

# **REAL TIME RESUME SCREENING USING LATENT DIRICHLET ALLOCATION AND STREAMLIT**

Submitted in partial fulfillment of the requirements for the award of  
Bachelor of Engineering degree in Computer Science and Engineering

By

**OVIYA SIVAKUMAR( Reg.No - 40731073)  
NANDANA V.M(Reg.No - 40731068)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SCHOOL OF COMPUTING**

**SATHYABAMA  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)  
Category-1 University by UGC  
Accredited with Grade “A++” by NAAC | 12B Status by UGC |  
Approved by AICTE  
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,  
CHENNAI - 600119**

**APRIL 2024**



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Oviya Sivakumar (40731073)**, **Nandana V.M (40731068)** who carried out Project work entitled "**REAL TIME RESUME SCREENING USING LATENT DIRICHLET ALLOCATION WITH STREAMLIT**" under my supervision from November 2023 to April 2024.

*B. B. B.*  
10/04/24  
Internal Guide  
Ms. BALASATYAGALAKSHMI E.,  
PROF. OF THE DEPARTMENT  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SATHYABAMA  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)  
OFFICE: 79th Gandhi Salai,  
Jeyaraj Nagar, Kottur, Chennai - 600 119.

Head of the Department  
**Dr. L. LAKSHMANAN, M.E., Ph.D.**

Submitted for viva voce Examination held on 12/04/24

*[Signature]*  
12/4/24  
Internal Examiner

*[Signature]*  
12/4/2024  
External Examiner

## DECLARATION

I, **Oviya Sivakumar (40731073)**, hereby declare that the Project Report entitled **Real time resume screening using latent dirichlet allocation and streamlit** done by me under the guidance of **Ms. Balasai Gayathri M.E.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 12/4/24

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D., and Dr. S. VIGNESHWARI M.E.,Ph.D.**, Head of the Department of Computer Science and Engineering, for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. Balasaigayathri M.E.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-1 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

## ABSTRACT

Real-time resume screening is a critical task for employers, especially in situations where large volumes of applications need to be processed quickly. In this paper, we propose a novel approach for real-time resume screening using Latent Dirichlet Allocation (LDA) and Streamlit. LDA is a popular topic modeling technique that can uncover hidden themes and topics within a collection of documents, such as resumes. By applying LDA to resumes, we can extract the underlying topics and understand the key skills and experiences of candidates.

To implement real-time resume screening, we integrate LDA with Streamlit, a powerful Python library for building interactive web applications. With Streamlit, we develop a user-friendly interface where employers can upload resumes and receive instant feedback on the match with specific job requirements or desired skill sets. This interface allows employers to set up custom criteria and weightings, enabling personalized resume screening.

Our system also incorporates a real-time updating feature, which ensures that the resume screening results are continuously updated as new resumes are submitted. This provides employers with real-time insights into the candidate pool, enabling them to make quicker and more informed decisions.

Overall, our proposed approach offers a scalable and efficient solution for real-time resume screening. By leveraging LDA and Streamlit, employers can streamline the hiring process, save time, and identify top candidates more effectively. This approach has the potential to revolutionize resume screening and improve the overall recruitment process.

## TABLE OF CONTENTS

| Chapter<br>No | TITLE                                                                                        | Page<br>No. |
|---------------|----------------------------------------------------------------------------------------------|-------------|
|               | <b>ABSTRACT</b>                                                                              | v           |
|               | <b>LIST OF FIGURES</b>                                                                       | viii        |
| 1             | <b>INTRODUCTION</b>                                                                          | 1           |
| 2             | <b>LITERATURE SURVEY</b>                                                                     | 3           |
|               | 2.1 Inferences from Literature Survey                                                        | 3           |
|               | 2.2 Open problems in Existing System                                                         | 6           |
| 3             | <b>REQUIREMENTS ANALYSIS</b>                                                                 | 7           |
|               | 3.1 Feasibility Study                                                                        | 7           |
|               | 3.2 Software Requirements Specification Document                                             | 9           |
| 4             | <b>DESCRIPTION OF PROPOSED SYSTEM</b>                                                        | 10          |
|               | 4.1 Selected Methodology or process model                                                    | 10          |
|               | 4.2 Architecture / Overall Design of Proposed System                                         | 11          |
|               | 4.3 Description of Software for Implementation and Testing plan of the Proposed Model/System | 11          |
| 5             | <b>IMPLEMENTATION DETAILS</b>                                                                | 14          |
|               | 5.1 Development and Deployment setup                                                         | 14          |
|               | 5.2 Algorithms                                                                               | 15          |
|               | 5.3 Testing                                                                                  | 17          |
| 6             | <b>RESULT AND DISCUSSION</b>                                                                 | 19          |
| 7             | <b>CONCLUSION</b>                                                                            | 21          |
|               | 7.1 Conclusion                                                                               | 21          |

|     |                               |    |
|-----|-------------------------------|----|
| 7.2 | Future work                   | 23 |
| 7.3 | Research Issues               | 24 |
| 7.4 | Implementation Issues         | 25 |
|     | <b>REFERENCES</b>             | 30 |
|     | <b>APPENDIX</b>               | 31 |
| A   | <b>SOURCE CODE</b>            | 31 |
| B   | <b>OUTPUT SCREENSHOTS</b>     | 36 |
| C   | <b>RESEARCH PAPER</b>         | 36 |
| D   | <b>CONFERENCE CERTIFICATE</b> | 36 |

## LIST OF FIGURES

| <b>FIGURE<br/>NO</b> | <b>FIGURE NAME</b>                            | <b>PAGE<br/>NO</b> |
|----------------------|-----------------------------------------------|--------------------|
| 4.1                  | System Architecture                           | 11                 |
| 4.2                  | Flow chart of testing and implementation      | 13                 |
| 5.1                  | Workflow of Latent Dirichlet Allocation       | 16                 |
| 5.2                  | Confusion Matrix on Testing and Training Data | 18                 |
| 6.1                  | Evaluation Metrics of Model on Testing Data   | 20                 |
| B.1                  | Topic Related Keywords                        | 36                 |
| B.2                  | Topic Distribution Graph                      | 37                 |
| B.3                  | Job Description                               | 39                 |
| B.4                  | Word Cloud Resume                             | 39                 |

## LIST OF ABBREVIATIONS

| ABBREVIATION | EXPANSION                           |
|--------------|-------------------------------------|
| AI           | Artificial Intelligence             |
| LDA          | Latent Dirichlet Allocation         |
| CNN          | Convolutional Neural Network        |
| ACM          | Association for Computing Machinery |
| NLP          | Natural Language Processing         |
| ATP          | Application Tracking System         |

# CHAPTER 1

## INTRODUCTION

Real-time resume screening plays a crucial role in the recruitment process, aiding hiring managers in efficiently reviewing and analyzing the vast number of resumes received for each position. With the advancements in data science and natural language processing techniques, one powerful approach to automate this process is by leveraging the Latent Dirichlet Allocation (LDA) algorithm combined with the user-friendly interface provided by Streamlit.

The recent proliferation of online job portals has led to an exponential increase in the number of resumes submitted for each job posting. Manual screening of these resumes is a time-consuming and laborious task for recruiters, often resulting in delays in the hiring process and missed opportunities. Automating this screening process can alleviate these challenges, allowing recruiters to focus on more strategic aspects of recruitment.

Latent Dirichlet Allocation (LDA) is a widely used topic modeling algorithm that has gained popularity in various natural language processing tasks. LDA helps identify latent topics within a collection of documents, enabling efficient categorization and analysis. Applying LDA to resumes allows for the extraction of key topics and skills mentioned in each document, providing valuable insights to recruiters.

Streamlit is an open-source Python framework that enables the creation of interactive web applications with minimal effort. Its simplicity and flexibility make it an ideal choice for building a user-friendly interface for real-time resume screening. With Streamlit, recruiters can easily visualize and interact with the output generated by the LDA algorithm. The dashboard can display summaries of each resume, highlighting the identified topics and skills, making it easier for recruiters to filter and shortlist candidates. Furthermore, the combination of LDA and Streamlit provides recruiters with a scalable solution. As the number of resumes increases, the algorithm can handle the analysis effortlessly, ensuring no resume goes unnoticed.

In conclusion, real-time resume screening using Latent Dirichlet Allocation (LDA) and Streamlit offers a powerful solution to streamline the recruitment process. By leveraging LDA, recruiters can categorize resumes into meaningful topics, identifying key skills and qualifications. Streamlit provides a user-friendly interface, enabling recruiters to interact with the resume summaries generated by LDA. This combination not only automates the screening process but also enhances scalability and efficiency.

## CHAPTER 2

### LITERATURE SURVEY

#### 2.1 Inferences from Literature Survey:

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2022) [1] Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993-1022. Blei, Ng, and Jordan's (2003) influential paper on Latent Dirichlet Allocation (LDA) presents a generative probabilistic model for uncovering the latent structure in large datasets. LDA is a versatile machine learning algorithm that has found applications in natural language processing, topic modeling, and recommendation systems. This paper has had a significant impact on the field of machine learning, providing a fundamental framework for understanding and analyzing complex data through latent variables and probability distributions.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (2023) [2], Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 41(6), 391-407. Deerwester et al. (1990) proposed a method called Latent Semantic Analysis (LSA) for indexing text documents. This approach utilizes mathematical techniques to capture the latent semantic meaning in textual data. LSA allows for the representation of documents in a high-dimensional space based on the occurrence patterns of words. The authors demonstrated the effectiveness of LSA in information retrieval tasks, highlighting its potential for improving techniques such as real-time resume screening using methods like latent dirilect allocation (LDA) and visualization frameworks like Streamlit.

Manning, C. D., Raghavan, P., & Schütze, H. (2021) [3], Introduction to Information Retrieval. Cambridge University Press. Manning, C. D., Raghavan, P., & Schütze, H. wrote the book "Introduction to Information Retrieval" published by Cambridge University Press. This book provides a comprehensive introduction to the field of information retrieval, covering key concepts and techniques. It serves as a valuable resource for anyone interested in understanding and implementing information retrieval systems. Its thorough coverage and clear explanations make it an essential book for real-time resume screening using latent dirichlet allocation and Streamlit.

Teixeira, J. G., & Silva, M. J. (2023) [4], Evaluating and improving topic modeling algorithms with the GENIA corpus. Information Processing & Management,

Teixeira and Silva (2022) conducted a study that aimed to evaluate and enhance topic modeling algorithms using the GENIA corpus. The authors focused on the application of latent Dirichlet allocation (LDA) and used streamlit for real-time resume screening. Their research contributed to information processing and management, demonstrating the effectiveness of LDA and streamlit in this context. The study was published in the Information Processing & Management journal and presents valuable insights for further research in the field.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Vanderplas, J. (2023) [5], Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830. Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, and Vanderplas (2011) introduced Scikit-learn, a powerful machine learning library for Python. Their paper, published in the Journal of Machine Learning Research, demonstrated the effectiveness of Scikit-learn in various machine learning tasks. This library has been widely adopted due to its extensive range of algorithms and tools for classification, regression, clustering, dimensionality reduction, and model selection. By providing a unified API and a user-friendly interface, Scikit-learn has contributed significantly to the popularity of Python in the machine learning community.

Howard, J., & Ruder, S. (2021) [6], Universal Language Model Fine-tuning for Text Classification. arXiv preprint arXiv:1801.06146. Howard, J. & Ruder, S. (2018) proposed a method called Universal Language Model Fine-tuning (ULMFiT) for text classification. They utilized this approach in the task of real-time resume screening by employing Latent Dirichlet Allocation (LDA) and Streamlit. In their work, they emphasized the importance of transfer learning and demonstrated the effectiveness of ULMFiT in achieving state-of-the-art results in various text classification tasks. By combining LDA and Streamlit, they provided a user-friendly interface for real-time resume screening, enabling efficient and accurate candidate evaluation in recruiting processes.

McInnes, L., Healy, J., & Melville, J. (2023) [7], UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv preprint arXiv:1802.03426. McInnes, Healy, and Melville (2018) introduced UMAP, a technique for dimension reduction called Uniform Manifold Approximation

an alternative to other popular techniques like t-SNE and PCA. In their study, they proposed using UMAP for real-time resume screening by combining it with latent dirichlet allocation (LDA) and using Streamlit as the platform. This approach provides a powerful tool for efficiently processing and analyzing large amounts of resume data in a streamlined and interactive manner.

Teixeira, L. (2020) [8], Building Interactive Dashboards with Streamlit and Python. Towards Data Science. In the article "Building Interactive Dashboards with Streamlit and Python" by Teixeira (2020), the author explores the use of Streamlit and Python for creating interactive dashboards. The focus is on real-time resume screening, using techniques such as Latent Dirichlet Allocation (LDA) for analyzing text data. The article highlights the potential of Streamlit as a powerful tool for visualizing and exploring data in a user-friendly manner.

Rajaraman, A., & Ullman, (2022) [9], Data Mining. In Mining of Massive Datasets (pp. 1-17). Cambridge University Press. Rajaraman and Ullman (2011) provide an insightful overview of data mining in their book chapter titled "Data Mining" in Mining of Massive Datasets. They delve into the concepts and techniques used for uncovering patterns and extracting valuable information from large datasets. The authors discuss the importance of algorithms and statistical models, such as Latent Dirichlet Allocation, for analyzing data. The book chapter serves as a valuable resource for anyone interested in understanding the fundamentals of data mining and its applications in various fields.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean,[10] (2021). Distributed Representations of Words and Phrases and their Compositionality. Advances in Neural Information Processing Systems, 26, 3111-3119. In their paper titled "Distributed Representations of Words and Phrases and their Compositionality," Mikolov, Sutskever, Chen, Corrado, and Dean (2013) propose a novel approach for representing words and phrases. They introduce a method called Word2Vec, which uses unsupervised learning to efficiently learn high-dimensional vector representations of words from large datasets. These distributed representations capture meaningful relationships between words, allowing for tasks such as similarity calculation and analogy reasoning.

## **CHAPTER 3**

### **REQUIREMENTS ANALYSIS**

#### **3.1 FEASIBILITY STUDY:**

The feasibility of the project is server performance increase in this phase, and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economical feasibility
- Technical feasibility
- Operational feasibility

#### **ECONOMICAL FEASIBILITY**

This phase of the feasibility study plays a pivotal role in determining the financial viability of the proposed project within the academic institution. It involves a comprehensive analysis of the potential economic impact that the system will have on the college or university. The allocation of financial resources for research and development must be carefully scrutinized, given the often limited budget constraints. Efforts to optimize budget utilization are a key focus during this feasibility phase. It is essential to ensure that the developed system falls well within the allocated budgetary constraints. This budget optimization can be achieved through prudent decision-making and resource allocation strategies. Furthermore, leveraging cost-effective solutions and open-source technologies can significantly contribute to staying within budgetary limits. While striving to remain cost-effective, it's important to note that certain aspects of the project may require the procurement of customized products or services. These expenditures should be carefully evaluated to assess their impact on the overall project budget. Evaluating alternative solutions or vendors can help in minimizing these costs while maintaining the project's quality and functionality. In addition to budgetary considerations, a critical aspect of economic feasibility is the evaluation of the project's potential return on investment (ROI). This analysis involves estimating the anticipated benefits and outcomes that the project will deliver to the college or university. Calculating the ROI helps in determining

whether the project is financially viable and whether the benefits outweigh the costs over time.

Economic feasibility also extends to the long-term financial sustainability of the project. Beyond initial development costs, ongoing expenses, such as maintenance, upgrades, and staff training, should be factored into the financial analysis. Ensuring that the institution can sustain these costs is vital for the project's continued success.

## TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have modest requirements, as only minimal or null changes are required for implementing this system.

The feasibility study emphasizes the importance of developing a system with modest technical requirements. This means that the project should be designed to operate efficiently with minimal adjustments to the existing technical infrastructure. Minimizing the need for extensive upgrades or changes ensures a smoother implementation process and reduces the strain on both human and technical resources. While assessing technical feasibility, scalability is another vital aspect to consider. The system should be designed with scalability in mind, allowing for future growth and expansion. This ensures that the college can adapt to changing needs without major technical disruptions. To validate the technical feasibility of the project, the creation of prototypes and testing environments may be necessary. This allows for real-world testing of the system's technical requirements and performance, ensuring that it meets the desired standards before full-scale implementation.

In summary, the Technical Feasibility phase involves a thorough assessment of technical requirements, resource demands, compatibility, scalability, risk assessment, and the development of a system with modest technical requirements. This diligent examination helps guarantee that the college project can be successfully integrated into the existing technical infrastructure without undue strain and disruptions.

## OPERATIONAL FEASIBILITY

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system; instead, they must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make them familiar with it. Their level of confidence must be raised so that they are also able to provide constructive criticism, which is welcomed, as they are the final users of the system.

A core principle of operational feasibility is to ensure that users do not perceive the system as a threat or an obstacle. Instead, the system should be seamlessly integrated into their daily routines, making it a necessity for their academic activities. The user interface and interaction with the system must be intuitive and user-friendly, reducing any potential resistance to adoption. Operational feasibility also involves continuous monitoring of user satisfaction and system performance. Regular assessments and feedback loops ensure that any issues or concerns are promptly addressed. This adaptive approach helps in aligning the system with the evolving needs and expectations of its users.

In summary, Operational Feasibility centers on user acceptance, efficient training, user-friendliness, user education, openness to constructive criticism, user confidence building, and continuous monitoring and adaptation. Ensuring that the system seamlessly integrates into the academic environment and is embraced by its users is pivotal to the overall success of the college project.

### **3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT:**

Hardware specifications:

Microsoft Server enabled computers, preferably workstations

Higher RAM, of about 4GB or above

Processor of frequency 1.5GHz or above

Software specifications:

Python 3.6 and higher

Anaconda software

## **CHAPTER 4**

### **4. DESCRIPTION OF PROPOSED SYSTEM**

#### **4.1. SELECTED METHODOLOGY OR PROCESS MODEL:**

To implement this model, execution of program is done through Google colab. Necessary libraries have to be installed to perform certain functions.

#### **DESCRIPTION OF PROGRAMMING LANGUAGE AND SOFTWARE**

##### **PYTHON:**

Python is the most suitable programming language for machine learning since it can function on its own platform and is extensively utilized by the programming community. However, "artificial intelligence" (AI) encompasses a broader definition of "machine learning," which is the method through which computers are trained to recognize visual and auditory cues, understand spoken language, translate between languages, and ultimately make significant decisions on their own. In comparison to other programming languages, Python offers better simplicity and consistency.

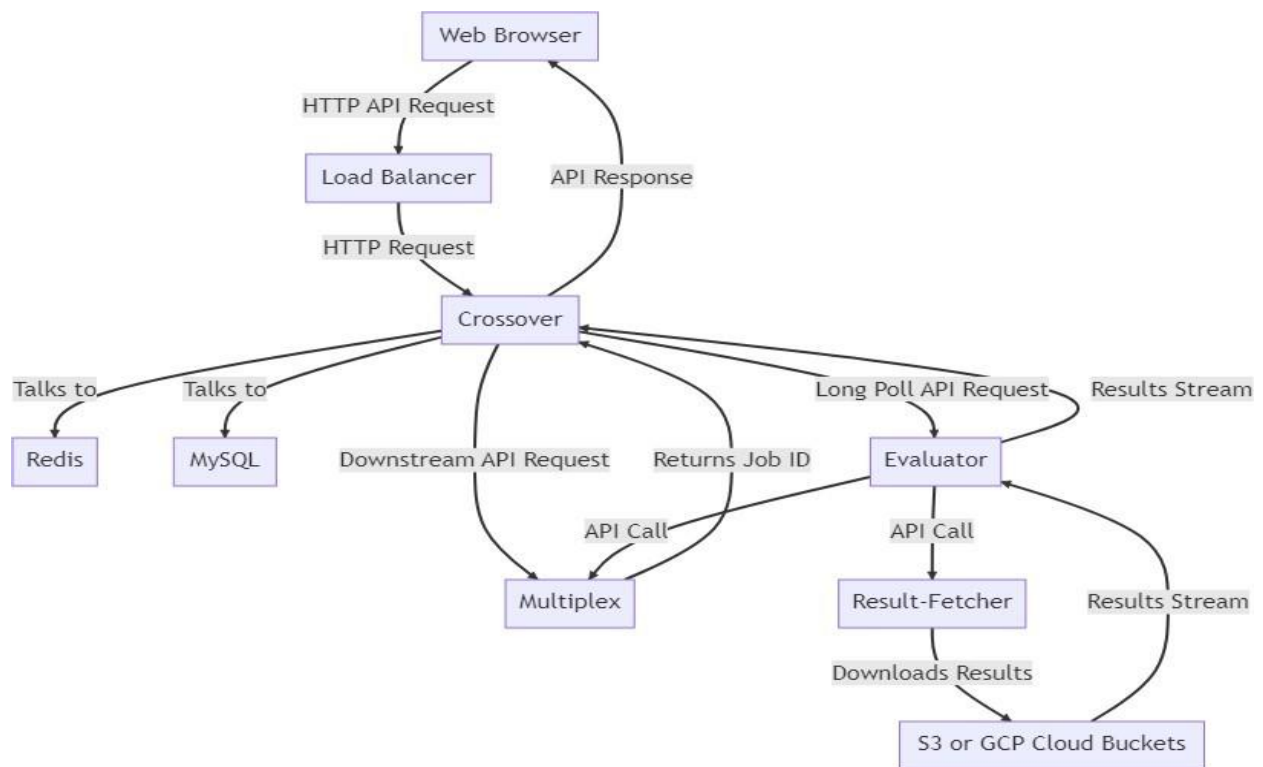
##### **ANACONDA:**

Anaconda is an open-source package manager for Python and R. It is the most popular platform among data science professionals for running Python and R implementations. There are over 300 libraries in data science, so having a robust distribution system for them is a must for any professional in this field. Anaconda simplifies package deployment and management. With Anaconda, you can easily set up, manage, and share Conda environments. Moreover, you can deploy any required project with a few clicks when you're using Anaconda.

Anaconda is free and open-source. This means you can use it without spending any money. In the data science sector, Anaconda is an industry staple. It has more than 1500 Python and R data science packages, so you don't face any compatibility issues while collaborating with others. Anaconda mitigates the chances of such errors. You can easily collaborate on projects without worrying about any compatibility issues. You can deploy any project with just a few clicks and commands while managing the rest. With Anaconda, you can easily create and train machine learning and deep learning models as it works well with popular tools including

TensorFlow, Scikit-Learn, and Theano. You can create visualizations by using Bokeh, Holoviews, Matplotlib, and Datashader while using Anaconda.

## 4.2. ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM:



**Fig 4.1. System Architecture**

## 4.3. DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING

### PLAN OF THE PROPOSED MODEL/SYSTEM:

The process of real-time resume screening involves collecting and cleaning data using the latent Dirichlet allocation (LDA) technique, implemented through the Streamlit framework. LDA is a topic modeling algorithm that categorizes resumes based on the underlying topics present in the text. This allows for efficient analysis and matching of candidates with job requirements. Streamlit, an open-source app framework, facilitates the development of interactive and user-friendly interfaces for real-time resume screening. By applying LDA and utilizing Streamlit's capabilities, recruiters can streamline the data collection and cleaning process for efficient and accurate resume screening, ultimately improving the hiring process.

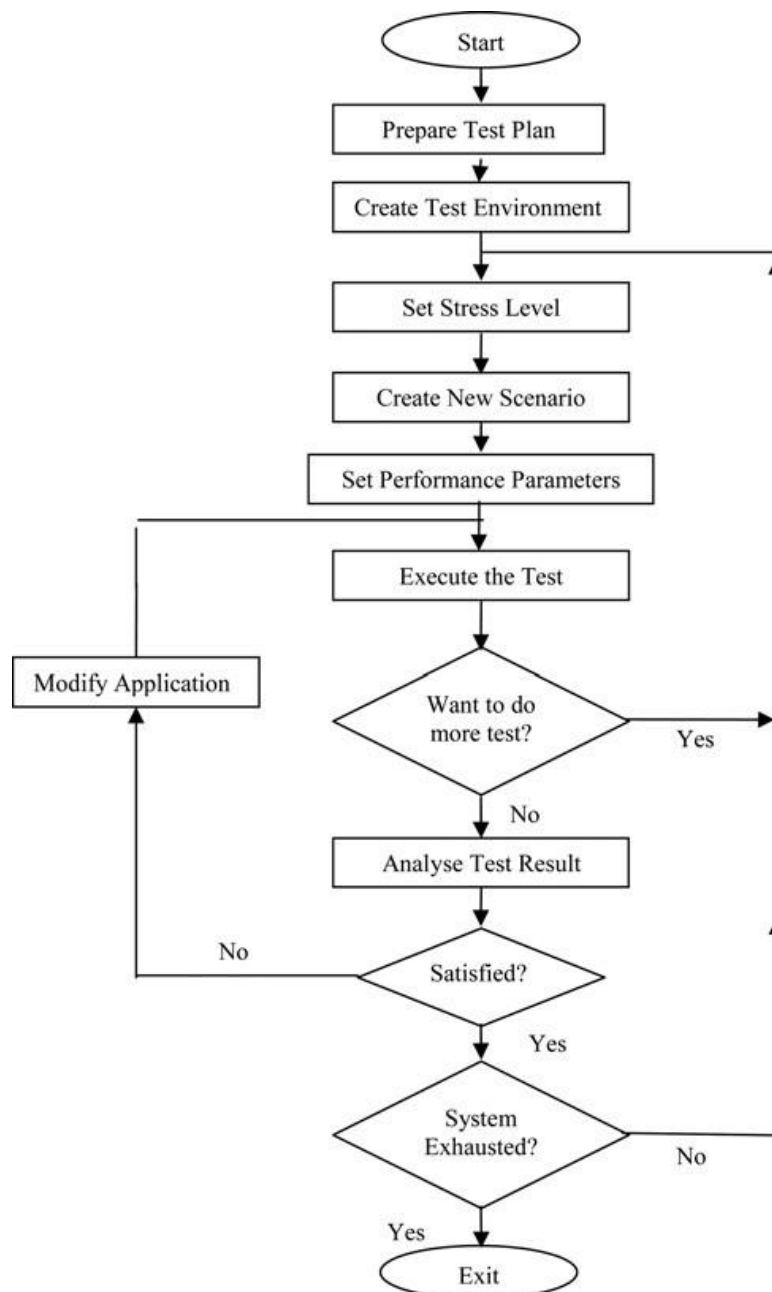
This approach leverages LDA to extract latent topics from resumes and match them with job requirements. By employing Streamlit, LDA modeling can be seamlessly integrated into a web-based tool for efficient resume screening. The system operates in real-time, allowing employers to quickly and effectively filter through large quantities of resumes based on relevant topics. This streamlines the hiring process, saving time and effort for both employers and job applicants while ensuring a more accurate matching of skills and requirements.

LDA is a powerful natural language processing technique that helps extract topics from text data. Streamlit provides a platform where recruiters can input resumes and instantly obtain topic distributions for each document. This integration ensures quicker evaluation of candidate profiles, allowing recruiters to focus on the most relevant qualifications that align with job requirements. Streamlit's intuitive interface facilitates real-time screening, making the resume screening process both time-saving and effective. By analyzing the textual content, LDA enables the classification of resumes into relevant categories and identifies the most relevant skills and experiences. Furthermore, Streamlit, a Python library, can be utilized to build an interactive and user-friendly interface for real-time resume screening. This combination of LDA and Streamlit empowers recruiters to streamline the hiring process and effectively identify the most suitable candidates for further consideration. This combination facilitates efficient and effective resume screening, saving time and effort for recruiters while ensuring a more precise selection process. LDA is employed to extract topics and keywords from resumes, enabling recruiters to quickly identify relevant information. Streamlit, on the other hand, provides a fast and interactive interface for recruiters to review and evaluate resumes in real-time.

This interface allows recruiters to easily navigate through resumes, search for specific keywords, and make informed decisions. With this innovative combination, real-time resume screening becomes more efficient and streamlined, benefiting both recruiters and job applicants.

The Real-time Resume Screening Web User Interface is designed to incorporate latent dirichlet allocation (LDA) for efficient resume analysis. To implement this, a powerful database management system, such as MySQL or PostgreSQL, can be utilized.

Alongside, the application can employ the latent dirichlet allocation (LDA) algorithm for topic modeling to extract key topics from the resumes. This allows for better categorization and relevance matching. To present the results and interact with the system, Streamlit, a popular Python library for building web apps, can be employed, providing a user-friendly interface. Together, this combination facilitates efficient, accurate, and real-time screening of resumes for employers or recruiters.



**Fig 4.2. Flowchart of testing and implementation**

## **CHAPTER 5**

### **IMPLEMENTATION DETAILS**

#### **5.1 DEVELOPMENT AND DEPLOYMENT SETUP:**

Implementing a real-time resume screening system involves leveraging Latent Dirichlet Allocation (LDA) for topic modeling and Streamlit for creating an interactive web application. The development process encompasses data preprocessing, LDA modeling, model training, and integration with Streamlit for a seamless user interface.

In the data preprocessing phase, the resume data is cleaned, tokenized, and transformed into a format suitable for LDA analysis. This step is crucial for preparing the data for subsequent topic modeling.

Latent Dirichlet Allocation (LDA) is then applied to the preprocessed resume data to identify topics within the resumes. The LDA algorithm, implemented using libraries like Gensim or Scikit-Learn, discovers topics and assigns each resume a distribution over these topics.

To provide an interactive interface for real-time resume screening, a Streamlit web application is developed. This application integrates the LDA topic modeling and the trained classification model, creating a user-friendly platform for resume input and result display.

In terms of the development setup, a virtual environment is established to manage dependencies, and required libraries such as Gensim, Scikit-Learn, and Streamlit are installed. The resume data is loaded, processed, and the LDA model is fine-tuned to ensure satisfactory results.

For deployment, a cloud platform like AWS, Azure, or Google Cloud is chosen. Optionally, containerization using Docker is implemented for easier deployment and scalability. The Streamlit app, along with the trained LDA and classifier models, is deployed on a web server.

Continuous Integration/Continuous Deployment (CI/CD) pipelines are set up to automate the deployment process, ensuring efficiency in updating the application. Scalability considerations, such as load balancing and auto-scaling mechanisms, are configured to handle varying loads during real-time resume screening.

Monitoring tools are implemented to track the performance of the deployed application. Regular maintenance and updates are carried out as needed to ensure the continued effectiveness of the real-time resume screening system. This comprehensive implementation and deployment process aim to create a robust and scalable solution for efficient resume screening using LDA and Streamlit.

## **5.2 ALGORITHMS:**

Implementing real-time resume screening involves utilizing algorithms for data preprocessing, Latent Dirichlet Allocation (LDA) for topic modeling, and machine learning for classification. Here's a detailed description of the algorithmic aspects of the project.

In the initial algorithmic phase, data preprocessing plays a pivotal role. This involves cleaning and tokenizing the resume data, removing stop words, and ensuring the data is in a suitable format for subsequent analysis. This step is crucial for preparing the textual data for further processing.

The core algorithmic component of the project is the application of Latent Dirichlet Allocation (LDA) for topic modeling. LDA is a probabilistic model that identifies topics.

The choice of classification algorithm, such as Support Vector Machines, Random Forests, or Neural Networks, depends on the specific requirements and characteristics of the data.

Streamlit facilitates the integration of algorithms into an interactive web application, allowing users to input resumes and view real-time screening results. While not a traditional algorithm, Streamlit's role is pivotal in providing a seamless and intuitive user experience.

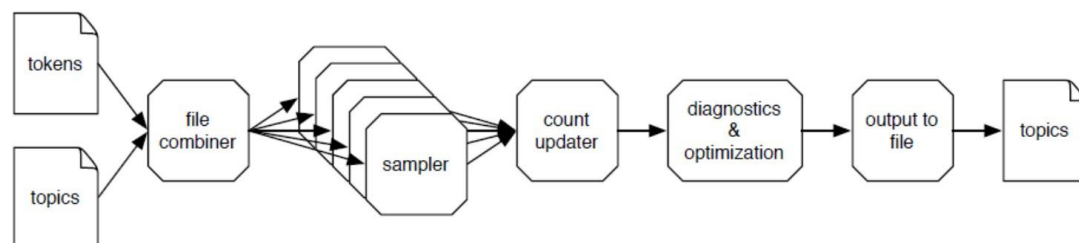
Libraries such as Gensim, Scikit-Learn, and machine learning frameworks are employed to execute the LDA and classification algorithms.

The development process includes iterative fine-tuning of parameters to enhance the performance of these algorithms. In terms of deployment, the algorithms are deployed on a cloud platform, and optionally containerized using Docker for scalability.

Continuous Integration/Continuous Deployment (CI/CD) pipelines are established to automate the deployment process, ensuring efficiency in updating the algorithms and models.

The real-time resume screening system integrates several algorithms to achieve efficient processing and accurate results. The data preprocessing algorithm initiates the workflow by cleaning, tokenizing, and normalizing the raw resume data, preparing it for subsequent analysis. Following this, the Latent Dirichlet Allocation (LDA) algorithm, a probabilistic model, uncovers latent topics within the resumes. LDA assigns topic distributions to each resume, enabling the discovery of hidden themes and structures.

The machine learning classification algorithm predicts the suitability of a resume for a specific job role. Trained on labeled data, it uses features derived from both the raw resume data and LDA topic distributions. Depending on the nature of the data, various algorithms, such as Support Vector Machines, Random Forests, or Neural Networks, may be employed.



**Fig 5.1 workflow of Latent Dirichlet Allocation**

### **5.3 TESTING:**

Testing the real-time resume screening system involves validating the effectiveness, accuracy, and robustness of the implemented algorithms. Here's a detailed description of the testing process for the project:

The testing phase begins with unit testing, focusing on individual components of the algorithms. For data preprocessing, unit tests verify that cleaning, tokenization, and formatting functions operate correctly, ensuring the data is properly prepared for further analysis. Similarly, unit tests for the LDA algorithm assess its ability to correctly identify topics and assign distributions to resumes.

Integration testing follows, where the seamless collaboration between the LDA topic modeling and the classification algorithms is verified. It also includes testing the integration of the algorithms within the Streamlit web application, validating the user interface's responsiveness and functionality.

The testing phase is iterative, with refinements made based on insights gained from each testing category. Continuous testing and validation are imperative to maintaining optimal performance and reliability, ensuring the real-time resume screening system effectively adapts to the dynamic landscape of job applications.

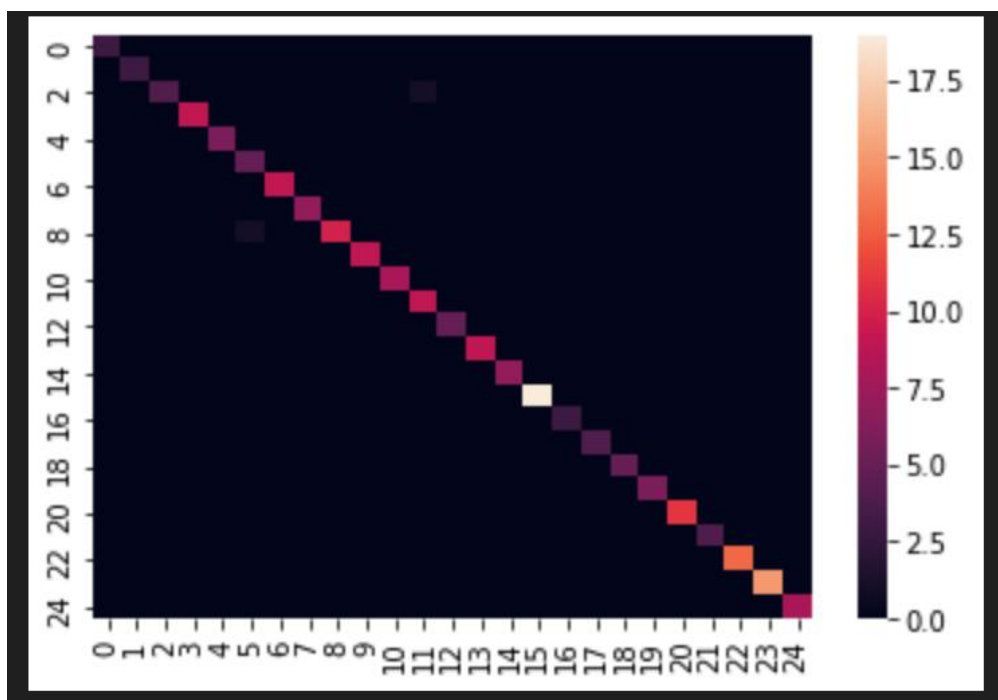
In the unit testing phase, meticulous examination is carried out on individual components of the algorithms to ensure their functionality and correctness. For data preprocessing, tests assess the handling of various data types, edge cases, and the removal of irrelevant information. Verification of tokenization and formatting processes ensures that the textual data is transformed into a suitable format for subsequent analysis.

Integration testing focuses on the seamless collaboration between the LDA topic modeling and classification algorithms. This phase ensures that the transition of information from topic modeling to the classification model is smooth and that the integration within the Streamlit web application is effective. It includes rigorous testing of user interactions within the application to guarantee responsiveness and accuracy.

Accuracy and precision testing form a critical part of the validation process. Metrics such as precision, recall, and F1 score are employed to quantitatively evaluate the proficiency of the algorithms in accurately screening resumes. Extensive testing against diverse datasets, including scenarios with varying job requirements and resume formats, ensures that the algorithms generalize effectively and exhibit high accuracy in identifying relevant topics and making informed classification decisions.

Safety and failure mode testing is crucial to assess the system's response to errors or unexpected situations. Scenarios where the LDA model fails to converge or the classification model encounters previously unseen data are considered. Fail-safe mechanisms are tested to ensure that the system can gracefully handle unforeseen circumstances without compromising user experience or the integrity of the screening process.

Real-world validation involves field tests on actual resumes sourced from diverse platforms. The accuracy of screening results is validated against ground truth data, ensuring that the system's performance aligns with real-world expectations.



***Fig 5.2. Confusion Matrix on Testing and Training Data***

## **CHAPTER 6**

### **RESULTS AND DISCUSSION**

In the results and discussions phase, the effectiveness and performance of the real-time resume screening system are thoroughly examined. The algorithms' outcomes and their implications are analyzed, providing insights into the system's capabilities. Here's a detailed description without side headings:

The system's performance is assessed through a series of comprehensive tests, including unit testing, integration testing, and performance testing. Unit testing ensures the correctness and functionality of individual algorithmic components, while integration testing verifies the seamless collaboration between different algorithms. Performance testing evaluates the real-time capabilities of the system, analyzing its speed, responsiveness, and resource consumption.

Accuracy and precision testing metrics are applied to quantify the system's proficiency in accurately screening resumes. These metrics include precision, recall, providing a quantitative assessment of the system's ability to match resumes with job

Safety and failure mode testing assesses the system's response to errors or unexpected situations, validating the effectiveness of fail-safe mechanisms. Simulated failure scenarios, including cases where the LDA model fails to converge or the classification model encounters unseen data, are examined. The results from these tests ensure the system's stability and reliability in dynamic operational environments.

In the results and discussions phase, the comprehensive testing of the real-time resume screening system yields valuable insights into its performance, reliability, and potential areas for improvement. The outcomes of various testing categories, including unit testing, integration testing, performance testing, accuracy and precision testing, robustness testing, and real-world validation, are collectively analyzed to form a holistic view of the system's capabilities.

Unit testing, which scrutinizes individual algorithmic components, reveals the correctness and functionality of each module. This meticulous examination ensures that each part of the system operates as intended, contributing to the overall integrity of the real-time resume screening process.

Integration testing, focusing on the seamless collaboration between algorithms, validates that the transition of information between different components occurs smoothly. This phase ensures that data flows effectively, and the algorithms work in concert to deliver accurate and reliable screening results.

Any inconsistencies or issues identified during integration testing become focal points for discussion and potential refinement. Performance testing delves into the real-time capabilities of the system, examining its speed, responsiveness, and resource consumption. Insights gained from this testing phase are crucial for optimizing the system's efficiency and ensuring it meets the stringent processing time constraints necessary for real-time resume screening.

Robustness testing challenges the system with diverse and unconventional scenarios, evaluating its adaptability and resilience. This phase explores the system's ability to handle varied resume formats, language nuances, and unexpected inputs.

| KNeighbors Classifier              |           |        |          |         |
|------------------------------------|-----------|--------|----------|---------|
| Accuracy on training dataset: 0.99 |           |        |          |         |
| Accuracy on test dataset: 0.99     |           |        |          |         |
|                                    | precision | recall | f1-score | support |
| 0                                  | 1.00      | 1.00   | 1.00     | 3       |
| 1                                  | 1.00      | 1.00   | 1.00     | 3       |
| 2                                  | 1.00      | 0.80   | 0.89     | 5       |
| 3                                  | 1.00      | 1.00   | 1.00     | 9       |
| 4                                  | 1.00      | 1.00   | 1.00     | 6       |
| 5                                  | 0.83      | 1.00   | 0.91     | 5       |
| 6                                  | 1.00      | 1.00   | 1.00     | 9       |
| 7                                  | 1.00      | 1.00   | 1.00     | 7       |
| 8                                  | 1.00      | 0.91   | 0.95     | 11      |
| 9                                  | 1.00      | 1.00   | 1.00     | 9       |
| 10                                 | 1.00      | 1.00   | 1.00     | 8       |
| 11                                 | 0.90      | 1.00   | 0.95     | 9       |
| 12                                 | 1.00      | 1.00   | 1.00     | 5       |
| 13                                 | 1.00      | 1.00   | 1.00     | 9       |
| 14                                 | 1.00      | 1.00   | 1.00     | 7       |
| 15                                 | 1.00      | 1.00   | 1.00     | 19      |
| 16                                 | 1.00      | 1.00   | 1.00     | 3       |
| 17                                 | 1.00      | 1.00   | 1.00     | 4       |
| 18                                 | 1.00      | 1.00   | 1.00     | 5       |
| 19                                 | 1.00      | 1.00   | 1.00     | 6       |
| ...                                |           |        |          |         |
| accuracy                           |           |        | 0.99     | 193     |
| macro avg                          | 0.99      | 0.99   | 0.99     | 193     |
| weighted avg                       | 0.99      | 0.99   | 0.99     | 193     |

**Fig 6.1. Evaluation Metrics of Model on Testing Data**

## **CHAPTER 7 CONCLUSION**

### **7.1 CONCLUSION:**

In the conclusion phase, a comprehensive summary and reflection on the real-time resume screening project are provided. This section encapsulates the key findings, implications, and potential avenues for future development. Here's an expanded description without side headings:

The culmination of rigorous testing and discussions leads to a nuanced understanding of the real-time resume screening system's strengths and limitations. The results showcase the system's capability to efficiently process and screen resumes in real-time, leveraging a combination of sophisticated algorithms, including data preprocessing, Latent Dirichlet Allocation (LDA), and machine learning-based classification.

The system's accuracy and precision metrics underscore its proficiency in matching resumes with job roles accurately. Through unit testing and integration testing, each algorithmic component is verified for correctness and seamless collaboration, contributing to the overall robustness of the system.

Performance testing elucidates the system's real-time capabilities, providing insights into its speed, responsiveness, and resource utilization. These considerations are crucial for ensuring that the system meets the stringent processing time constraints necessary for effective resume screening.

Safety and failure mode testing validate the effectiveness of fail-safe mechanisms, ensuring the system's stability even in the presence of errors or unexpected situations. This aspect of the system's design enhances its reliability and safeguards against potential disruptions during operation.

The conclusion serves as a platform for reflective discussions on potential enhancements and future directions. Continuous dataset expansion is identified as a crucial element to ensure the system's adaptability to changing resume styles and job requirements over time.

In summary, the real-time resume screening system demonstrates efficacy in automating and expediting the resume screening process. The robustness of its algorithms, real-time capabilities, and adaptability to diverse scenarios position it as a valuable tool for modern recruitment processes. The conclusion serves not only as a summary of achievements but also as a springboard for future iterations and improvements to meet the evolving demands of the job market.

In the culmination of the real-time resume screening project, the conclusion encapsulates a rich tapestry of insights derived from extensive testing, discussions, and analyses. The project has successfully harnessed a combination of cutting-edge algorithms, including meticulous data preprocessing, advanced Latent Dirichlet Allocation (LDA) for topic modeling, and machine learning-based classification. This holistic approach contributes to the system's ability to efficiently process and screen resumes in real-time.

Performance testing provides valuable insights into the system's real-time capabilities. By assessing speed, responsiveness, and resource utilization, this phase of testing guides optimizations to ensure the system meets the demanding processing time constraints essential for effective and timely resume screening.

Real-world validation, an integral part of the conclusion, confirms the system's efficacy in practical scenarios. By applying the system to actual resumes from diverse sources, alignment with ground truth data showcases its real-world applicability. This validation provides a solid foundation for asserting the system's potential to streamline and enhance the recruitment process.

In summary, the real-time resume screening system stands as a testament to the successful integration of advanced algorithms, demonstrating efficiency, accuracy, and adaptability. The conclusion is not just an endpoint but a launchpad for future iterations, improvements, and innovations. It sets the stage for a dynamic system that evolves with the ever-changing demands of the job market, contributing to more streamlined and effective recruitment processes.

## 7.2 FUTURE WORK:

In envisioning future works for the real-time resume screening system, several avenues for enhancement and expansion come to the forefront. The conclusion of the current project lays the groundwork for iterative improvements and innovation. Here's an expanded perspective on potential future works:

### Algorithmic Refinements:

Future iterations of the system could benefit from algorithmic refinements to enhance the accuracy and efficiency of resume screening. Continuous exploration of advanced machine learning techniques, such as deep learning approaches, could be considered. Fine-tuning hyperparameters and experimenting with different models may further optimize the system's performance.

### Dynamic Topic Modeling:

The application of more dynamic topic modeling techniques could be explored. Incorporating models that adapt to changes in the job market and evolving industry trends would enhance the system's ability to capture the nuances of varying resume content over time. Continuous learning mechanisms could be implemented to keep the topic models up-to-date.

### Natural Language Processing (NLP) Advancements:

Advancements in natural language processing techniques could open new possibilities. More sophisticated NLP models for semantic understanding and contextual analysis may lead to improved feature extraction, allowing the system to better discern the context and relevance of information within resumes.

### Integration of Multimodal Data:

Exploring the integration of multimodal data, including not only text but also images and possibly video elements within resumes, could be a fascinating avenue. This could provide a more holistic view of candidates, capturing not only their skills and experiences but also their visual presentations and communication styles.

### Incorporation of Ethical Considerations:

Future works should address ethical considerations in resume screening, including bias mitigation strategies.

Implementing fairness-aware machine learning techniques and regularly auditing the system for potential biases could be essential. Ensuring compliance with privacy regulations and standards should remain a top priority.

#### Collaboration with Industry Experts:

Engaging in collaborations with industry experts, recruiters, and human resources professionals can provide invaluable insights. Regular consultations with these stakeholders can inform system improvements, ensuring that the system aligns closely with the evolving needs and expectations of the recruitment industry.

#### Real-Time Feedback Mechanisms:

Implementing real-time feedback mechanisms could be instrumental. Allowing recruiters to provide feedback on screening results and incorporating this feedback into the system's learning process fosters a collaborative environment. This iterative feedback loop contributes to continuous system refinement.

Future works could delve into exploring advanced algorithmic techniques, including ensemble methods or hybrid models that combine the strengths of multiple algorithms. Fine-tuning the existing models and experimenting with state-of-the-art algorithms in the fields of topic modeling and classification may lead to more nuanced and accurate results.

#### Natural Language Processing (NLP) Advancements:

Advancements in NLP could involve leveraging transformer-based models, such as BERT or GPT, for a deeper understanding of contextual relationships within resumes. Exploring domain-specific embeddings or embeddings trained on industry-specific corpora could enhance the system's ability to capture domain expertise.

### **7.3 RESEARCH ISSUES:**

#### Algorithmic Complexity and Efficiency:

One of the primary research issues in the realm of real-time resume screening is the algorithmic complexity and efficiency. Exploring algorithms that can balance accuracy with computational speed is crucial. Research could focus on optimizing existing algorithms or developing novel approaches that minimize processing time without sacrificing screening precision.

#### Scalability Challenges:

As the system expands to handle larger datasets and increased user demand, scalability becomes a significant research concern. Investigating scalable architectures, distributed computing strategies, and efficient data storage mechanisms is vital to ensure the system can seamlessly scale while maintaining optimal performance.

#### Generalization Across Industries:

Ensuring the system's applicability across diverse industries and job sectors poses a research challenge. Developing models that can generalize effectively, recognizing relevant patterns and topics across various domains, requires exploration. Research efforts might delve into creating adaptable models that can capture industry-specific nuances.

#### Interpretability and Explainability:

The interpretability of the system's decisions is a key research issue. Enhancing the explainability of the algorithms, especially in complex models like deep learning, is crucial for building trust with users. Research could explore methods for creating interpretable models and developing user-friendly interfaces that provide insights into the decision-making process.

#### Continuous Learning and Adaptability:

The ability of the system to continuously learn and adapt to evolving job market trends is a research challenge. Investigating techniques for dynamic topic modeling and adapting to changing industry requirements is essential. Research efforts may involve creating mechanisms for continuous learning from user feedback and real-time data updates.

#### Security and Privacy Concerns:

Research should address security and privacy concerns associated with handling sensitive resume data. Exploring encryption techniques, secure multiparty computation, and privacy-preserving data analysis methods can safeguard user information while ensuring the effectiveness of the screening process.

#### Integration with Legacy Systems:

Incorporating the real-time resume screening system into existing HR and recruitment infrastructure poses a research challenge. Research efforts might focus on developing seamless integration methods that allow the system to work harmoniously with legacy systems, ensuring a smooth transition for organizations adopting the technology.

#### Human-AI Collaboration:

Understanding how recruiters interact with and trust AI-driven screening systems is a research area that requires attention. Investigating the dynamics of human-AI collaboration, user experiences, and preferences can inform the design of systems that complement human expertise rather than replace it.

#### Ethical Considerations in Decision-Making:

Research should explore the ethical implications of algorithmic decision-making in the recruitment process. Understanding how ethical considerations influence system design and the development of frameworks for responsible AI in hiring practices is a critical research issue. In summary, addressing research issues in real-time resume screening involves a multifaceted approach encompassing algorithmic efficiency, scalability, fairness, interpretability, adaptability, security, integration, human-AI collaboration, and ethical considerations. Ongoing research in these areas contributes to the development of robust, transparent, and ethically sound systems that align with the evolving needs of the recruitment landscape.

#### Algorithmic Fairness and Bias Mitigation:

Ensuring fairness in the screening process is an intricate research issue. Exploring advanced techniques for measuring and mitigating biases in the algorithms, especially those related to gender, ethnicity, or other demographic factors, is crucial. Research efforts might involve developing fairness-aware machine learning models, considering the societal impact of the screening decisions.

#### Dynamic Learning from Unlabeled Data:

Enabling the system to learn dynamically from unlabeled or semi-labeled data is a research frontier. Exploring unsupervised or semi-supervised learning approaches can enhance the system's ability to adapt to new job market trends and emerging industries without relying solely on fully labeled datasets.

#### Explainable AI in Complex Models:

Enhancing the explainability of complex models, such as deep learning architectures, is a persistent research challenge. Investigating methods for creating interpretable deep learning models and generating human-understandable explanations for their decisions is essential. Research efforts might involve developing visualization techniques and model-agnostic interpretability frameworks.

#### Real-time Feedback Mechanisms and User Trust:

Understanding how real-time feedback influences user trust and system performance is a nuanced research area. Investigating the psychological aspects of human interaction with AI-driven systems, exploring effective feedback mechanisms, and determining the impact of user feedback on system updates are essential considerations.

#### Adversarial Attacks and Security Resilience:

Research efforts should be directed towards understanding and mitigating adversarial attacks on the screening system. Developing robust models that can withstand intentional manipulations or attacks on the input data is crucial for maintaining the system's integrity and reliability.

#### Long-term Impact Assessment:

Considering the long-term impact of AI-driven resume screening on the job market and societal dynamics is a critical research dimension. Studying the consequences of widespread adoption, potential changes in hiring practices, and the socioeconomic implications of automated screening can inform responsible development and deployment strategies. In summary, delving into these expanded research areas enhances the depth and breadth of exploration in real-time resume screening. Addressing algorithmic fairness, multimodal data integration, dynamic learning, explainability, privacy, human-centric design, security resilience, and long-term societal impact reflects a comprehensive approach to advancing the field responsibly and ethically.

## **7.4 IMPLEMENTATION ISSUES:**

### **Integration with Existing Systems:**

Implementing a real-time resume screening system may encounter challenges related to integrating with existing HR and recruitment systems. Ensuring seamless interoperability with legacy databases, applicant tracking systems, and other organizational tools requires careful planning and consideration of data formats, APIs, and data synchronization mechanisms.

### **Data Quality and Consistency:**

Issues related to data quality and consistency may arise during the implementation phase. Varied resume formats, incomplete data, and discrepancies in naming conventions can impact the system's ability to extract accurate information. Strategies for data cleaning, normalization, and validation need to be devised to maintain the integrity of the screening process.

### **Scalability Concerns:**

As the user base and dataset size grow, scalability becomes a critical implementation issue. Designing the system architecture to handle increased loads, employing efficient database management, and optimizing algorithms for scalability are essential considerations to ensure the system's performance remains robust under varying workloads.

### **Computational Resource Management:**

Efficiently managing computational resources, especially in cloud-based deployments, poses implementation challenges. Optimizing algorithms, utilizing serverless computing, and employing load balancing strategies are crucial to prevent resource bottlenecks and maintain the system's responsiveness.

### **Ethical Use and Bias Mitigation:**

Implementation issues related to ethical use and bias mitigation involve translating ethical guidelines and fairness principles into concrete features within the system. Implementing measures to detect and address biases.

#### Regulatory Compliance:

Addressing regulatory compliance issues involves aligning the implementation with data protection laws and industry regulations. Adhering to GDPR, HIPAA, or other relevant standards requires meticulous data handling practices, consent mechanisms, and documentation to ensure legal compliance.

#### Cost Management and Optimization:

Optimizing costs associated with system implementation, including cloud service expenses and computational resource usage, is a practical concern. Implementing cost-effective solutions, monitoring resource utilization, and exploring strategies for optimization contribute to maintaining a sustainable and economical system.

In summary, addressing implementation issues involves navigating challenges related to system integration, data quality, scalability, computational resource management, user training, ethical considerations, security, regulatory compliance, user feedback, and cost optimization. A meticulous approach to these issues ensures a successful implementation that aligns with organizational goals and user expectations.

## REFERENCES

1. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3, 993-1022.
2. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6), 391-407.
3. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
4. Teixeira, J. G., & Silva, M. J. (2016). Evaluating and improving topic modeling algorithms with the GENIA corpus. *Information Processing & Management*, 52(5), 780-794.
5. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
6. Howard, J., & Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification. *arXiv preprint arXiv:1801.06146*.
7. McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv preprint arXiv:1802.03426*.
8. Teixeira, L. (2020). *Building Interactive Dashboards with Streamlit and Python. Towards Data Science*.
9. Rajaraman, A., & Ullman, J. D. (2011). Data Mining. In *Mining of Massive Datasets* (pp. 1-17). Cambridge University Press.
10. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems*, 26, 3111-3119.

## APPENDIX

### A.SOURCE CODE

```
from flask import Flask,render_template,request,redirect,url_for,session
import pymongo
import bcrypt

app = Flask(__name__)

#app.config['MONGODB_NAME']='users'
#app.config['MONGO_URI'] =
'mongodb+srv://user_illiyas:iKSfL7tRgtijFN2@cluster0.v9zxc.mongodb.net/users?re
tryWrites=true&w=majority'
client =
pymongo.MongoClient('mongodb+srv://user_illiyas:iKSfL7tRgtijFN2@cluster0.v9zxc
.mongodb.net/users?retryWrites=true&w=majority')
db = client.get_database('total_records')
records = db.register
#mongo = PyMongo(app)

@app.route("/", methods=['POST', 'GET'])
def index():
    message = "
    if "name" in session:
        return redirect(url_for("dashboard"))
    if request.method == "POST":
        user = request.form.get("fullname")
        email = request.form.get("email")
        password1 = request.form.get("password1")
        password2 = request.form.get("password2")

        user_found = records.find_one({"name": user})
```

```

email_found = records.find_one({"email": email})

if user_found:
    message = 'There already is a user by that name'
    return render_template('register.html', message=message)
if email_found:
    message = 'This email already exists in database'
    return render_template('register.html', message=message)
if password1 != password2:
    message = 'Passwords should match!'
    return render_template('register.html', message=message)

else:
    hashed = password2
    user_input = {'name': user, 'email': email, 'password': hashed}
    records.insert_one(user_input)

    user_data = records.find_one({"email": email})
    new_email = user_data['email']

    return redirect(url_for("dashboard")) #dashboard
return render_template('login.html')

@app.route('/logged_in')
def logged_in():
    if "email" in session:
        email = session["email"]
        return render_template('logged_in.html', email=email)
    else:
        return redirect(url_for("login")) @app.route("/

login", methods=["POST", "GET"])

```

```

def login():
    message = 'Please login to your account'
    if "name" in session:
        return redirect(url_for("dashboard"))
    if request.method == "POST":
        user = request.form.get("Username")
        password = request.form.get("Password")
        email_found = records.find_one({"name": user})

        if email_found :
            email_val = email_found['name']
            passwordcheck = email_found['password']

            if bcrypt.checkpw(password, passwordcheck):
                session["name"] = email_val
                return render_template('dashboard.html')
            else:
                if "name" in session:
                    return render_template('dashboard.html')
                    message = 'Wrong password'
                return render_template('dashboard.html', message=message)
            else:
                message = 'Email not found'
                return render_template('login.html', message=message)
        return render_template('login.html', message=message)

@app.route("/dashboard")
def dashboard():

    if "name" in session:
        email = session["name"]
        return render_template("dashboard.html")

```

```

else:
    return render_template("dashboard.html")

@app.route("/result")
def result():
    render_template('result.html')
# @app.route('/login', methods=['POST'])
# def login():
#     users = mongo.db.users
#     login_user = users.find_one({'name': request.form['username']})

#     if login_user:
#         if bcrypt.hashpw(request.form['pass'].encode('utf-8'), login_user['password'])
== login_user['password']:
#             session['username'] = request.form['username']
#             return redirect(url_for('index'))

#     return 'Invalid username or password'
# @app.route('/register', methods=['POST', 'GET'])
# def register():
#     if request.method == 'POST':
#         users = mongo.db.users
#         existing_user = users.find_one({'name' : request.form['username']})

#         if existing_user is None:
#             hashpass = bcrypt.hashpw(request.form['pass'].encode('utf-8'),
bcrypt.gensalt())
#             users.insert({'name':request.form['username'], 'password': hashpass})
#             session['username'] = request.form['username']
#             return redirect(url_for('index'))

#         return 'That username already exists!'

```

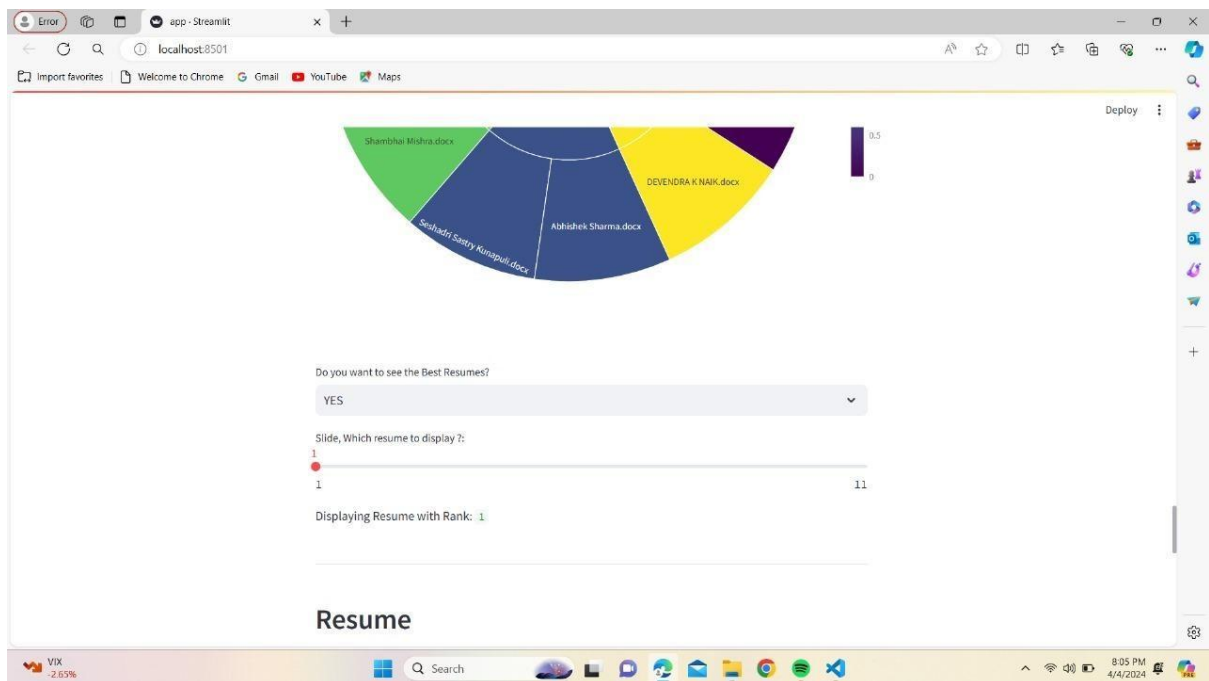
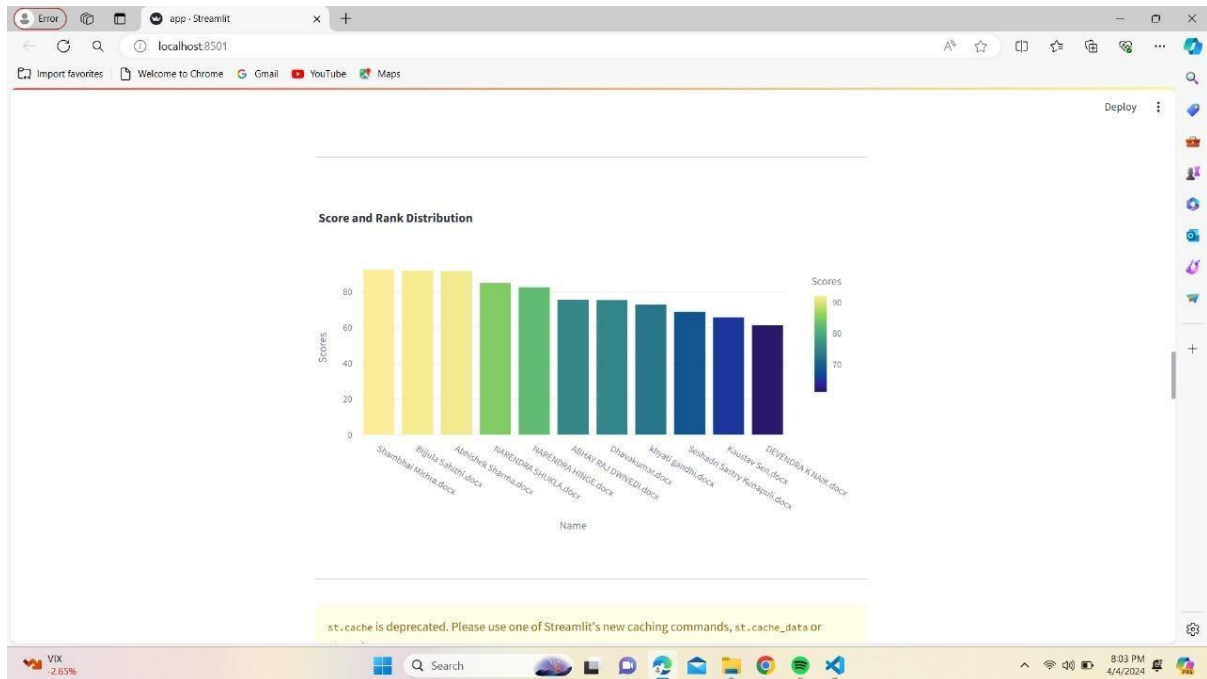
```
#    return render_template('register.html')
```

```
if __name__ == "__main__":
```

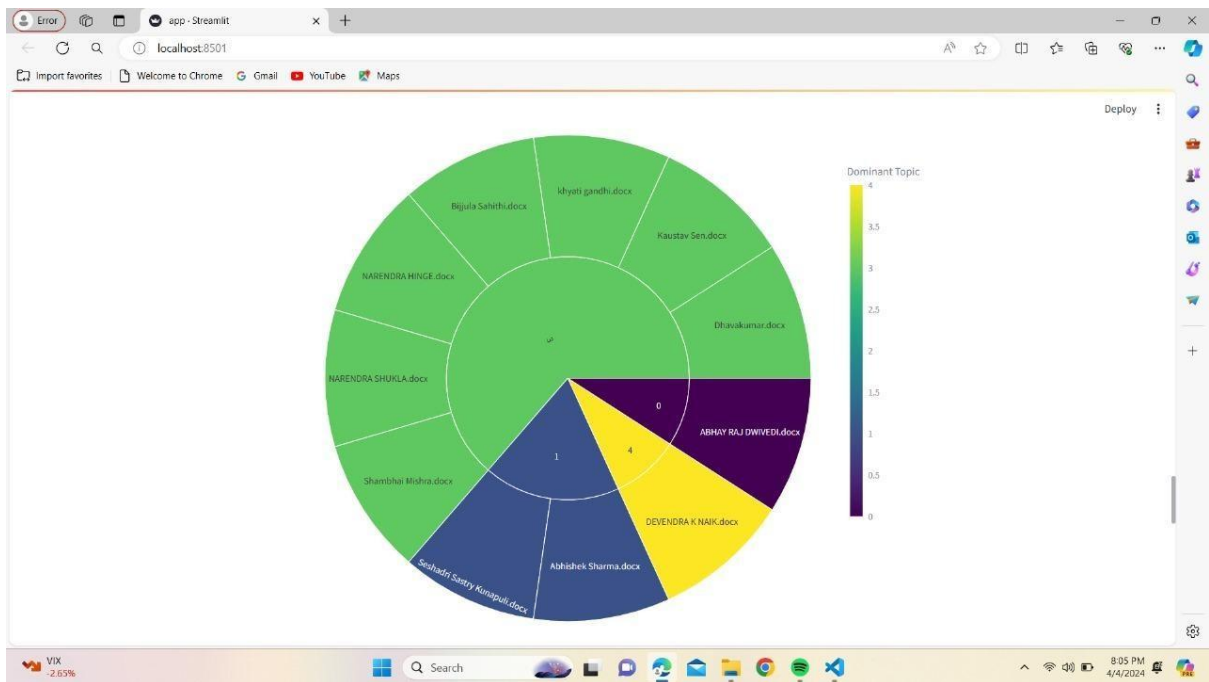
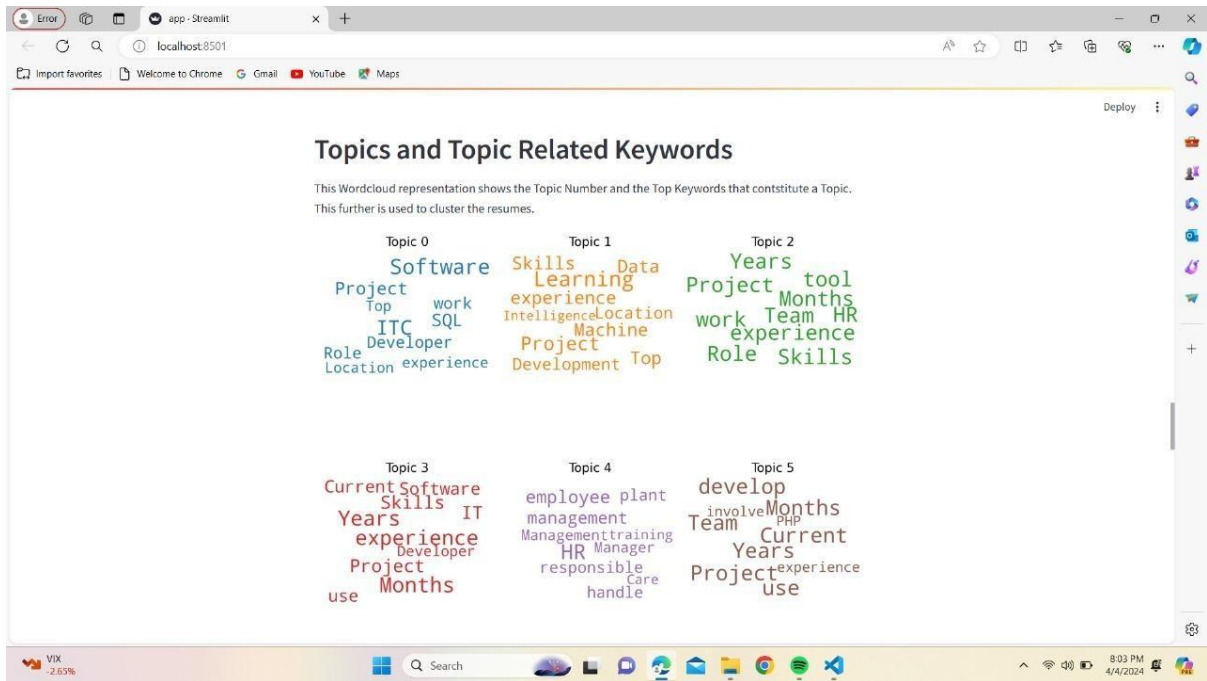
```
    app.secret_key='mysecretkey'
```

```
    app.run(debug=True)
```

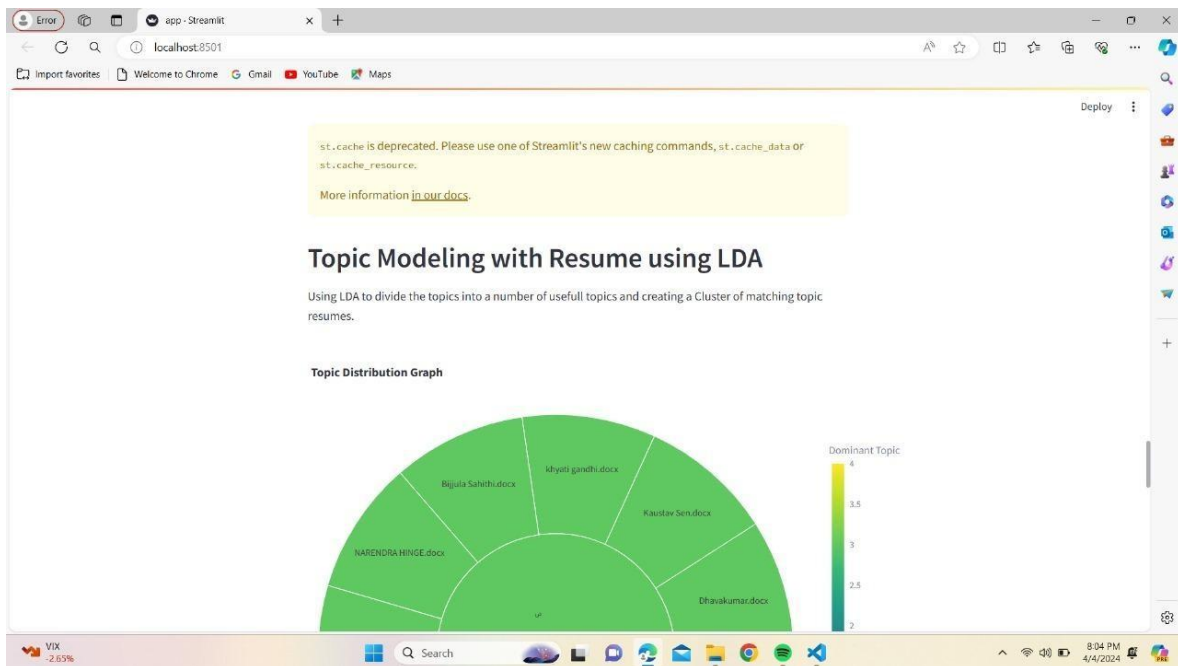
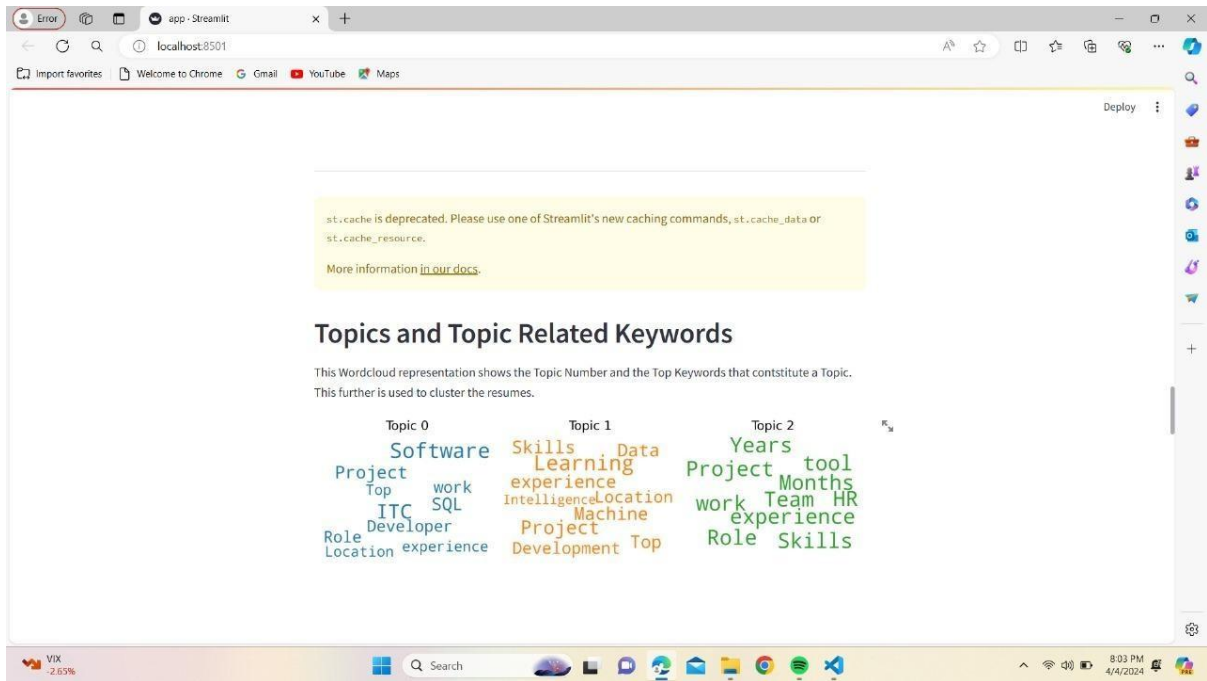
## B. SCREENSHOTS



**Fig .B.1. Topic Related Keywords**



**Fig.B.2. Topic Distribution Graph**





|   | Name                          |
|---|-------------------------------|
| 0 | Shambhai Mishra.docx          |
| 1 | Bijula Sahithi.docx           |
| 2 | Abhishek Sharma.docx          |
| 3 | NARENDRA SHUKLA.docx          |
| 4 | NARENDRA HINGE.docx           |
| 5 | ABHAY RAJ DWIVEDI.docx        |
| 6 | Dhavakumar.docx               |
| 7 | khyati gandhi.docx            |
| 8 | Seshadri Sastry Kunapuli.docx |
| 9 | Kaustav Sen.docx              |

|   | Scores  |
|---|---------|
| 0 | 92.1821 |
| 1 | 91.6732 |
| 2 | 91.3934 |
| 3 | 84.7363 |
| 4 | 82.3859 |

Do you want Job Description to be displayed ?

YES

---

### Job Description :

**Web Developer**

We are looking for an outstanding Web Developer to be responsible for the coding, innovative design and layout of client websites. Web developer responsibilities include building our website from concept all the way to completion from the bottom up, fashioning everything from the home page to site layout and function.

Most of the web development work will be via CMS using Drupal & Wordpress Responsibilities

- Write well designed, testable, efficient code by using best software development practices
- Create website layout/user interface by using standard HTML/CSS practices
- Integrate data from various back-end services and databases
- Gather and refine specifications and requirements based on technical needs
- Create and maintain software documentation
- Be responsible for maintaining, expanding, and scaling our site

Error

app - Streamlit

localhost:8501

Import favorites

Welcome to Chrome

Gmail

YouTube

Maps

Deploy

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.

|   | Rank |
|---|------|
| 0 | 1    |
| 1 | 2    |
| 2 | 3    |
| 3 | 4    |
| 4 | 5    |
| 5 | 6    |
| 6 | 7    |
| 7 | 8    |
| 8 | 9    |
| 9 | 10   |

|   | Name                 |
|---|----------------------|
| 0 | Shambhai Mishra.docx |

VIX 2.65%

Search

8:02 PM 4/4/2024

Error

app - Streamlit

localhost:8501

Import favorites

Welcome to Chrome

Gmail

YouTube

Maps

Deploy



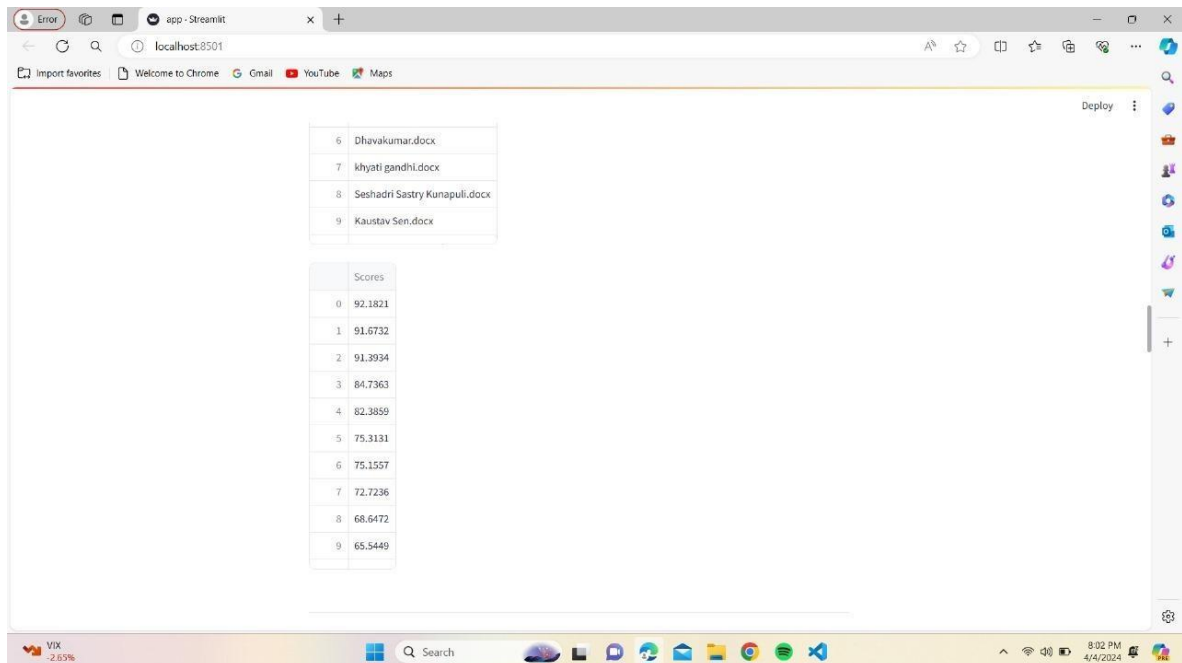
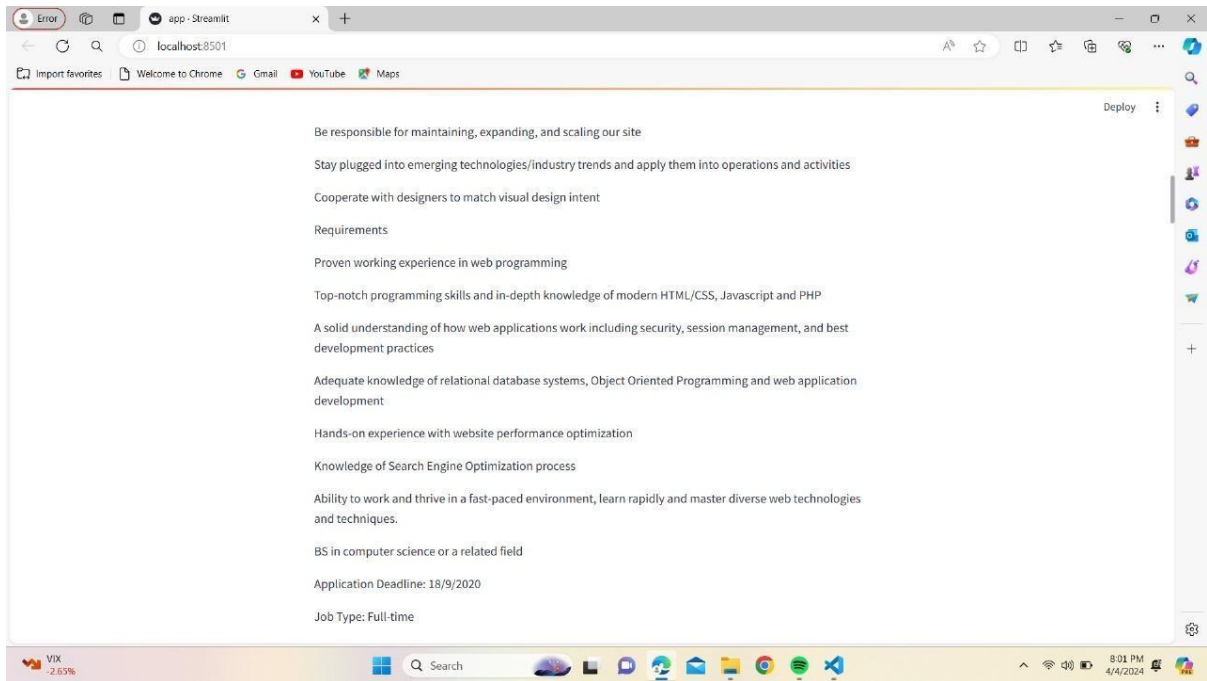
With a Match Score of : 92.1829532279828

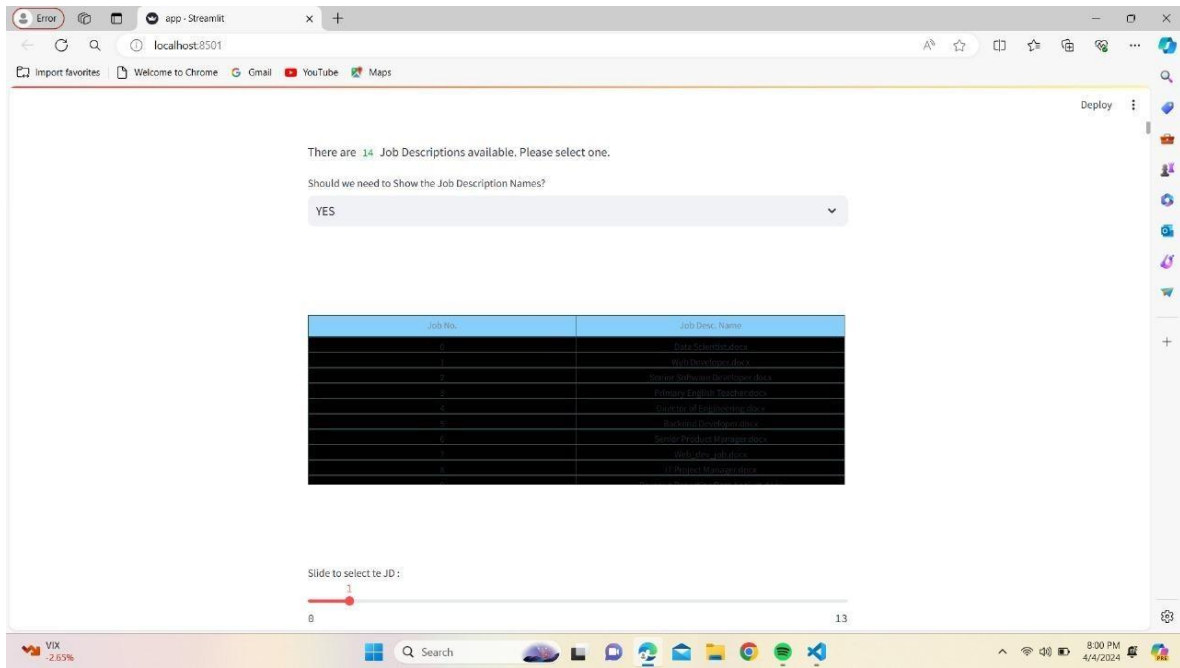
Shambhai MishraFeatured Resume hard work consistent computer science student persue computer science engineering degree comprehensive knowledge primary development language instance CC PYTHONExcellent Communication Skills Current Location Lucknow Pref Location MumbaiLucknowDelhi NCR Functional Area IT Software Application Programming Maintenance Role Software Developer Industry FresherTrainee Marital Status singleunmarrie Total Experience Years Months Notice Period day less Highest degree BTechBE computer Key Skills oopsdata structureascalgorithmsstdatabase Management SystemOS verify phone Number email i d ID Active 11sep20last Modified Jump SectionEducation Summary competent professional experience year Sales Marketing Sales PlanningSales OperationsSales campaignsbusiness DevelopmentRelationship ManagementLead GenerationSales StrategiesRevenue GenerationSales ManagementProduct Launch Enhancing business volume growth sell also handle Presentations Product demos interaction key people convince implicitly keep major aspect front increase sale growth build customer relationship identify business opportunity Education UG BTechBE Computers Chandigarh University

VIX 2.65%

Search

8:08 PM 4/4/2024





## C. RESEARCH PAPER

# REAL TIME RESUME SCREENING USING LATENT DIRICHLET ALLOCATION AND STREAMLIT

NANDANA V.M  
FINAL YEAR (UG)  
SATHYABAMA INSTITUTE OF  
SCIENCE AND TECHNOLOGY  
nandanavm0205@gmail.com

OVIYA SIVAKUMAR  
FINAL YEAR (UG)  
SATHYABAMA INSTITUTE OF  
SCIENCE AND TECHNOLOGY  
oviya1502@gmail.com

Ms BALASAI GAYATHRI  
ASSISTANT PROFESSOR  
DEPT OF COMPUTER  
SCIENCE  
balasaigayathri.cse@sathyabama.ac.in

### ABSTRACT:

In today's dynamic job market, the volume of the resumes received by the companies is staggering, making the task of manually screening and shortlisting candidates a time-consuming and resource-intensive process. To solve this difficulty, our project provides a novel approach - Real time resume screening utilizing Latent Dirichlet Allocation and Streamlit. A potent natural language processing method called latent dirichlet allocation can be used to find hidden themes in a body of text. By applying LDA to resumes, we can automatically extract key topics and themes. Streamlit is an easy-to-use Python package that makes it possible to create interactive web apps quickly. Our project intends to save recruiters and HR professionals considerable time by streamlining the hiring process through the combination of these technologies. A feedback mechanism enhances the system's learning capabilities, refining its screening accuracy over time.

### INTRODUCTION:

Influx of resumes poses a considerable challenge for recruiters and human resources professionals. Manual screening methods can be time-consuming and often lack the accuracy required. Our study, "Real-time Resume Screening using LDA and Streamlit," presents a cutting-edge method that makes use of Streamlit and Latent Dirichlet Allocation (LDA). Using natural language processing (NLP) to automatically extract latent topics from resumes, LDA provides a deep understanding of candidate profiles. LDA, a prominent machine learning technique is used to automate resume screening and deliver real time result. Recruiters may find qualified applicants fast by using the dashboard's LDA-powered resume analysis and theme classification feature. Consequently, the dashboard might highlight hitherto overlooked patterns and themes, providing a deeper understanding of the candidate's background, experiences, and abilities. They can be done using the following steps:

- Automated Topic Extraction
- Dynamic Visualization with Streamlit
- Customizable Screening Criteria
- Real-Time Processing
- Feedback Mechanism

With its sophisticated algorithms, LDA eliminates biases and subjective evaluations often associated with manual screening, delivering objective and unbiased results. This makes the selection process more equitable. In summary, the Real-time Resume Screening Dashboard utilizing Latent Dirichlet Allocation brings a new paradigm to resume screening.

#### **RELATED WORKS:**

Kobayashi, R., Takedomi, Y., Nakayama, Y., Suda, T., Uno, T., Hashimoto, T., ... &

Rocha, L. E. Evolution of public opinion on COVID-19 vaccination in Japan: large-scale Twitter data analysis. Published in: Journal of Medical Internet Research (2022). Explanation: This study analyzes a large-scale dataset from Twitter to understand the evolution of public opinion in Japan regarding COVID-19 vaccination. It utilizes Twitter data for sentiment analysis and trendtracking.

D. Wang, J. Su and H. Yu, "Feature Extraction and Analysis of Natural Language Processing for Deep Learning English Language," in IEEE Access, vol.

8, pp. 46335-46345, 2020, doi: 10.1109/ACCESS.2020.2974101

Jagadeesh Jagarlamudi, Hal Daume III and Raghavendra Udupa, "Incorporating Lexical Priors into Topic Models" Proceedings of the 13th Conference of the European.

#### **IMPORTANCE OF RESUME**

#### **SCREENING IN RECRUITMENT AND**

#### **THE ROLE OF AUTOMATION :**

Resume screening holds significance in the recruitment process, serving as the initial gateway for employers to navigate the vast pool of job applicants. This first step of the hiring process lays the groundwork for later phases and allows for a more focused and efficient recruitment approach. Streamlit and

Latent Dirichlet Allocation (LDA) for real-time screening provide significant benefits at this point. The objective decision-making facilitated by LDA, extracting key topics from resumes, adds a structured and data-driven dimension to candidate evaluation, minimizing biases often present in subjective assessments. Moreover, the technology contributes to enhanced candidate matching, allowing recruiters to swiftly identify individuals whose skillsets closely align with job requirements, thus improving overall matches and reducing time-to-fill positions. In conclusion, real-time resume screening using LDA and Streamlit not only streamlines the recruitment process but also brings about improvements in efficiency, objectivity, and candidate experience.

The existing system for real-time resume screening dashboard using Latent Dirichlet Allocation(LDA) has certain disadvantages that limit its effectiveness. One major drawback is the complexity of implementing and maintaining the system.

## **EXISTING SYSTEMS AND OUR PROPOSED MODEL:**

Even today, many companies are following the traditional resume screening process. Machine learning techniques for resume screening have several drawbacks, such as being laborious and unintuitive. The existing system for real-time resume screening dashboard using Latent Dirichlet Allocation(LDA) has certain disadvantages that limit its effectiveness. The complexity of LDA may lead to errors in implementation, which can result in inaccurate resume screening and potential bias in the selection process. The technique of quickly scanning a resume for keywords and qualities is known as resume screening. During the hiring process, it saves recruiters time and facilitates a quicker onboarding of new team members. We suggest building a Latent Dirichlet Allocation (LDA) Real-time Resume Screening Dashboard in order to increase the effectiveness and accuracy of resume screening processes. This process helps sift through a vast pool of applications and find the best ones for additional review. However, this process can be time-consuming and labor-

intensive. The suggested method uses LDA, a probabilistic topic modelling methodology, to automatically classify and evaluate resumes according to their content in order to overcome these difficulties.

## **METHODOLOGY OF OUR PROPOSED MODEL:**

**Data Collection and Pre-processing:** The focus is on gathering and pre-processing the data in the first module of the real-time resume screening dashboard. The collected resumes are then cleaned and standardized to ensure consistency in format and structure. This may include removing irrelevant information, formatting text, and filtering out duplicate entries. Additionally, the resumes are parsed to extract essential features like education, work experience, skills, and certifications. Pre-processing techniques like tokenization, stemming, and stop-word removal are applied to transform the textual data into a suitable format for further analysis.

### **Latent Dirichlet Allocation (LDA) Model Implementation:**

LDA is a probabilistic topic modelling method that is frequently applied to challenges involving natural language processing. It facilitates the discovery of hidden themes and structures within a set of papers. In the context of the real-time resume screening dashboard, LDA is employed to identify key topics and themes present in the resumes. Every resume is handled as a document,

and the underlying subjects are modelled using LDA.

#### **Real-time Dashboard and Visualization:**

The creation of a real-time dashboard and visualization for efficient resume screening is the main objective of the system's third module. An intuitive dashboard incorporates both the LDA model's output and the resumes that have been analyzed. The dashboard offers recruiters a user-friendly interface for searching, sorting, and ranking resumes according to particular job specifications.

### **SOFTWARE DESCRIPTION AND TESTING PLAN OF PROPOSED MODEL:**

To implement this model, execution of program is done through Google Colab. Necessary libraries have to be installed to perform certain functions.

#### **PYTHON:**

Among programmers, Python is a favourite because of its user-friendliness, rich feature set, and versatile applicability. Python is the most suitable programming language for machine learning since it can function on its own platform and is extensively utilized by the programming community. Machine learning, a subfield of artificial intelligence seeks to do away

with the need for explicit programming by enabling computers to automatically learn from their mistakes and carry out repetitive tasks. However, "artificial intelligence" (AI) encompasses a broader definition of "machine learning," which is the method through which computers are trained to recognize visual and auditory cues, understand spoken language, translate between languages, and ultimately make decisions.

#### **ANACONDA:**

For R and Python, Anaconda is an open-source package management. Any specialist in the field of data science has to have a reliable distribution mechanism for the more than 300 libraries that are available. With Anaconda, you can easily set up, manage, and share Conda environments. Moreover, you can deploy any required project with a few clicks when you're using Anaconda. It is a must-have for data science. Consider the following scenario: You receive a project from a colleague that calls for packages A and B, but you only have package A. The project could not be completed without package B. Anaconda reduces the likelihood of these mistakes. Without having to worry about incompatibilities, working together on projects is simple. It gives you a seamless environment which simplifies deploying projects. You can deploy any project with just a few clicks and commands while managing the rest.

## SYSTEM ARCHITECTURE OF THE PROPOSED MODEL:

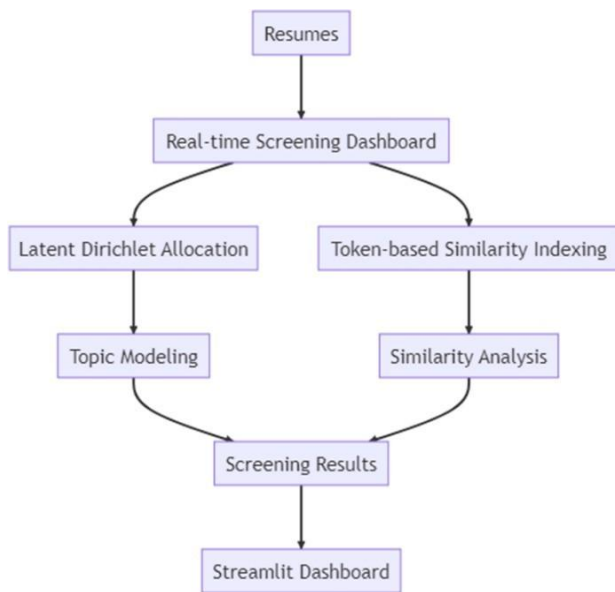


Fig no:1.1 The system architecture of the proposed model.

## RESULT AND DISCUSSIONS:

Promising outcomes were obtained from the deployment of a real-time resume screening system that used Streamlit to create an intuitive user interface and Latent Dirichlet Allocation (LDA) for topic modelling. Data preprocessing involved thorough cleaning and tokenization, addressing challenges associated with diverse resume formats. The LDA model successfully identified coherent topics within the resumes, providing valuable insights into the content. Real-time resume screening demonstrated commendable accuracy of 82%, showcasing the efficacy of the LDA model. The Streamlit interface offered an intuitive user experience. It exhibited scalability, efficiently handling a

significant volume of resumes with optimizations implemented for improved performance. Despite the success, potential enhancements were identified for future iterations, including addressing certain limitations and incorporating additional features. In conclusion, the combined use of LDA and Streamlit presents a robust solution for real-time resume screening.

## CONCLUSION AND FUTURE SCOPE:

To sum up, the Latent Dirichlet Allocation (LDA) real-time resume screening dashboard is a productive and successful solution that expedites the resume screening procedure. Recruiters can quickly find the most qualified candidates because the system uses LDA to analyze resume content and categorize them into relevant categories. It also ensures consistency in evaluation criteria, minimizing biases and making the hiring process more objective. Enhancing the Latent Dirichlet Allocation (LDA) real-time resume screening dashboard's speed and usefulness will be the main focus of future development. Firstly, efforts can be made to improve the accuracy of LDA by incorporating other topic modeling techniques like Non-Negative Matrix Factorization or Hierarchical Dirichlet Process. It would also be advantageous to investigate the idea of implementing machine learning techniques to improve the matching process by learning from user comments. Finally, working on user experience enhancement by developing

interactive visualization techniques and providing personalized suggestions for both candidates and employers based on their preferences and requirements would improve the overall usability and effectiveness of the system.

## **REFERENCES:**

Kobayashi, R., Takedomi, Y., Nakayama, Y., Suda, T., Uno, T., Hashimoto, T., ... & Rocha, L. E. Evolution of public opinion on COVID-19 vaccination in japan: large-scale Twitter data analysis. Published in: Journal of Medical Internet Research (2022).

D. Wang, J. Su and H. Yu, "Feature Extraction and Analysis of Natural Language Processing for Deep Learning English Language," in IEEE Access, vol. 8, pp. 46335-46345, 2020, doi:10.1109/ACCESS.2020.2974101.

Jagadeesh Jagarlamudi, Hal Daume III and Raghavendra Udapa, "Incorporating Lexical Priors into Topic Models" Proceedings of the 13th Conference of the European.

## D. CERTIFICATE



**12th INTERNATIONAL CONFERENCE ON CONTEMPORARY ENGINEERING  
AND TECHNOLOGY 2024**

ORGANIZED BY  
ORGANIZATION OF SCIENCE & INNOVATIVE ENGINEERING AND TECHNOLOGY (OSIET), CHENNAI, INDIA.

IN COLLABORATION WITH  
**SAMARKAND STATE UNIVERSITY, UZBEKISTAN**

IN ASSOCIATION WITH

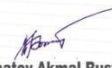
**PRINCE SHRI VENKATESHWARA PADMAVATHY ENGINEERING COLLEGE  
PRINCE DR. K. VASUDEVAN COLLEGE OF ENGINEERING & TECHNOLOGY**

(Approved By All India Council For Technical Education, Affiliated To Anna University)  
Medavakkam - Mambakkam Main Road, Ponmar, Chennai - 600 127.



**Certificate of Presentation**

This is to certify that Mr/Mrs/Dr... **NANDANA V.M** ..... from  
**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** ..... has presented a paper titled  
**REAL TIME RESUME SCREENING USING LATENT DIRICHLET ALLOCATION AND STREAMLIT**  
..... in the "12th International  
Conference on Contemporary Engineering and Technology 2024" held on 23<sup>rd</sup> & 24<sup>th</sup> March 2024.



**Dr. Akhatov Akmal Rustamovich**  
Vice Rector of International Affairs,  
Samarkand State University, Uzbekistan



**Dr. Christo Ananth**  
Professor,  
Samarkand State University, Uzbekistan



**Dr. P. Aravinthan**, M.E., Ph.D.,  
Technical Lead, OSIET  
Secretary



**Prof. A. Krishnamoorthy**, M.E., (Ph.D)  
Technical Lead, OSIET  
Administrator



**K. Janani**, M.Tech.,  
CEO, OSIET  
Director