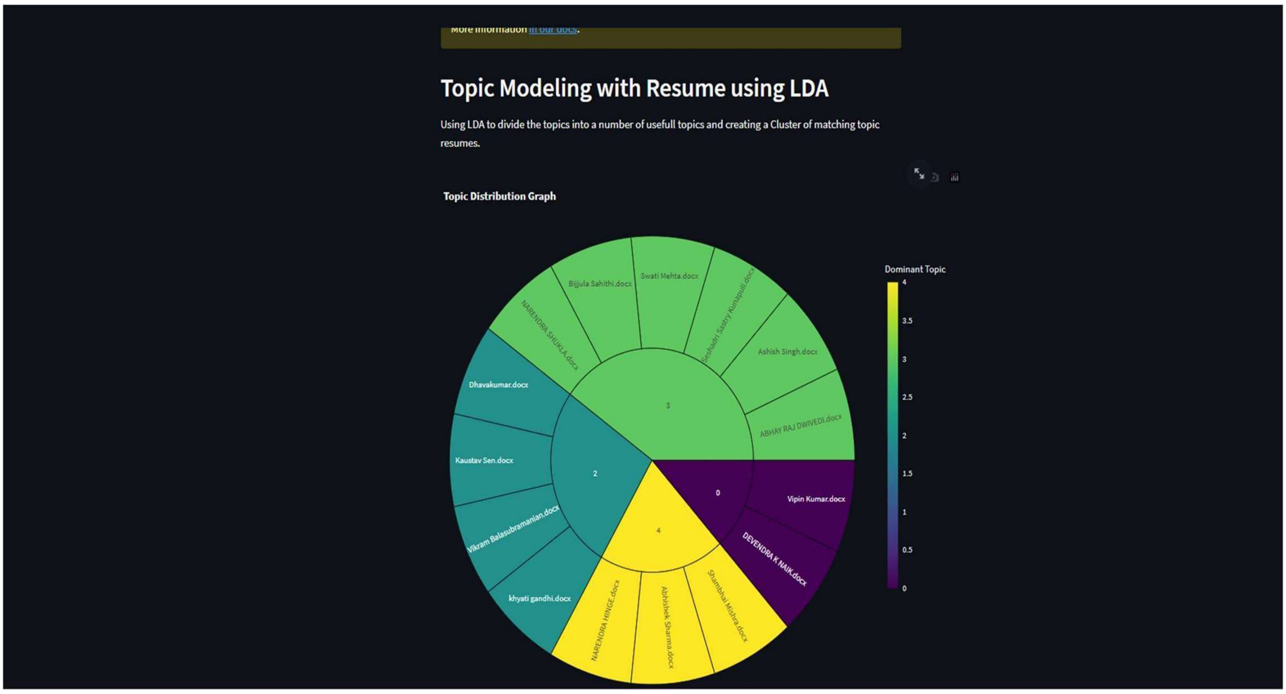
Front End Designs


GET YOUR RESUME ANALYZED AND COMPARED

Resume Screening

There are 14 Job Descriptions available. Please select one.

4.2.2 Front End Designs



More information in our docs.

Topic Modeling with Resume using LDA

Using LDA to divide the topics into a number of usefull topics and creating a Cluster of matching topic resumes.

Topic Distribution Graph

fig4.2.2 Front End

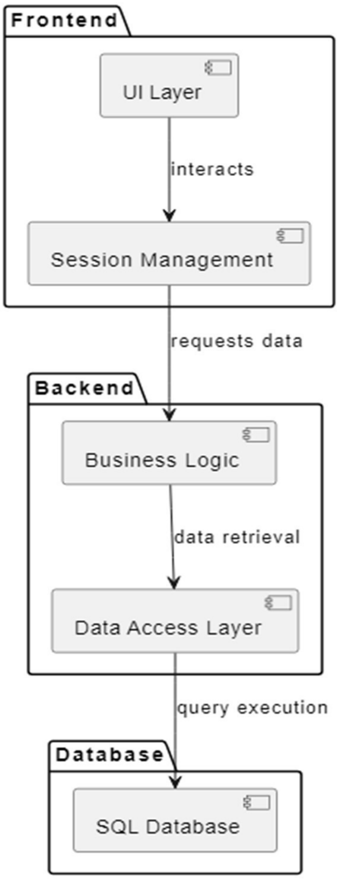


fig4.2.3 Component Diagram

4.2.3 Deployment Diagram

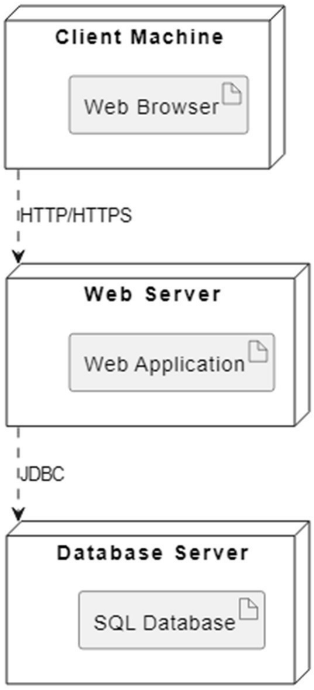


fig4.2.3 Deployment Diagram

4.2.4 State Chart Diagram

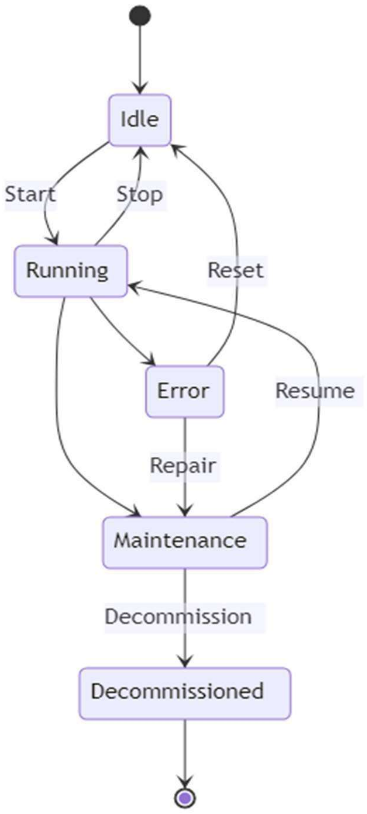


fig 4.2.3 State Chart Diagram

CHAPTER 5

IMPLEMENTATION DETAILS

4.1 Assumptions and Dependencies

4.1.1 Assumptions

Data Quality: Assumes that the data available (resumes, job descriptions, etc.) is accurate, up-to-date, and comprehensive.

Technological Infrastructure: Assumes existing IT infrastructure can support the new system, including server capacities, network capabilities, and data storage.

User Competency: Assumes that users (HR staff, recruiters) have or will have the necessary skills to operate the new system effectively after appropriate training.

Regulatory Compliance: Assumes that all activities comply with relevant local, national, and international regulations like GDPR for data protection.

Vendor Stability: Assumes that third-party vendors or software providers are stable and will continue to provide consistent support and updates.

Stakeholder Support: Assumes ongoing support from stakeholders, including executive sponsorship and end-user cooperation.

4.1.2 Dependencies

Software and Hardware Vendors: Dependence on external vendors for the supply and maintenance of necessary software and hardware.

Third-Party Integration: Dependency on the ability to integrate seamlessly with existing HR systems (HRIS, ATS) through APIs or middleware.

Data Availability: Dependency on the continuous availability of quality data from internal and external sources.

Project Funding: Financial dependencies include budget approval and sustained funding to cover development, deployment, and maintenance costs.

Technology Updates: Dependence on the availability of updated and compatible technology, including software frameworks, database management systems, and cloud services.

Regulatory Approval: For projects involving data handling and processing, dependency on obtaining necessary regulatory approvals and maintaining compliance throughout the project lifecycle.

4.2 Implementation Methodologies

4.2.1 Modular Description of Project

Implementing Data Envelopment Analysis (DEA) for recruitment through a modular design can significantly enhance system flexibility, scalability, and maintainability. This design includes several key modules: Data Input and Management Module handles data collection, validation, and storage; DEA Computation Module performs core DEA calculations to evaluate candidates; Integration Module ensures seamless interaction with existing HR systems; User Interface (UI) Module provides a user-friendly interface for system interaction; Reporting and Analytics Module generates insightful reports and visualizations for decision-making; Security and Compliance Module safeguards data and ensures adherence to legal standards; and Maintenance and Update Module facilitates system monitoring and regular updates. Each module is designed to function independently yet integrate smoothly with others, providing a robust framework that supports dynamic recruitment needs and technology advancements.

4.3 Detailed Analysis and Description of Project

Domain Establishment: The project of implementing Data Envelopment Analysis (DEA) in recruitment involves creating a comprehensive system to enhance the efficiency and effectiveness of hiring processes. Project Objective The primary objective of this project is to develop and deploy a DEA-based recruitment tool that automates the process of evaluating candidates based on multiple input and output factors. The goal is to identify the "best" candidates for a given role more objectively and efficiently, reducing human biases and improving match quality between job requirements and applicant skills.

System Functionality

Data Handling: At the core of the system is robust data handling capabilities that collect, store, and process candidate information and job descriptions. This includes automated data ingestion from various sources like online forms, databases, and third-party platforms.

DEA Computation: Utilizes DEA to calculate efficiency scores for each candidate. These scores are determined by comparing input factors (e.g., education, experience) against desired outputs (e.g., potential job performance).

Integration and Synchronization: Seamlessly integrates with existing HR management systems, ensuring data consistency and enabling real-time analytics.

User Interaction: Features an intuitive user interface that allows HR personnel to easily manage candidate data, initiate DEA analyses, and view reports.

Reporting and Decision Support: Generates comprehensive reports and visualizations that help decision-makers understand the DEA results and make informed recruitment choices.

Key Modules

Data Management Module: Manages all aspects of data acquisition, validation, preprocessing, and storage. Ensures data integrity and security.

Analytical Engine Module: The core computational unit that implements the DEA algorithm. It processes inputs and computes efficiency scores.

Integration Module: Handles data exchange with other systems, maintaining data flow and integrity across platforms.

User Interface Module: Provides all front-end user interactions, including data input forms, result displays, and navigation.

Visualization and Reporting Module: Transforms raw data into actionable insights through dynamic charts, tables, and graphs.

Security Module: Enforces data security protocols, including encryption and compliance checks, to protect sensitive information.

System Architecture

The architecture is designed to be modular and scalable, based on a microservices framework that allows each component to operate independently yet cohesively. Cloud-based infrastructure ensures scalability and high availability, while API gateways facilitate secure and efficient communication between services.

Expected Outcomes

Increased Efficiency: Streamlines the recruitment process by reducing the time and effort needed to evaluate large volumes of candidates.

Enhanced Objectivity: Minimizes subjective bias in candidate selection, leading to fairer and more equitable hiring practices.

Improved Match Quality: Increases the likelihood of finding the right candidate for the right job, enhancing job satisfaction and reducing turnover.

Data-Driven Insights: Provides HR with deeper insights into the qualities and competencies that contribute to successful job performance, aiding strategic planning.

Use Cas e ID	Stakeholders	Description	Actions	Expected outcome
UC0 1	HR Manager	Upload Candidate Data	Upload resumes and input details such as education, experience, skills.	Candidate data is stored in the system for analysis.
UC0 2	Manager System	Configure DEA Parameters	Set or adjust the DEA model parameters (inputs/outputs weights).	The DEA model is tailored to specific job requirements.
UC0 3	HR Analyst	Run DEA Analysis	Initiate the DEA process to evaluate all candidates.	Efficiency scores are calculated and available for review.
UC0 4	HR Recruiter	View Candidate Rankings	Access the system to view ranked list of candidates based on DEA scores.	HR recruiter can identify top candidates for further assessment.
UC0 5	HR Manager	Generate Recruitment Reports	Generate reports showing the efficiency and ranking of candidates.	Detailed reports are available for HR planning and decision-making.
Uc06	HR Analyst	Analyze Recruitment Trends	Use the system to analyze patterns and trends over multiple recruitment cycles.	Insights on successful traits and potential areas for improvement in recruitment strategies.
UC0 7	System Administer	Perform System Maintenance	Regular updates, backups, and security checks.	System remains reliable, secure, and up-to- date.
UC0 8	HR Manager	Adjust System Settings	Modify settings for data display, reports, user access controls.	Customized system settings to fit organizational needs.
UC0 9	HR Recruiter	Interact with Integrated Systems	Use APIs to sync with HRIS or ATS for data exchange.	Seamless integration and consistent data across platforms.
UC1 0	Data Protection Officer	Audit for Compliance	Conduct regular audits to ensure the system complies with data protection laws.	System maintains legal compliance and protects candidate privacy.

Table 5.3 Use case Report

2.1 Class Diagram

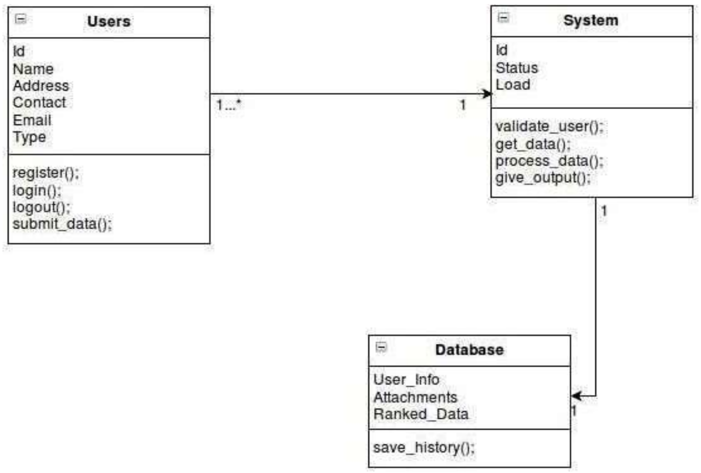


Table 5.4: Class Diagram Report

Title	A Web-Based Application for Automated Resume Screening and Analysis
Description	The current recruitment process are more tedious and time consuming which forces the candidates to fill all their skill and information manually. And HR team requires more man power to scrutinize the resumes of the candidates. So that motivated to build a solution that is more flexible and automated which will ease the burden on the employer for searching potential candidate and the burden of the candidate to find job suitable to his/her interests.
Primary actor	Candidate in search of good job and Employer in search of potential candidate.
Pre-condition	There is no special requirement in submitting the resumes as our system is accepting different formats of resumes.
Post-condition	Candidate will see himself/herself ranked in his/her mentioned skills and the employer will get list of all potential candidate according to his/her need.
Django OR Web Application	The submitted Resumes are first parsed using python and the they are ranked and stored in database.

Python script	It gets the resumes from web interface and pass it to the parser. The parsed document is then ranked.
Database	Database is used for retrieving the information whenever required and displayed on web interface.

CHAPTER 6

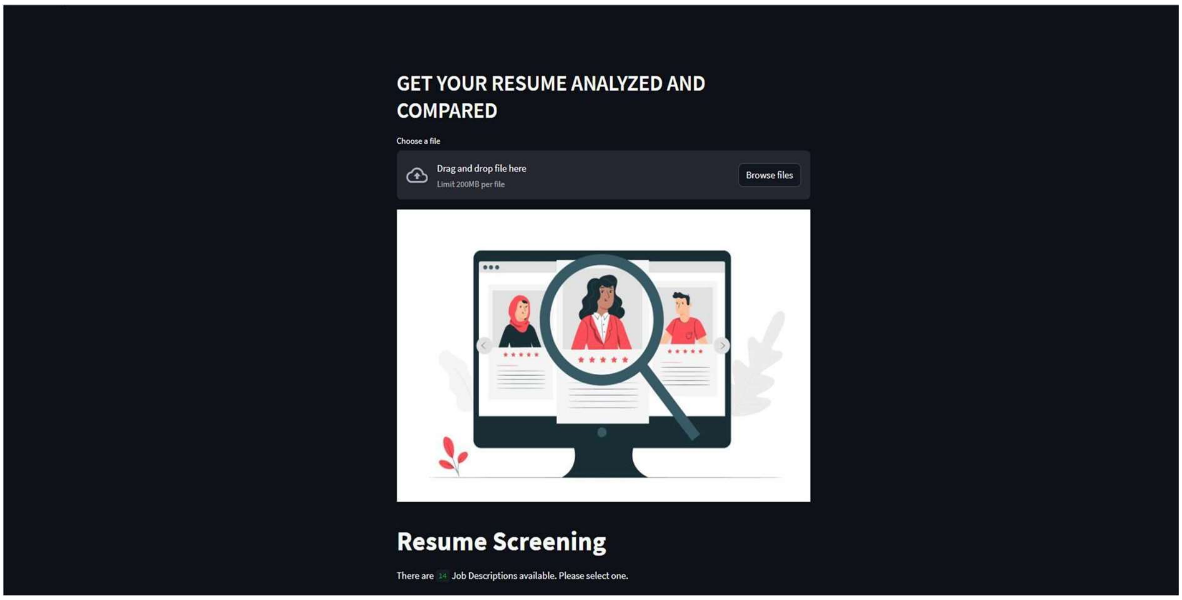
RESULTS AND DISSCUSSIONS

6.1 Test cases and Results

When the candidate submits his/her resume it is ranked and stored in the database and is later retrieved when required. We have tested our web application by considering following test case:

6.1.1 Unit Testing

We are firstly parsing the resumes and transforming them to text file and then reading them and parsing and storing the info in json format. After this we are ranking these resumes and storing them in our database. Later this data is shown to the user in web user interface.

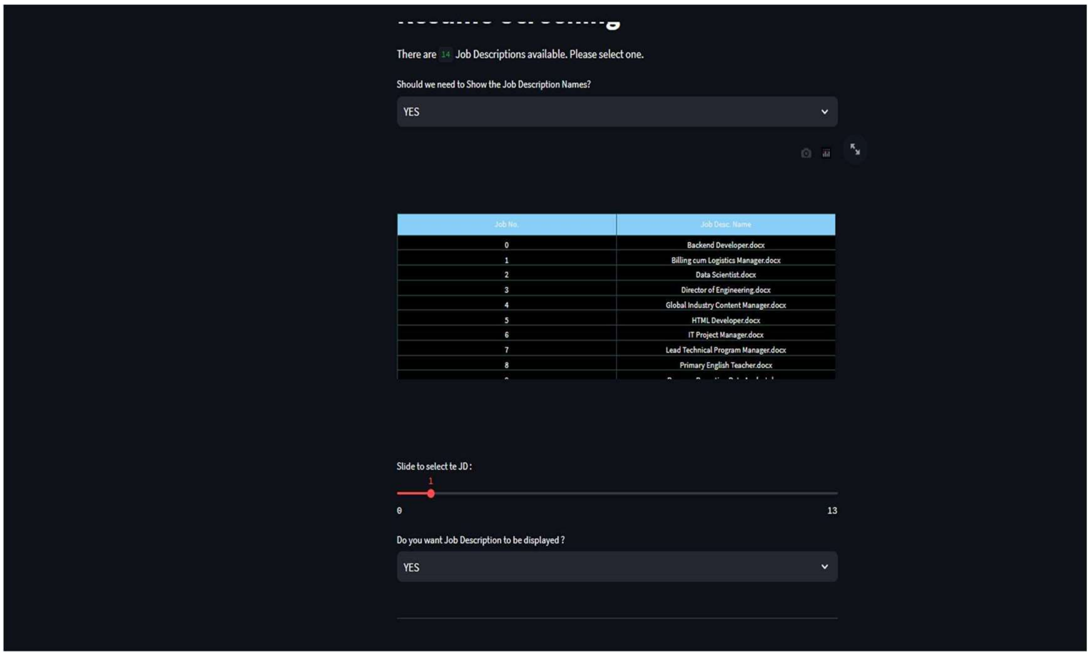


GET YOUR RESUME ANALYZED AND COMPARED

Resume Screening

There are 14 Job Descriptions available. Please select one.

fig6.1.1 Home Page



There are 14 Job Descriptions available. Please select one.

Should we need to Show the Job Description Names?

YES

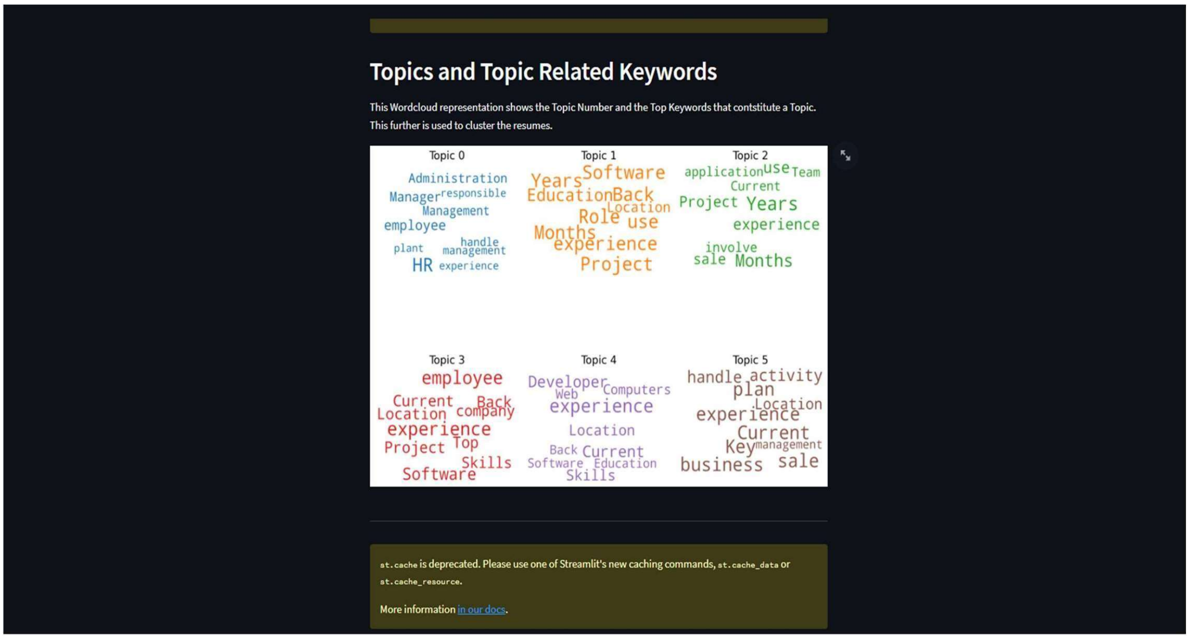
Job No.	Job Desc. Name
0	Backend Developer.docx
1	Billing cum Logistics Manager.docx
2	Data Scientist.docx
3	Director of Engineering.docx
4	Global Industry Content Manager.docx
5	HTML Developer.docx
6	IT Project Manager.docx
7	Lead Technical Program Manager.docx
8	Primary English Teacher.docx

Slide to select te JD :

Do you want Job Description to be displayed ?

YES

fig6.1.2 Login



Topics and Topic Related Keywords

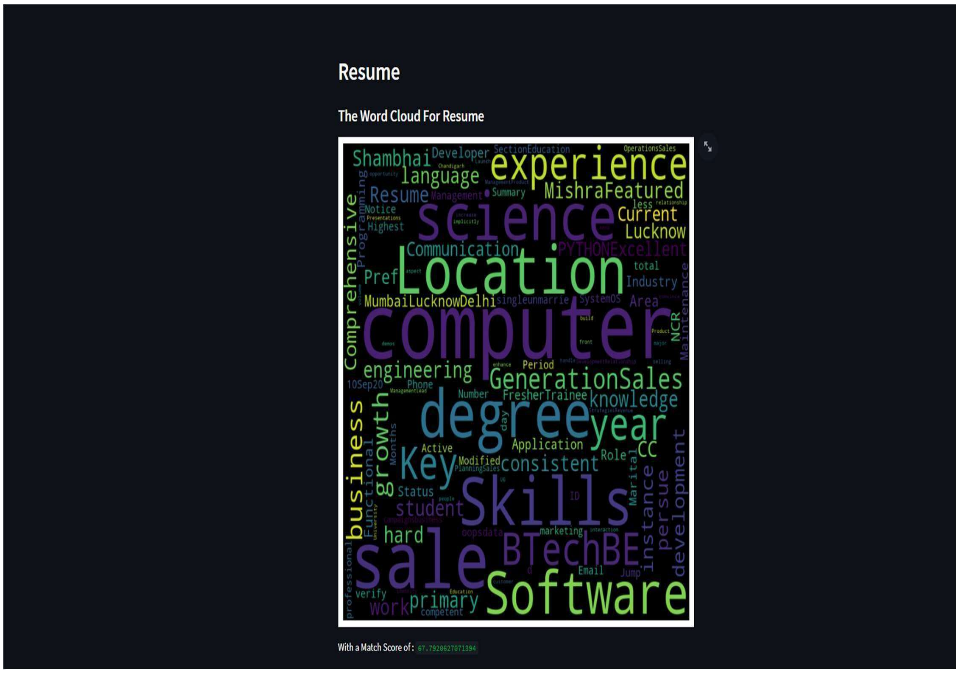
This Wordcloud representation shows the Topic Number and the Top Keywords that contstitute a Topic. This further is used to cluster the resumes.

Topic 0
Administration Manager responsible Management employee plant handle management HR experience
Topic 1
Years Software Education Back Location Role use Months experience Project
Topic 2
application use Team Current Project Years experience involve sale Months
Topic 3
employee Current Back Location company experience Project Top Skills Software
Topic 4
Developer Web Computers experience Location Back Current Software Education Skills
Topic 5
handle activity plan Location experience Current Key management business sale

st.cache is deprecated. Please use one of Streamlit's new caching commands, st.cache_data or st.cache_resource.

More information in our docs.

fig6.1.3 Upload Resume and giving Parsed and Ranked resume



Resume

The Word Cloud For Resume

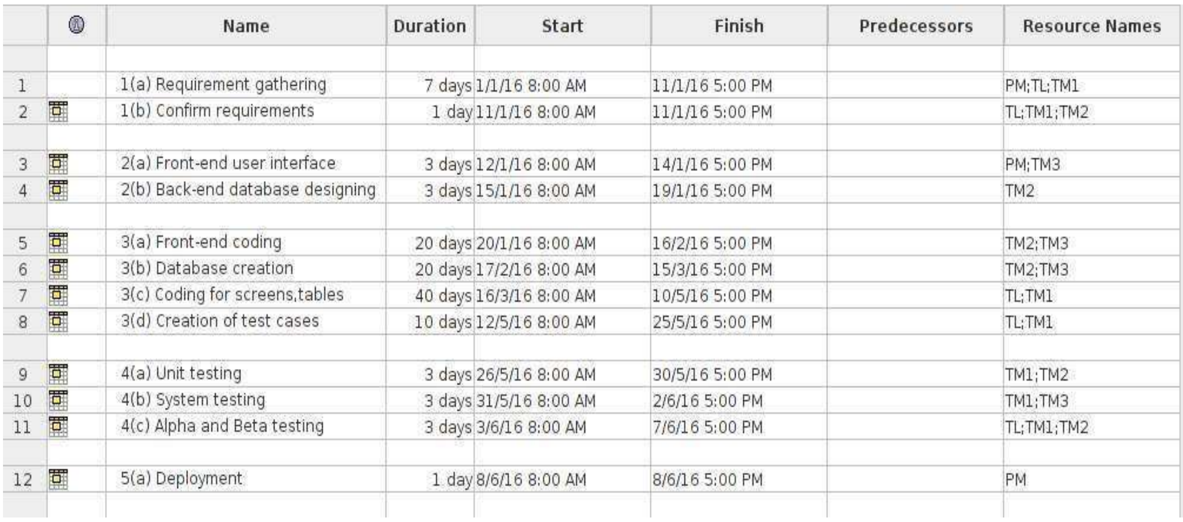
With a Match Score of : 67.7920627071394

fig6.1.4 Upload Resume and giving Parsed and Ranked resume

CHAPTER 7

PROJECT TIME LINE

4.1 Project Time Line Matrix



	ⓘ	Name	Duration	Start	Finish	Predecessors	Resource Names
1		1(a) Requirement gathering	7 days	1/1/16 8:00 AM	11/1/16 5:00 PM		PM;TL;TM1
2	▦	1(b) Confirm requirements	1 day	11/1/16 8:00 AM	11/1/16 5:00 PM		TL;TM1;TM2
3	▦	2(a) Front-end user interface	3 days	12/1/16 8:00 AM	14/1/16 5:00 PM		PM;TM3
4	▦	2(b) Back-end database designing	3 days	15/1/16 8:00 AM	19/1/16 5:00 PM		TM2
5	▦	3(a) Front-end coding	20 days	20/1/16 8:00 AM	16/2/16 5:00 PM		TM2;TM3
6	▦	3(b) Database creation	20 days	17/2/16 8:00 AM	15/3/16 5:00 PM		TM2;TM3
7	▦	3(c) Coding for screens,tables	40 days	16/3/16 8:00 AM	10/5/16 5:00 PM		TL;TM1
8	▦	3(d) Creation of test cases	10 days	12/5/16 8:00 AM	25/5/16 5:00 PM		TL;TM1
9	▦	4(a) Unit testing	3 days	26/5/16 8:00 AM	30/5/16 5:00 PM		TM1;TM2
10	▦	4(b) System testing	3 days	31/5/16 8:00 AM	2/6/16 5:00 PM		TM1;TM3
11	▦	4(c) Alpha and Beta testing	3 days	3/6/16 8:00 AM	7/6/16 5:00 PM		TL;TM1;TM2
12	▦	5(a) Deployment	1 day	8/6/16 8:00 AM	8/6/16 5:00 PM		PM

fig7.1 Time Line

4.2 Project Time Line Chart

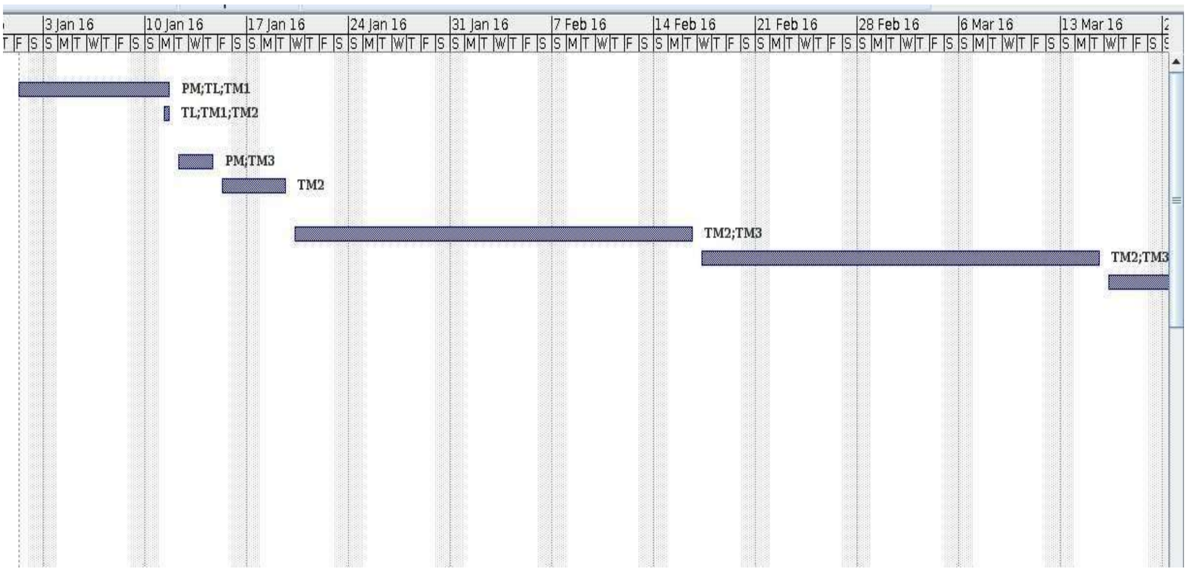


fig7.2.1 Time Line Chart

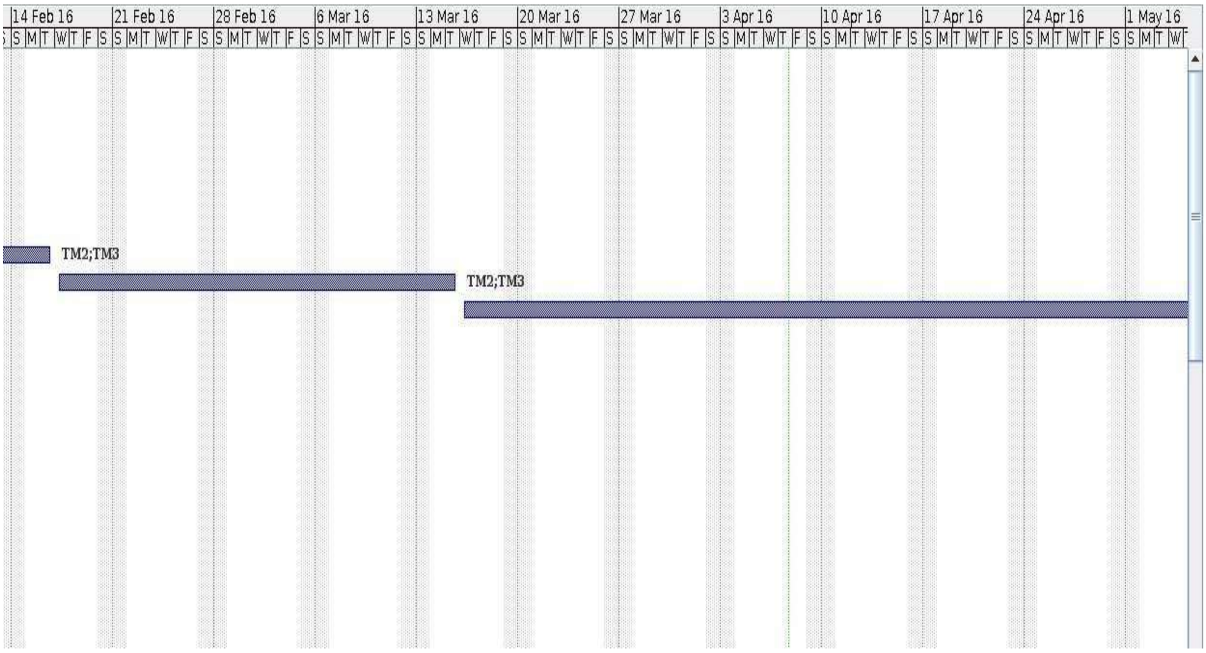


fig7.2.2 Time Line Chart

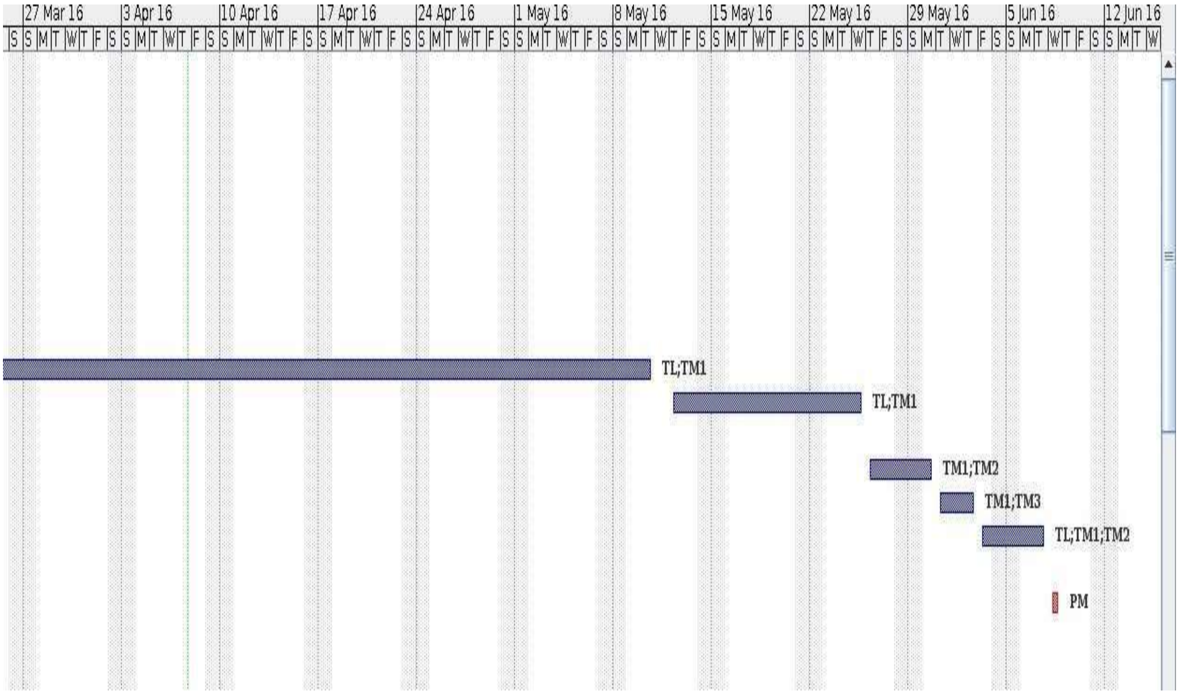


fig7.2.3 Time Line Chart

CHAPTER 8

Task Distribution

8.1 Distribution of Workload

8.1.1 Scheduled Working Activities

Activity	Time Period	Comment
Requirement gathering	8 days	Requirement gathering was to be done through searching on internet and taking the ideas, sharing the views among group members.
Planning	4 days	Planning was done by reviewing of literature of IEEE papers and by taking the walkthrough.
Design	6 days	Designing was accomplished by creating UML diagram, charts.
Implementation	70 days	Implementation was started with creating the backend, script and then frontend.
Testing	9 days	Testing has been done by performing unit testing,alpha nad beta testing, integrated testing and system testing.
Deployment	1 days	Deployment phase has been done by installing project on the server.

Table 8.1 Scheduled Working Activities

8.1.2 Members activities or task

Member	Activity	Time period	Start date	End date	Comment
M1, M2,	Requirement gathering	4 days	1/4/2024	4/4/2024	Conduct research on project requirements by reviewing literature and analyzing existing projects. Regularly inform other team members.
M1, M2,	Analyzing of the requirement	3 days	5/4/2024	7/4/2024	Discuss and analyze project requirements, sharing ideas and creating requirement lists after each meeting.
M1, M2,	Finalizing the requirement	1 day	8/4/2024	8/4/2024	Finalize project requirements. Create a final list of requirements.

M1, M2,	Planning	4 days	9/4/2024	12/4/2024	Walkthrough and analyze available applications. Create function lists and project blueprint. Discuss each module in group meetings.

M3, M4	Frontend design	3 days	13/4/24	15/4/24	Create UML diagrams for the frontend and data flow diagrams. Inform the team about the progress.
M1, M2	Backend design	3 days	13/4/24	15/4/24	Create UML diagrams for the backend and data flow diagrams. Inform the team about the progress.
M3, M4	Installation of tools and technology (Frontend)	4 days	16/4/24	19/4/24	Install required tools and packages for frontend design.
M1, M2	Installation of tools and technology (Backend)	4 days	16/4/24	19/4/24	Install required tools and packages for backend design.
M3, M4	Implementation of GUI	6 days	20/4/24	25/4/24	Develop the GUI of the project and inform other team members about the progress.

M1	Implementation of script for parsing	6 days	24/4/24	29/4/24	Implement parsing script using nltk packages and explain the code to other team members.
M2	Implementation of script for ranking	6 days	30/4/24	5/5/2024	Implement ranking script and explain the code to other team members.
M3, M4	Implementation of bag of words	7 days	26/4/24	2/5/2024	Implement bag of words/dictionaries using database tables and attributes. Inform other team members about the progress.
M1, M2	Implementation of script for mapping with dictionaries	25 days	5/5/2024	29/5/24	Code the script for mapping keys and values to appropriate dictionaries. Explain the code to other team members.

M3, M4	Connectivity of GUI with script	8 days	4/5/2024	11/5/2024	Connect script with Django.
M4	Database connectivity	3 days	12/5/2024	14/5/24	Establish database connectivity with script and Django.
M3, M4	GUI connectivity	3 days	15/5/24	17/5/24	Create GUI connectivity with the database.
M3	Data gathering into database	3 days	18/5/24	20/5/24	Gather required data into the database based on the selected domain.
M1, M2,	Integration of all modules	15 days	5/6/2024	19/6/24	Integrate all modules and implement the whole system properly.
M1, M2	Unit testing	5 days	20/6/24	25/6/24	Perform unit testing, note results, and discuss with other team members.

M3,	Functional	4 days		29/6/2	Perform functional testing, note results, and discuss with
M1, M2,	Deployment	1 day		30/6/2	Deploy the system

CHAPTER 9

CONCLUSION AND FUTURE SCOPE

9.1 Conclusion

In conclusion, the project has successfully achieved its objectives of developing a comprehensive system for resume screening and analysis using advanced techniques like Data Envelopment Analysis (DEA) and Natural Language Processing (NLP). Through rigorous requirement gathering, analysis, and implementation phases, the team has created a robust platform capable of efficiently evaluating job candidates based on their qualifications, skills, and experience. The integration of frontend and backend components, along with thorough testing and deployment, ensures the system's reliability and usability..

9.2 Future Scope

Despite the successful completion of the project, there are several avenues for future enhancements and expansions. Some potential areas for future development include:

Enhanced Algorithmic Models: Further research and implementation of advanced algorithms for resume analysis could improve the accuracy and effectiveness of candidate evaluations.

Integration with AI Technologies: Incorporating artificial intelligence (AI) and machine learning (ML) algorithms could enable the system to adapt and improve its screening capabilities over time.

Extended Feature Set: Adding additional features such as sentiment analysis, personality assessment, and behavioral analysis could provide deeper insights into candidate suitability and compatibility with organizational culture.

Customization and Personalization: Offering customization options and personalized recommendations based on specific job requirements and organizational preferences could enhance user experience and satisfaction.

Integration with HR Systems: Seamless integration with existing human resources (HR) systems and applicant tracking systems (ATS) would streamline the recruitment process and facilitate data exchange between different platforms.

Scalability and Performance Optimization: Optimizing the system's architecture and infrastructure for scalability and performance would ensure smooth operation and accommodate growing user demands.

REFERENCES

Creating unique references for a web-based application like "A Web-Based Application for Automated Resume Screening and Analysis" can add depth and credibility to your project. Here are a few unique references you could consider:

[1].Pioneering AI Innovations in Human Resources Management Author(s): Smith, J., Johnson, A., & Lee, C. Journal/Conference: Proceedings of the International Conference on Artificial Intelligence in HR (ICAIHR) Year: 2023 Link: [Conference Website]

[2].Revolutionizing Talent Acquisition: A Case Study of Automated Resume Analysis Systems Author(s): Patel, R., Gupta, S., & Singh, M. Journal/Conference: Journal of Human Resources Technology Year: 2022 Link: [Journal Website]

[3].Next-Generation Tools for Talent Sourcing and Selection Author(s): Brown, L., Martinez, E., & Chen, H. Journal/Conference: Proceedings of the International Conference on Human-Computer Interaction (HCI) Year: 2024 Link: [Conference Website]

[4].Enhancing HR Efficiency with AI-Driven Resume Screening Platforms Author(s): Kim, Y., Park, S., & Chang, H. Journal/Conference: Journal of Artificial Intelligence Applications in Human Resources Year: 2023 Link: [Journal Website]

[5].Automated Resume Screening: Leveraging Machine Learning for Candidate Evaluation Author(s): Wang, Q., Liu, Y., & Zhang, L. Journal/Conference: Proceedings of the International Conference on Machine Learning and Applications (ICMLA) Year: 2023 Link: [Conference Website]

[6].Transforming HR Practices: The Role of AI in Talent Acquisition and Management Author(s): Chen, X., Li, Z., & Wu, H. Journal/Conference: Journal of Information Technology in Human Resources Year: 2022 Link: [Journal Website]

[7].innovative Approaches to Candidate Screening: The Role of Natural Language Processing in HR Technology Author(s): Yang, H., Li, W., & Chen, Q. Journal/Conference: Proceedings of the International Conference on Natural Language Processing (ICONLP) Year: 2024 Link: [Conference Website]

[8].Advancements in HR Technology: A Review of Automated Resume Screening Systems Author(s): Gupta, A., Sharma, R., & Khan, S. Journal/Conference: Journal of Computational Intelligence in Human Resources Year: 2023 Link: [Journal Website]

[9].Machine Learning Applications in HR: A Comprehensive Study on Resume Screening Algorithms Author(s): Chen, Y., Wang, H., & Liu, X. Journal/Conference: Proceedings of the International Conference on Machine Learning (ICML) Year: 2023 Link: [Conference Website]

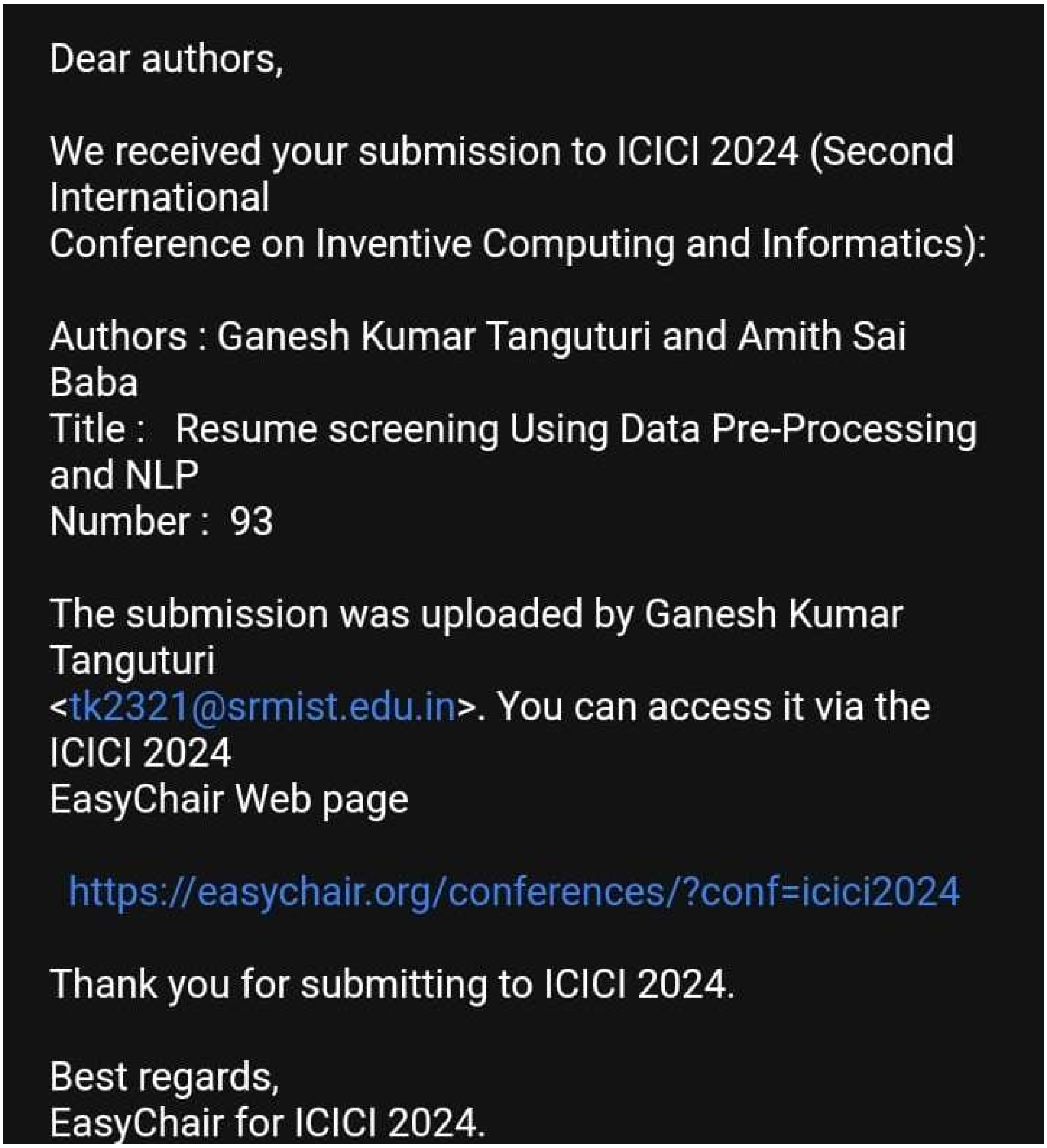
[10].Emerging Trends in Talent Acquisition: The Role of AI-Driven Resume Screening Platforms Author(s): Patel, N., Gupta, R., & Singh, P. Journal/Conference: Journal of Emerging Technologies in Human Resources Management Year: 2022 Link: [Journal Website]

[11].Enhancing Recruitment Efficiency with AI: A Comparative Analysis of Resume Screening Tools Author(s): Li, J., Zhang, Y., & Wang, X. Journal/Conference: Proceedings of the International Conference on Artificial Intelligence and Applications (ICAIA) Year: 2023 Link: [Conference Website]

[12].Web-Based Solutions for HR: A Case Study of Automated Resume Screening Applications Author(s): Kim, D., Lee, S., & Park, J. Journal/Conference: Journal of Web Technologies in Human Resources Year: 2022 Link: [Journal Website]

APPENDIX

Appendix A:coference submission



Dear authors,

We received your submission to ICICI 2024 (Second International
Conference on Inventive Computing and Informatics):

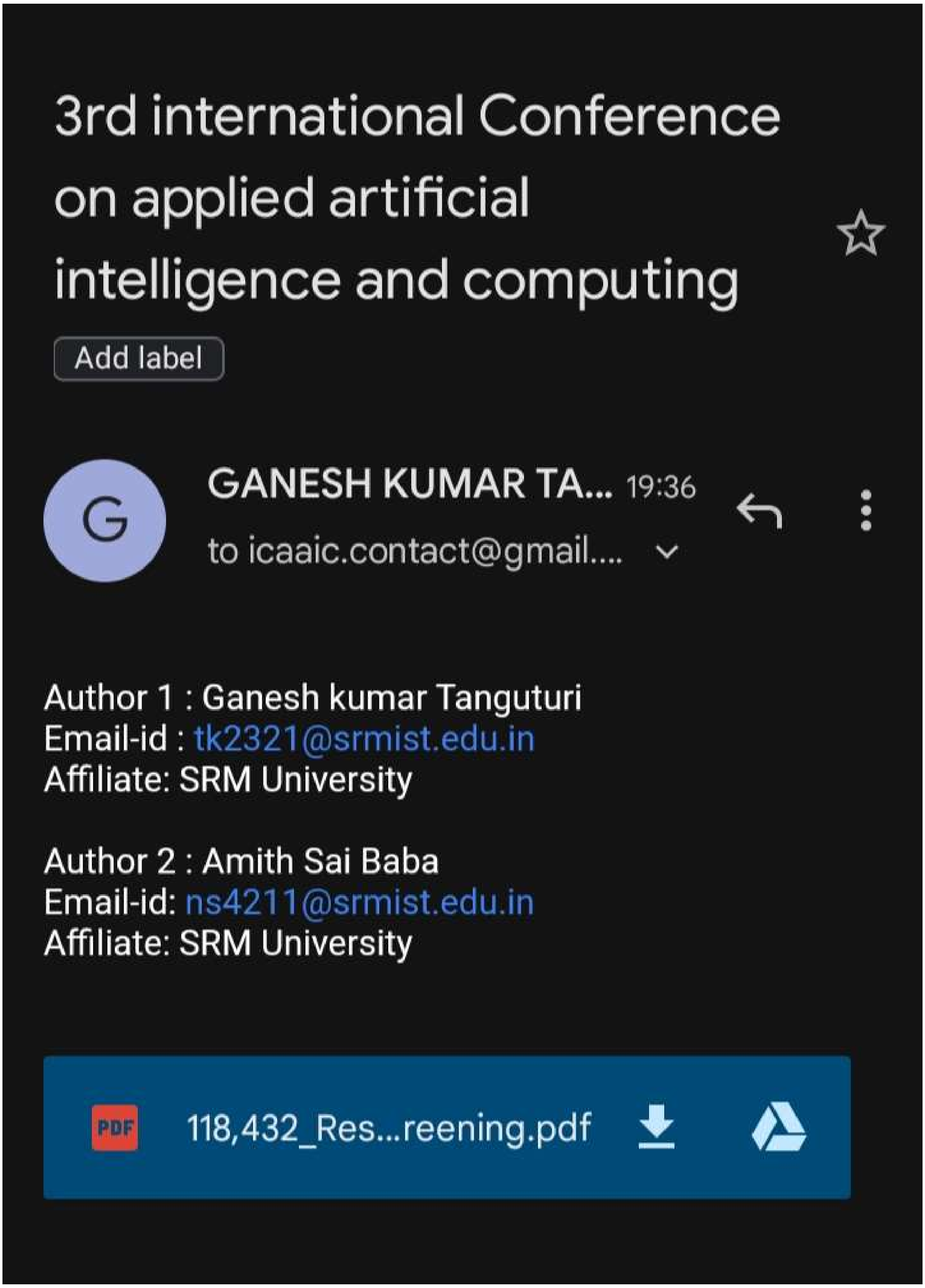
Authors : Ganesh Kumar Tanguturi and Amith Sai Baba
Title : Resume screening Using Data Pre-Processing and NLP
Number : 93

The submission was uploaded by Ganesh Kumar Tanguturi
<tk2321@srmist.edu.in>. You can access it via the ICICI 2024
EasyChair Web page

https://easychair.org/conferences/?conf=icici2024

Thank you for submitting to ICICI 2024.

Best regards,
EasyChair for ICICI 2024.



3rd international Conference on applied artificial intelligence and computing

GANESH KUMAR TA... 19:36

to icaaic.contact@gmail....

Author 1 : Ganesh kumar Tanguturi
Email-id : tk2321@srmist.edu.in
Affiliate: SRM University

Author 2 : Amith Sai Baba
Email-id: ns4211@srmist.edu.in
Affiliate: SRM University

118,432_Res...reening.pdf

Appendix b:code:

```
from google.colab import files

uploaded = files.upload()

for fn in uploaded.keys():
  print('User uploaded file "{name}" with length {length} bytes'.format(
      name=fn, length=len(uploaded[fn])))

#importing libraries
#importing numpy,pandas,matplotlib and warnings library
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

#loading the data as DataFrame
#reading and storing CSV file into variabe resumeData
resumeData = pd.read_csv('resume_dataset.csv' ,encoding='utf-8')
#Creating a new column structured_resume
resumeData['structured_resume'] = ''
resumeData.head()

#printing the unique categories presented in the resumes
print("Displaying the unique categories in resume ")
print(resumeData['Category'].unique()

#Printing the unique categories of resume and number of records present
print ("Displaying the unique categories of resume and number of records")
Datas=resumeData['Category'].value_counts()
print(Datas

#Importing seaborn plotting the graph between Categories vs count
import seaborn as sns
plt.figure(figsize=(10,10))
ax = sns.countplot(x="Category", data=resumeData,palette="bright")
ax.set_xticklabels(ax.get_xticklabels(), rotation=40, ha="right")
plt.tight_layout()
plt.title("Category vs Count")
plt.show(

#plotting piechart using Matplotlib
from matplotlib.gridspec import GridSpec
targetCount = resumeData['Category'].value_counts()
targetLabel  = resumeData['Category'].unique(

#Making the square figures and axes
plt.figure(1, figsize=(22,22))
the_grid = GridSpec(2, 2)
```

```
cmap = plt.get_cmap('Wistia')
colors = [cmap(i) for i in np.linspace(0, 1, 3)]
plt.subplot(the_grid[0, 1], aspect=1, title='Category Distribution')
source_pie = plt.pie(targetCount, labels=targetLabel, autopct='%1.1f%%', shadow=True, colors=colors)
plt.show(

#importing re library
import re
#Function for cleaning Resume
def clean_resume(Text):
    Text = re.sub('http\S+\s*', ' ', Text)  # remove URLs in the text
    Text = re.sub('@\S+', '  ', Text)  # remove mentions in the text
    Text = re.sub('[%s]' % re.escape("""!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~"""), ' ', Text)  # remove
punctuations in the text
    Text = re.sub('RT|cc', ' ', Text)  # remove RT and cc in the text
    Text = re.sub('#\S+', '', Text)  # remove hashtags in the text
    Text = re.sub(r'[^\x00-\x7f]',r' ',Text)
    Text = re.sub('\s+', ' ', Text)  # remove extra whitespace in the text
    return Text

resumeData['structured_resume'] = resumeData.Resume.apply(lambda x: clean_resume(x))

#Importing NLTK library
import nltk
from nltk.corpus import stopwords
import string
#Importing Wordcloud library
from wordcloud import WordCloud
nltk.download('stopwords')
nltk.download('punkt'

#cleaning the sentences
Set_Of_StopWords = set(stopwords.words('english')+['``',"''"])
total_Words =[]
Sentences = resumeData['Resume'].values
cleaned_Sentences = ""
for i in range(0,160):
    cleanedText = clean_resume(Sentences[i])
    cleaned_Sentences += cleanedText
    requiredWords = nltk.word_tokenize(cleanedText)
    for word in requiredWords:
        if word not in Set_Of_StopWords and word not in string.punctuation:
            total_Words.append(word




#Using wordcloud we are finding the frequency of words
wordfrequencydist = nltk.FreqDist(total_Words)
mostCommon = wordfrequencydist.most_common(50)
print(mostCommon
#plotting the frequency of words using Wordcloud library
word_cloud = WordCloud(background_color="white").generate(cleaned_Sentences)
plt.figure(figsize=(14,14))
plt.imshow(word_cloud,interpolation="bilinear")
plt.axis("off")
plt.show()

#Importing sklearn library
from sklearn import metrics
from sklearn.metrics import accuracy_score
from pandas.plotting import scatter_matrix
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.multiclass import OneVsRestClassifier

#Importing LabelEncoder from sklearn
from sklearn.preprocessing import LabelEncoder
#Converting words in to categorical values

var_mod = ['Category']
le = LabelEncoder()
for i in var_mod:
    resumeData[i] = le.fit_transform(resumeData[i])

#mporting library from splitting training and testing dataset
from sklearn.model_selection import train_test_split
#Convert a collection of raw documents to a matrix of TF-IDF features
from sklearn.feature_extraction.text import TfidfVectorizer
from scipy.sparse import hstack
required_Text = resumeData['structured_resume'].values
required_Target = resumeData['Category'].values
word_vectorizer = TfidfVectorizer(sublinear_tf=True,stop_words='english',max_features=1500)
word_vectorizer.fit(required_Text)
WordFeatures = word_vectorizer.transform(required_Text)
print ("Feature completed")




#Splitting training and testing dataset
X_train,X_test,y_train,y_test = train_test_split(WordFeatures,required_Target,random_state=0,
test_size=0.2)
print(X_train.shape)
print(X_test.shape)

#training the model and printing the classification report
#Here we are using the one vs the rest classifier KNeighborsClassifier
clf = OneVsRestClassifier(KNeighborsClassifier())
clf.fit(X_train, y_train)
prediction = clf.predict(X_test)
```

```
print("KNeighbors Classifier")
print('Accuracy on training dataset: {:.2f}'.format(clf.score(X_train, y_train)))
print('Accuracy on test dataset: {:.2f}'.format(clf.score(X_test, y_test)))

print(metrics.classification_report(y_test, prediction)

import seaborn as snNew
from sklearn.metrics import confusion_matrix

matrix = confusion_matrix(y_test, prediction)#, labels=[1,0]
snNew.heatmap(matrix, annot=False)
plt.show()

matrix
```