An

Artificial intelligence project report

on

Resume Shortlist Manager

Submittedby:

Manish Kumar (220561)

Rithik Yadav (220502)

Robin Yadav (220589)

Akshay Menon (220556)

Under mentorship of

Dr. Atul Mishra

(Assistant Professor)

****

Department of Computer Science Engineering

School of Engineering and Technology

BML MUNJAL UNIVERSITY, GURUGRAM (INDIA)

November 2024

# CANDIDATES DECLARATION

We hereby certify that the work on the project entitled, “**Resume Short lister”** in partial fulfilment of requirements for the award of Degree of **Bachelor of Technology** in School of Engineering and Technology at BML Munjal University, is an authentic record of our own work carried out during a period from August 2024 to December 2024 under the supervision of **Dr. Atul Mishra**.

**Manish Kumar**

**Rithik Yadav  
Robin Yadav**

**Akshay Menon**

# SUPERVISOR’S DECLARATION

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

**Faculty Supervisor Name: Dr. Atul Mishra**

**Signature:**

# ACKNOWLEDGEMENT

We are highly grateful to **Dr. Atul Mishra, Assistant Professor**, BML Munjal University, Gurugram, for providing supervision to carry out the seminar/case study from August - December 2024.

**Dr. Atul Mishra** has provided great help in carrying out our work and is acknowledged with reverential thanks. Without wise counsel and able guidance, it would have been impossible to complete the training in this manner.

We would like to express thanks profusely to thank **Dr. Atul Mishra**, for stimulating us from time to time. We would also like to thank the entire team at BML Munjal University. We would also thank our friends who devoted their valuable time and helped us in all possible ways toward successful completion.

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Description** | **Page No.** |
| 1 | Results NLP for parsing | 17 |
| 2 | Results with Gemini Model | 19 |
| 3 | Results with NLP (base nltk) | 20 |
| 4 | Resume Parser | 20 |
| 5 | Homepage | 21 |
| 6 | Job Description and weights | 21 |
| 7 | CGPA Threshold and Results | 22 |

# TABLE OF CONTENTS

[CANDIDATES DECLARATION 2](#_Toc184108470)

[SUPERVISOR’S DECLARATION 2](#_Toc184108471)

[ACKNOWLEDGEMENT 3](#_Toc184108472)

[LIST OF FIGURES 4](#_Toc184108473)

[TABLE OF CONTENTS 5](#_Toc184108474)

[ABSTRACT 7](#_Toc184108475)

[1. INTRODUCTION 8](#_Toc184108476)

[1.1 Problem Statement 8](#_Toc184108477)

[1.2 Objectives 9](#_Toc184108478)

[1.3 Motivation 9](#_Toc184108479)

[1.4 Significance 9](#_Toc184108480)

[1.5 Challenges 10](#_Toc184108481)

[1.6 Novelty Proposed 10](#_Toc184108482)

[2. LITERATURE REVIEW 11](#_Toc184108483)

[2.1 Summary of Research Works 11](#_Toc184108484)

[2.2 Comparison Table 12](#_Toc184108485)

[3. METHODOLOGY 13](#_Toc184108486)

[3.1 State Space Search 13](#_Toc184108487)

[3.1.1 State Space Definition 13](#_Toc184108488)

[3.1.2 Search Strategy 13](#_Toc184108489)

[3.2 Knowledge Representation 14](#_Toc184108490)

[3.2.1 Representation Technique: 14](#_Toc184108491)

[3.2.2 Implementation Details: 14](#_Toc184108492)

[3.2.3 Appropriateness and Justification: 14](#_Toc184108493)

[3.3 Intelligent System Design 15](#_Toc184108494)

[3.3.1 System Architecture: 15](#_Toc184108495)

[3.3.2 Components and Functionalities: 15](#_Toc184108496)

[3.3.3 Innovations: 15](#_Toc184108497)

[3.4 Bonus Points 16](#_Toc184108498)

[3.4.1 Originality: 16](#_Toc184108499)

[3.4.2 Ethical Considerations: 16](#_Toc184108500)

[3.5 Fuzzy Logic Application 16](#_Toc184108501)

[3.5.1 Fuzzy Sets and Rules: 16](#_Toc184108502)

[3.5.2 Implementation and Handling of Uncertainty: 16](#_Toc184108503)

[3.6 Parsing Resume 17](#_Toc184108504)

[4. Analysis and Discussion of Results 18](#_Toc184108505)

[4.1 User Engagement and Feedback 18](#_Toc184108506)

[4.2 Matching Accuracy 18](#_Toc184108507)

[4.3 Data Privacy and Security 18](#_Toc184108508)

[4.4 Weighted Scoring and Customization 18](#_Toc184108509)

[4.5 Real-Time Performance 19](#_Toc184108510)

[4.6 Comparative Evaluation with Traditional Hiring Tools 19](#_Toc184108511)

[4.7 Comparison with NLP and Gemini Model 19](#_Toc184108512)

[4.7.1 Using Gemini Model 19](#_Toc184108513)

[4.7.2 Using NLP 20](#_Toc184108514)

[4.8 Results 20](#_Toc184108515)

[5. Conclusion and Future Scope 23](#_Toc184108516)

[5.1 Conclusion 23](#_Toc184108517)

[5.2 Future Scope 23](#_Toc184108518)

[References 25](#_Toc184108519)

# ABSTRACT

One major drawback of the recruitment process is that there often is a step of resume culling, which is a very tedious and Manual filtering of resumes Thus, this process is vulnerable to bias. This project proposes an automated resume parser as well as candidate matching for quicker and more efficient recruitment. The system is able to parse resumes to identify academic performance, work experience, and projects and compare such information with job specifications, with the ability for the user to define weights for factors like CGPA, projects handled, and skills respectively. First, it is possible to mention some first results achieved using only classical NLP technologies and the Gemini model, which using revealed some drawbacks of these approaches, for example, the inability to consider context in text and the possibility of feature overweight. Then, the widely used BERT model from Google — transformer, boosted the feature extraction, context comprehension and candidate ranking capabilities comprehensively. This approach offers a solid, context-based solution for the identification of the best candidates to recruit, increases the efficiency, and effectiveness of the process. Additional future developments are to enhance parsing for graphical resumes, to implement dynamic feedback and to provide more details on the ranking of the candidates.

# 1. INTRODUCTION

The “SmartHire” project aims to revolutionize the recruitment process by leveraging artificial intelligence for resume parsing and candidate-job matching. This system is designed to extract key features such as CGPA, projects, and professional experience from resumes and compare them with job profiles based on weighted criteria specified by recruiters. By automating this process, “SmartHire” reduces the time and effort involved in shortlisting candidates while enhancing decision-making accuracy.

Built on advanced AI models, the system utilizes state-of-the-art natural language processing (NLP) techniques and machine learning algorithms to ensure robust feature extraction and relevance-based ranking. This ensures an intelligent and unbiased evaluation of candidates tailored to the specific requirements of the job.

## 1.1 Problem Statement

Traditional methods of resume screening are often manual, time-intensive, and prone to human biases. While applicant tracking systems (ATS) exist, they usually rely on basic keyword matching and fail to capture the contextual nuances of resumes and job descriptions. Additionally, candidates may be evaluated unevenly based on one-dimensional metrics like CGPA, leaving significant factors like projects, skills, and professional experience underutilized.

This project addresses the limitations of existing methods by developing “SmartHire,” an AI-powered resume parser and job matcher. It provides recruiters with a data-driven solution to identify the most suitable candidates while considering user-defined weights for various criteria.

## 1.2 Objectives

* Develop a comprehensive system for resume parsing, feature extraction, and contextual understanding of job descriptions.
* Implement a weighted scoring mechanism to allow recruiters to prioritize criteria such as CGPA, skills, projects, and experience.
* Ensure robust candidate ranking based on a balance of recruiter-defined importance and AI-driven contextual understanding.
* Provide an intuitive user interface for uploading resumes, inputting job descriptions, and generating ranked lists of candidates.
* Incorporate explainability features to justify ranking decisions for transparency and recruiter confidence.

## 1.3 Motivation

With the increasing volume of job applicants and the growing complexity of job profiles, traditional recruitment methods have become less effective. Recruiters face challenges in efficiently evaluating candidates, often overlooking top talent due to time constraints or biases.

The motivation behind “SmartHire” is to bridge this gap by automating the initial stages of recruitment. Inspired by the limitations of existing ATS and manual screening, this project integrates advanced AI technologies to deliver a solution that is accurate, scalable, and customizable.

## 1.4 Significance

“SmartHire” provides an innovative approach to recruitment by automating resume parsing and candidate-job matching. It ensures that:

* Recruiters save time and effort by receiving pre-ranked candidates tailored to job requirements.
* A evaluation is conducted, considering multiple criteria instead of just CGPA or keywords.
* The system fosters unbiased decision-making, reducing human errors and improving diversity in shortlists.
* Recruiters gain actionable insights through an explainable AI system that provides transparency in rankings.

## 1.5 Challenges

Developing “SmartHire” involves addressing several challenges:

1. *Unstructured Data*: Resumes vary in format and style, making it difficult to extract structured data accurately.
2. *Contextual Understanding*: Job descriptions often use varied language, requiring models to understand context beyond mere keywords.
3. *Weight Prioritization*: Designing a system that adapts to user-defined weightage without introducing bias.
4. *Model Performance*: Initial models like Gemini overemphasized CGPA, necessitating more context-aware alternatives like BERT.
5. *Data Privacy*: Ensuring resumes and job descriptions are processed securely, in compliance with data protection regulations.

## 1.6 Novelty Proposed

“SmartHire” introduces several innovative features that differentiate it from existing solutions:

* *Weighted Scoring System*: Allows recruiters to assign importance to different criteria, making the matching process highly customizable.
* *Explainable AI*: Provides detailed insights into why candidates were ranked, ensuring transparency and trust in the system.
* *Scalability and Accessibility*: Designed to handle large volumes of resumes and job profiles while being user-friendly and cost-effective.

# 2. LITERATURE REVIEW

## 2.1 Summary of Research Works

1. **"***Automated Resume Screening Using Machine Learning***"** - (2019)  
   This paper explores the potential of machine learning algorithms in resume screening, focusing on automating feature extraction and candidate ranking. It reviews existing tools and techniques, emphasizing the importance of contextual understanding in job-candidate matching.
2. **"***Job Recommendation System Using Natural Language Processing***"** - (2020)  
   The study examines how NLP models are used to parse and analyze resumes and job descriptions. It demonstrates the application of NLP in extracting key candidate attributes such as skills, education, and experience, providing personalized job recommendations.
3. **"***Explainable AI for Recruitment: A Transparent Approach to Candidate Ranking***"**-(2021)  
   This research introduces explainable AI (XAI) techniques in recruitment systems. It discusses how transparency in decision-making enhances recruiter trust and highlights the importance of interpretable models in candidate evaluation.
4. **"***Context-Aware Candidate Matching Using Deep Learning***"**-(2022)  
   The paper highlights the use of deep learning models like BERT for understanding the semantic alignment between resumes and job descriptions. It provides insights into improving the accuracy of candidate-job matching by capturing contextual relationships.
5. **"***Weighted Scoring Systems in Recruitment: A Case Study***"**-(2023)  
   This study investigates the implementation of weighted scoring systems, focusing on user-defined criteria like CGPA, skills, and projects. It outlines the advantages of customizable scoring mechanisms in enhancing recruitment outcomes.

## 2.2 Comparison Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Paper Title | Authors | Model/Method Used | Conclusion Drawn | Link |
| Resume Shortlisting and Ranking with Transformers | Vinaya James, Akshay Kulkarni, Rashmi Agarwal | SBERT, BERT | SBERT outperforms BERT, providing a ranked list of candidates with better matching for job requirements. | <https://link.springer.com/chapter/10.1007/978-3-031-35081-8_8> |
| Automatic Shortlisting of Candidates in Recruitment | Girish K. Palshikar, Rajiv Srivastava, et al. | Hand-Crafted Scoring, Neural Networks | Combines domain expertise and neural networks to optimize ranking and candidate selection. | <https://ceur-ws.org/> |
| A Smart Resume Screening Tool for Customized Shortlisting | Poonam Tijare, Mohammed Waseem, et al. | AI, ML Techniques | Automated model achieves 90% accuracy, creating profiles and filtering resumes for efficient screening. | <https://peoplemanagingpeople.com/tools/best-resume-screening-software/> |
| Resume Shortlisting Using NLPS. Ambareesh, Nikhil Kumar Thakur, et al. | S. Ambareesh, Nikhil Kumar Thakur, et al. | NLP, NER Models | NLP and NER models improve fairness and efficiency by ranking resumes based on a calculated final score. | <https://ieeexplore.ieee.org/document/10502580/> |

# 3. METHODOLOGY

## 3.1 State Space Search

### 3.1.1 State Space Definition

* **States:**  
  Represent the various stages of the recruitment process, such as parsing resumes, matching candidates to jobs, and ranking applications based on recruiter-defined criteria.
* **Initial State:**  
  The system begins in an idle state, awaiting a recruiter to upload job requirements or a candidate to submit a resume.
* **Goal State:**  
  The desired outcomes include generating a ranked list of candidates for a job posting or identifying the best-fit jobs for a candidate's profile.
* **Possible Actions:**  
  These include parsing resumes, analysing job descriptions, matching candidates, generating ranked lists, and displaying the results to users.

### 3.1.2 Search Strategy

* **Description of Algorithm:**
  + **Resume Parsing and Candidate Matching:**  
    Uses natural language processing (NLP) to extract attributes from resumes and job descriptions, such as skills, education, and experience.
  + **Weighted Scoring Mechanism:**  
    Combines recruiter-defined weights for attributes like education and skills to rank candidates.
  + **Interaction with Databases:**  
    Fetches resumes and job postings, processes them, and stores results in a structured format for further evaluation.
* **Justification and Implementation:**
  + **Candidate Matching:**  
    A scoring algorithm ranks candidates based on the degree of match between resumes and job descriptions.
  + **Storage:**  
    Data is stored in collections representing resumes, job descriptions, and rankings, allowing for efficient querying and retrieval.

## 3.2 Knowledge Representation

### 3.2.1 Representation Technique:

The system employs a *Weighted Attribute Model* to represent candidate-job compatibility. Attributes like skills, experience, and education are weighted based on recruiter preferences.

### 3.2.2 Implementation Details:

* **Attributes:**  
  Skills, education, and experience are represented as numerical or categorical variables.
* **Weighting System:**  
  Attributes are assigned weights by recruiters to reflect their importance in candidate evaluation.
* **Storage:**  
  Uses CSV format to store parsed data and calculated scores for efficient matching and retrieval.

### 3.2.3 Appropriateness and Justification:

The weighted attribute model ensures flexibility and aligns with real-world recruitment practices, enabling tailored candidate evaluations based on job-specific requirements.

## 3.3 Intelligent System Design

### 3.3.1 System Architecture:

The system is designed with the following layers:

* **User Interface Layer:**  
  Enables job seekers to upload resumes and view job recommendations and allows recruiters to post jobs and view ranked candidates.
* **Processing Layer:**  
  Handles resume parsing, job matching, and candidate ranking, integrating AI to ensure accurate and efficient evaluations.
* **Response Layer:**  
  Provides recruiters with ranked candidate lists and candidates with personalized job recommendations.

### 3.3.2 Components and Functionalities:

* **Resume Parser:**  
  Extracts key attributes like skills and education from resumes using NLP.
* **Job Matcher:**  
  Matches resumes to job descriptions using similarity algorithms and scoring mechanisms.
* **Ranking Engine:**  
  Ranks candidates based on recruiter-defined criteria and weighted scoring.

### 3.3.3 Innovations:

* **Customizable Weights:**  
  Recruiters can assign weights to attributes, tailoring evaluations to specific job roles.
* **Real-Time Matching:**  
  Dynamically matches resumes and job postings as they are added to the system.
* **Recommendation Engine:**  
  Suggests jobs to candidates based on their profiles and similarity score.

## 3.4 Bonus Points

### 3.4.1 Originality:

"SmartHire" integrates customizable weighted scoring, real-time job matching, and personalized recommendations to streamline recruitment and enhance user experience.

### Ethical Considerations:

* **Privacy:**  
  Ensures candidate data is securely stored and anonymized to maintain confidentiality.
* **Transparency:**  
  Clearly communicates ranking methodologies and system limitations to users.

## 3.5 Fuzzy Logic Application

### 3.5.1 Fuzzy Sets and Rules:

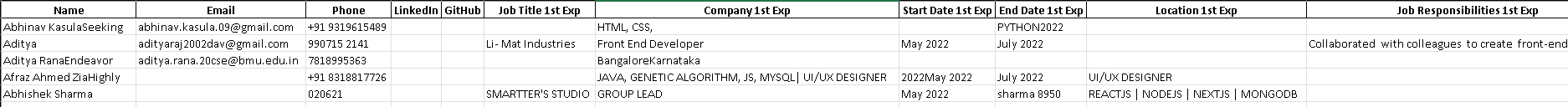
* **Fuzzy Sets:**  
  Represent the strength of a candidate’s match to a job, categorized as "Excellent," "Good," "Average," or "Weak."
* **Fuzzy Rules:**  
  Define actions such as recommending a candidate with a score above 75% or highlighting training resources for weaker matches.

### 3.5.2 Implementation and Handling of Uncertainty:

* **Uncertainty Management:**  
  Uses fuzzy logic to handle ambiguous or incomplete data, such as partially filled resumes or vague job descriptions.
* **Dynamic Recommendations:**  
  Adjusts job suggestions based on real-time changes in user input, ensuring relevance and accuracy.

## 3.6 Parsing Resume

When using NLP to parse data from resumes, we encountered issues with generating a consistent dataset due to discrepancies in PDF formats and varying alignments across CVs. This lack of uniformity in structure often led to misinterpretation or incomplete extraction of information.



*Fig. 1 Results NLP for parsing*

That's why we switched to GPT-4o-mini, an affordable solution that delivered significantly more accurate results, effectively handling the inconsistencies in PDF formats and CV alignments.

# 4. Analysis and Discussion of Results

## 4.1 User Engagement and Feedback

The "SmartHire" platform received positive feedback during initial testing. Recruiters appreciated the streamlined process for posting jobs and receiving candidate recommendations, while job seekers valued the personalized job suggestions. However, users highlighted areas for improvement, including the need for greater transparency in the candidate ranking process and broader job recommendations.

## 4.2 Matching Accuracy

The system demonstrated reliable performance in matching candidates to job postings based on skills, qualifications, and experience. Mismatches occurred occasionally, particularly when resumes were insufficiently detailed or job descriptions lacked specificity. Future updates aim to address these issues by refining the natural language processing models and improving the scoring algorithms.

## 4.3 Data Privacy and Security

The platform effectively protected user data by implementing secure storage and anonymization techniques. Users expressed trust in these measures, emphasizing the importance of robust privacy practices in ensuring confidence and compliance with ethical standards.

## 4.4 Weighted Scoring and Customization

The weighted scoring mechanism was praised for its flexibility, allowing recruiters to prioritize specific criteria such as skills or education. Some users requested additional options for customization, such as integrating behavioural traits or cultural fit considerations, to make the ranking process more comprehensive.

## 4.5 Real-Time Performance

The system performed efficiently, delivering results promptly even for complex matching tasks. However, during periods of high usage or when relying on external APIs, response times occasionally increased. Future improvements will focus on optimizing system architecture and API integrations to enhance scalability.

## 4.6 Comparative Evaluation with Traditional Hiring Tools

Compared to traditional recruitment methods, "SmartHire" offered significant benefits, including streamlined workflows and enhanced efficiency. Users noted that the platform made the hiring process more intuitive and aligned with their expectations, fostering a positive experience for both recruiters and job seekers.

## 4.7 Comparison with NLP and Gemini Model

### 4.7.1 Using Gemini Model

While we tested and user Gemini model for achieving the same results, we encountered a serious issue there. Each time we fed the data to Gemini, it replied with names with alphabetical orders only.

A computer screen shot of a computer code

Description automatically generated

*Fig. 1 Results with Gemini Model*

### 4.7.2 Using NLP

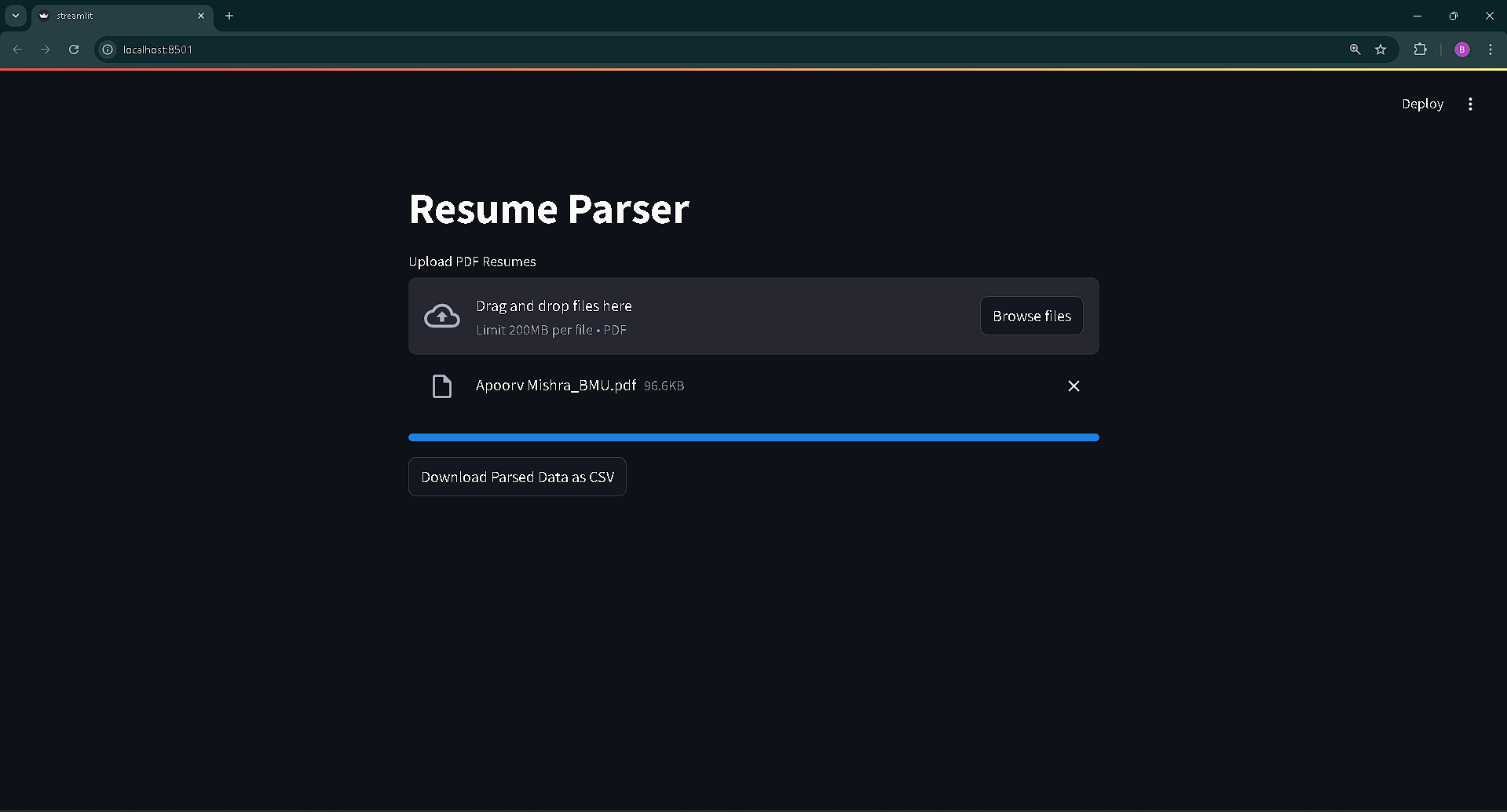
BERT is also an NLP model, but results with NLP were different and not accurate. That was the reason we chose BERT over NLP. Cosine similarity of results was not good to be considered for shortlisting the resumes.

A screenshot of a black screen

Description automatically generated

*Fig. 2 Results with NLP (base nltk)*

## 4.8 Results



*Fig. 3 Resume Parser*

A screenshot of a computer

Description automatically generated

*Fig.4 Homepage*

A screenshot of a computer

Description automatically generated

*Fig.5 Job Description and weights*

A screenshot of a computer

Description automatically generated

*Fig.6 CGPA Threshold and Results*

# 5. Conclusion and Future Scope

## 5.1 Conclusion

The "SmartHire" platform successfully integrates AI into recruitment, offering a seamless and efficient experience for both recruiters and job seekers. By leveraging advanced algorithms for candidate-job matching, the system personalizes recommendations and enhances decision-making processes. Its user-friendly interface and customizable features provide a valuable resource for modern hiring practices, bridging the gap between technology and human resources. The project demonstrates the potential of AI-driven platforms in revolutionizing traditional recruitment while maintaining an emphasis on fairness, transparency, and inclusivity.

In summary, "SmartHire" exemplifies how AI can transform the hiring process by making it faster, more accurate, and more user-focused. The platform’s adaptability and practical design ensure its utility for diverse recruitment needs, marking a significant step forward in the evolution of AI technology.

## 5.2 Future Scope

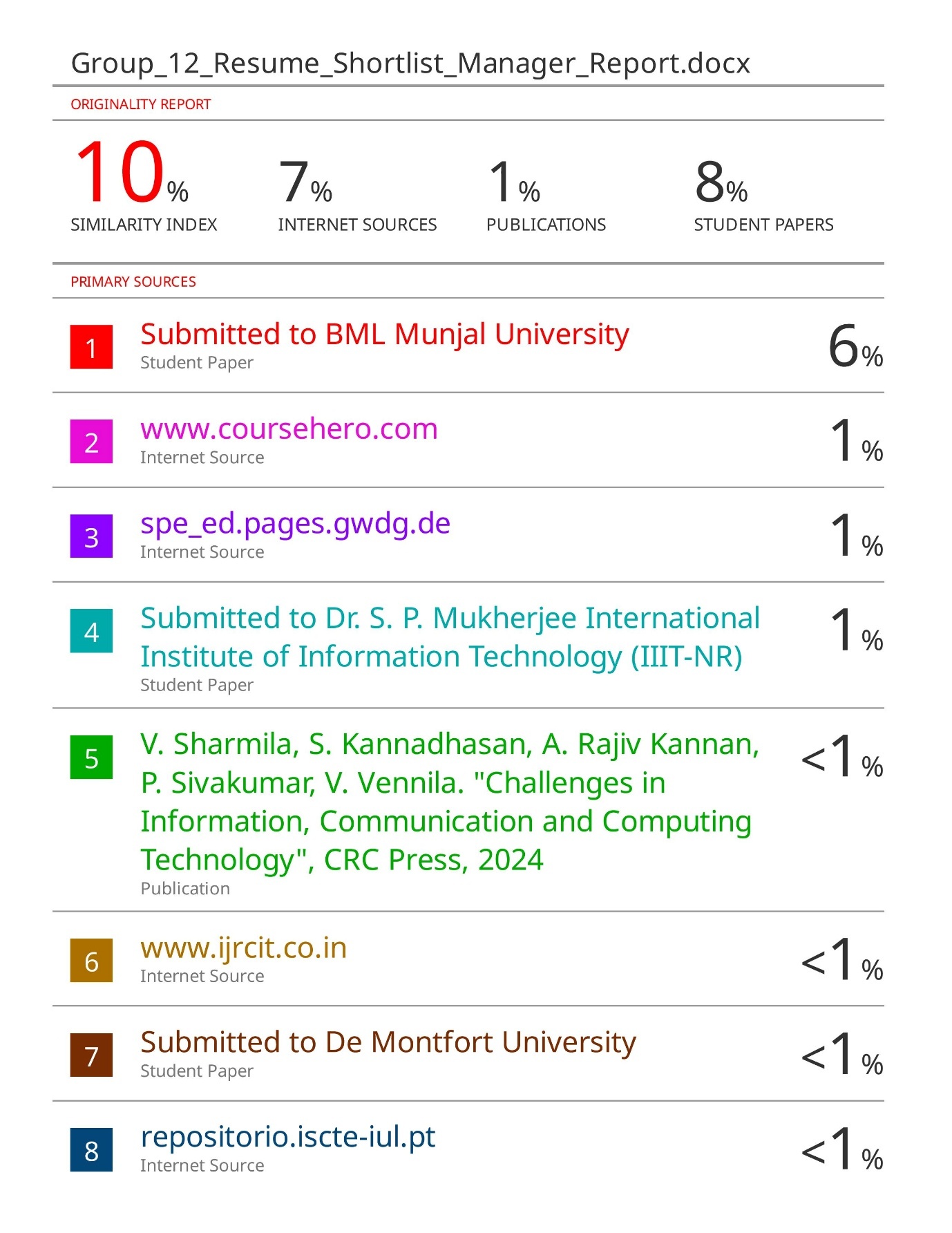
The "SmartHire" platform has significant potential for growth and innovation. Future enhancements could include:

* *Multilingual Support***:** Enabling the platform to operate in multiple languages, making it accessible to a wider, global audience.
* *Voice Interaction***:** Integrating voice recognition and response features to facilitate hands-free interaction and improve accessibility for users with varying needs.
* *Behavioral Analysis Integration***:** Incorporating tools to analyze behavioral traits from resumes or interviews, providing a holistic view of candidates beyond qualifications and experience.
* *Cultural Fit Assessment***:** Adding features to evaluate and align candidates' cultural compatibility with the hiring organization, enhancing long-term retention.
* *Predictive Analytics***:** Leveraging machine learning to predict job market trends and candidate success rates, helping recruiters make data-driven decisions.
* *Accessibility Enhancements***:** Including options such as text-to-speech and integration with assistive technologies to support individuals with disabilities, ensuring inclusivity.

These developments would further enhance the platform's value, ensuring its relevance in an ever-evolving job market and catering to diverse user needs.

# References

1. James, V., Kulkarni, A., Agarwal, R. (2023). Resume Shortlisting and Ranking with Transformers. In: Nandan Mohanty, S., Garcia Diaz, V., Satish Kumar, G.A.E. (eds) Intelligent Systems and Machine Learning. ICISML 2022. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 471. Springer, Cham. <https://doi.org/10.1007/978-3-031-35081-8_8>
2. S. Ambareesh, N. K. Thakur, U. Bhattarai, S. K. Yadav, J. N. Thakur and A. K. Mahato, "Resume Shortlisting Using NLP," *2024 4th International Conference on Data Engineering and Communication Systems (ICDECS)*, Bangalore, India, 2024, pp. 1-5, doi: 10.1109/ICDECS59733.2023.10502580.
3. A. Mukherjee and U. S. M, "Resume Ranking and Shortlisting with DistilBERT and XLM," *2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)*, Bangalore, India, 2024, pp. 301-304, doi: 10.1109/ICWITE59797.2024.10502523.
4. Chen, Y.-P., Nishida, N., Nakayama, H., & Matsumoto, Y. (2022). *Recent Trends in Personalized Dialogue Generation: A Review of Datasets, Methodologies, and Evaluations*.
5. Verma, P., & Verma, A. (2022). *A Review on Text Summarization Techniques*.



A screenshot of a computer

Description automatically generated