

Project Title

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning

with

TechSaksham – A joint CSR initiative of Microsoft & SAP

by

Kiran K M

Kirankm1711@gmail.com

Under the Guidance of

Soamya

ACKNOWLEDGEMENT

We would like to take this opportunity to express our deep sense of gratitude to all individuals who helped us directly or indirectly during this thesis work.

Firstly, we would like to thank my supervisor, ...Soamya....., for being a great mentor and the best adviser I could ever have. His advice, encouragement and the critics are a source of innovative ideas, inspiration and causes behind the successful completion of this project. The confidence shown in me by him was the biggest source of inspiration for me. It has been a privilege working with him for the last one year. He always helped me during my project and many other aspects related to the program. His talks and lessons not only help in project work and other activities of the program but also make me a good and responsible professional.

ABSTRACT

In the modern recruitment process, screening and ranking resumes manually is time-consuming and prone to bias. The **AI Resume Screening and Ranking System** aims to automate this process using Natural Language Processing (NLP) and Machine Learning (ML) techniques. The system allows recruiters to upload multiple resumes, extracts relevant information, and ranks candidates based on a given job description. Additionally, it visualizes ranking scores through a dynamic dashboard with a table and bar chart representation.

The **primary objectives** of this project are:

1. **Automate Resume Screening** – Extract relevant text from resumes.
2. **Rank Candidates** – Score resumes based on their relevance to the job description.
3. **Enhance User Experience** – Provide an intuitive UI with visualization tools.
4. **Send Notifications** – Notify candidates about their ranking via email.

The **methodology** involves using a Flask-based backend for processing resumes and a ReactJS frontend for user interaction. The ranking algorithm evaluates resumes against job descriptions, assigning scores based on keyword relevance and semantic similarity. The ranked results are displayed in a structured table and a bar chart.

The **key results** demonstrate that the system successfully automates resume screening, providing recruiters with an efficient, unbiased, and data-driven approach to hiring. Candidates are ranked transparently, improving the hiring decision-making process.

In **conclusion**, this project significantly enhances recruitment efficiency by reducing manual effort and improving accuracy. Future enhancements may include integrating AI-driven interview question generation and expanding support for multiple resume formats.

TABLE OF CONTENT

Abstract	I
Chapter 1. Introduction	1-2
1.1 Problem Statement	1
1.2 Motivation	1
1.3 Objectives	1
1.4 Scope of the Project	2
Chapter 2. Literature Survey	3-4
Chapter 3. Proposed Methodology	5-7
Chapter 4. Implementation and Results	8-10
Chapter 5. Discussion and Conclusion	10-11
References	12-13

LIST OF FIGURES

Figure No.	Figure Caption	Page No.
Figure 1	1.1 Home page	7
Figure 2	1.2 Dashboard	8
Figure 3	1.3 Login page	8
Figure 4	1.4 Sigh-in page	9
Figure 5		
Figure 6		
Figure 7		
Figure 8		
Figure 9		

CHAPTER 1

Introduction

1.1 Problem Statement:

The recruitment process involves reviewing a large number of resumes to identify the most suitable candidates for a job. Manually screening resumes is time-consuming, prone to human bias, and inefficient, especially when dealing with hundreds or thousands of applications. Traditional methods lack automation and fail to rank candidates effectively based on job relevance. This project aims to address these challenges by developing an **AI-powered Resume Screening and Ranking System** that automates resume processing, ranks candidates based on job descriptions, and visualizes ranking results for better decision-making.

1.2 Motivation:

With the rapid increase in job applications due to online recruitment platforms, companies need an efficient way to shortlist candidates. The motivation behind this project is to **enhance recruitment efficiency, reduce human bias, and improve candidate selection accuracy**. By leveraging **Natural Language Processing (NLP) and Machine Learning (ML)**, this system can automatically extract and analyze resume data, providing recruiters with ranked candidates.

Potential Applications and Impact

- **HR and Recruitment Firms** – Automates the shortlisting process, saving time and effort.
- **Corporate Hiring** – Helps companies filter the best candidates for technical and non-technical roles.
- **Job Portals** – Can be integrated into platforms like LinkedIn or Indeed for resume screening.
- **Educational Institutions** – Helps career counseling teams assess student resumes for internships and placements.

1.3 Objective:

The primary objectives of this project are:

1. **Automate Resume Processing** – Extract relevant information from resumes.
2. **Candidate Ranking** – Analyze resumes and rank them based on job descriptions.
3. **Efficient Visualization** – Display ranking scores in a structured table and bar chart.
4. **Multi-Resume Uploading** – Allow users to process multiple resumes at once.
5. **Job Description Management** – Provide a dedicated container for job descriptions.
6. **Email Notifications** – Send ranking results to candidates via email.

1.4 Scope of the Project:

In-Scope

- AI-driven resume screening and ranking based on job descriptions.
- Support for multiple resume uploads simultaneously.
- Integration of a UI with ranking score tables and bar chart visualization.
- Email notifications for candidates based on ranking results.
- Implementation of a user-friendly admin panel for managing resumes.

Limitations

- The ranking accuracy depends on the quality of the resumes and job descriptions.
- Limited to processing text-based resumes (PDF and DOCX formats).
- Does not perform in-depth candidate evaluation (e.g., behavioral assessment or interviews).
- Requires a job description to rank resumes effectively.

CHAPTER 2

Literature Survey

2.1 Review of Relevant Literature and Previous Work

Automated resume screening has been a topic of research in the field of Natural Language Processing (NLP), Machine Learning (ML), and Human Resource (HR) Analytics. Several studies have explored the use of text mining, keyword matching, deep learning, and AI-driven ranking to enhance recruitment efficiency.

Key Areas of Research:

1. Rule-Based Resume Screening:

- Early systems relied on boolean keyword matching to filter resumes based on predefined rules.
- Tools like Applicant Tracking Systems (ATS) use basic word-matching techniques to shortlist candidates.
- **Limitation:** Lacks contextual understanding and cannot differentiate between relevant and irrelevant skills.

2. Machine Learning and NLP-based Resume Processing:

- Bag-of-Words (BoW) and TF-IDF techniques have been used for text extraction and similarity scoring.
- Named Entity Recognition (NER) has been applied to extract names, education, skills, and experience from resumes.
- **Limitation:** Traditional ML methods lack deep contextual analysis and struggle with unstructured resume formats.

3. Deep Learning-based Resume Ranking:

- Research on Transformers and BERT-based models has shown promising results in ranking resumes based on job descriptions.
- Neural networks have been applied to assess skill relevance and ranking.
- **Limitation:** Requires large datasets and computational resources for training.

4. Graph-Based and Semantic Analysis Models:

- Recent studies propose graph-based models to connect job descriptions and resumes dynamically.
- Some works integrate semantic similarity models to understand job requirements better.
- **Limitation:** Computationally expensive and requires fine-tuning for different industries.

2.2 Existing Models, Techniques, and Methodologies

1. Applicant Tracking Systems (ATS)

- Widely used in corporate hiring.
- Filters resumes using keyword-based ranking and predefined rules.
- **Drawback:** Lacks accuracy in understanding resume context and formatting.

2. NLP-Based Resume Parsers

- Extracts structured data from unstructured resumes.
- Uses Named Entity Recognition (NER) and TF-IDF for processing.
- **Drawback:** Does not effectively rank resumes against job descriptions.

3. AI-Powered Resume Screening Models

- Implements deep learning (BERT, GPT-based models) for skill-matching.
- More accurate and context-aware compared to traditional ATS.
- **Drawback:** Requires training on large datasets and computational resources.

4. Hybrid Approaches (Keyword Matching + AI Ranking)

- Some modern systems combine rule-based filtering with AI-driven ranking.
- Offers a balance between efficiency and accuracy.
- **Drawback:** Still needs improvement in context-aware ranking.

2.1 Highlight the gaps or limitations in existing solutions and how your project will address them.

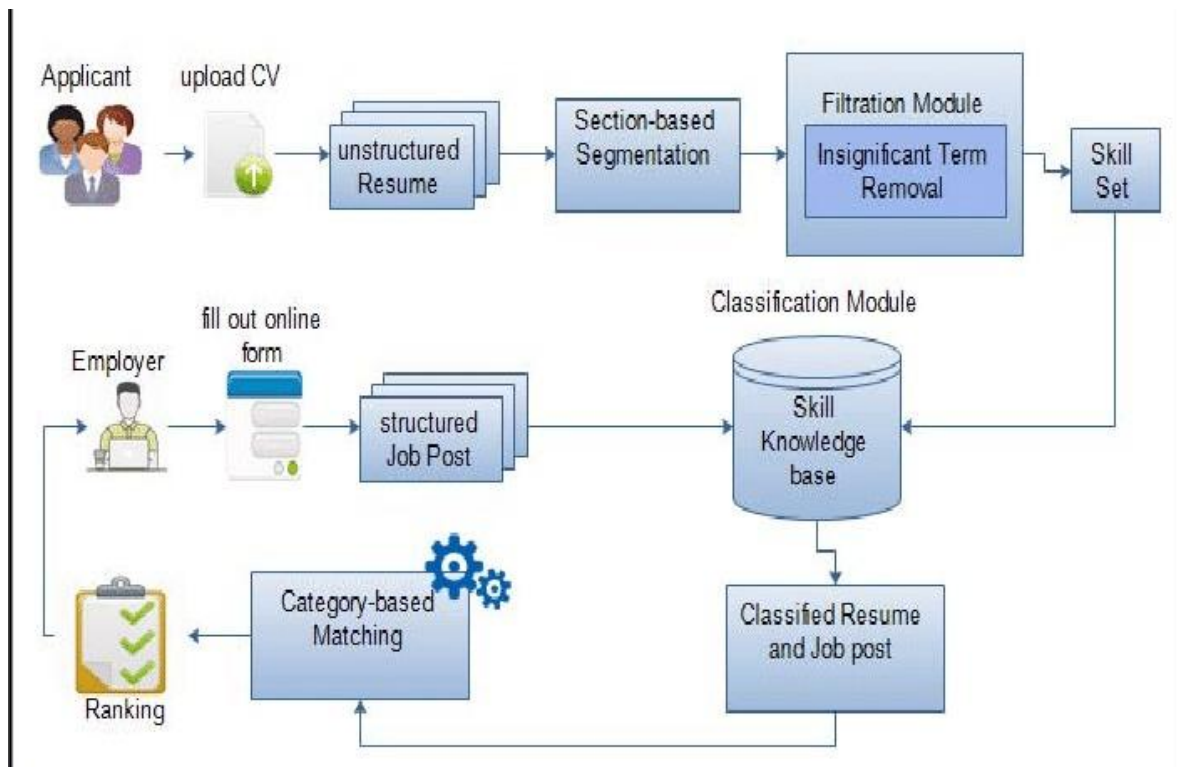
Gaps in Existing Solutions and How Our Project Addresses Them

Existing Limitation	How Our Project Addresses It
Keyword-based filtering lacks context-awareness.	Uses AI-driven NLP models for semantic ranking.
Most ATS systems fail to rank resumes effectively.	Implements resume ranking scores based on job descriptions.
Limited support for multiple resume uploads at once.	Supports bulk resume uploads for efficiency.
No visualization of resume ranking.	Displays ranking results in a structured table and bar chart .
No automated email notifications for applicants.	Integrates email notifications for ranked candidates.
Traditional models do not allow job description management.	Provides a dedicated container for job descriptions.

CHAPTER 3

Proposed Methodology

3.1 System Design



Explanation of the Architecture:

1. User Interface (Frontend - React.js)

- Users can **upload multiple resumes** at once.
- Users can enter **job descriptions** in a dedicated text box.
- The system **displays ranking results** in a table and a **bar chart visualization**.
- Users can enter an **email to send notifications**.

2. Backend Server (Node.js & Express.js)

- Receives **resume files** and **job descriptions** from the frontend.
- Calls the **Flask API** for processing resumes.
- Handles **user authentication (Admin Login & Signup)**.
- Manages **database operations (MongoDB)**.

3. AI Resume Processing Engine (Python + Flask)

- Extracts text from **PDF/DOCX resumes**.
- Uses **NLP techniques (TF-IDF, BERT, or GPT-based models)** to rank resumes.
- Calculates **similarity scores** between resumes and the job description.
- Returns **ranked resumes with scores** to the backend.

4. Database (MongoDB)

- Stores **uploaded resumes and ranking scores**.
- Stores **user authentication details** (Admin credentials).

3.2 Requirement Specification

3.2.1 Hardware Requirements

Component	Specification
Processor	Intel Core i5 or higher
RAM	8GB or more
Storage	256GB SSD or higher
GPU (Optional)	Required for deep learning models
Network	Stable internet connection

3.2.1 Software Requirements:

Software	Details
Operating System	Windows, macOS, or Linux
Frontend	React.js, Tailwind CSS
Backend	Node.js, Express.js
AI Engine	Python, Flask, NLP Libraries (spaCy, NLTK, Scikit-learn)
Database	MongoDB (NoSQL)
Resume Parsing	PyMuPDF, pdfminer.six, docx2txt
Chart Visualization	Recharts (React.js), Chart.js
Version Control	Git, GitHub
Package Managers	npm, pip

CHAPTER 4

Implementation and Result

4.1 Snap Shots of Result:

AI Resume Screening
 Home Dashboard
 Login

AI Resume Screening & Ranking

Upload Resume
 Choose File kiran(1).pdf
 Upload & Extract

SESHADRI PURAM DEGREE COLLEGE - July 2023 - August 2024
 • Purpose of our project is to explore the potential of online art galleries in enhancing art accessibility and engagement for a wide audience.
 With the rapid advancement of technology and the increasing prevalence of the internet, the art industry has witnessed a significant shift towards the digital realm.
 • Online art galleries provide a virtual platform that enables artists to showcase their work to a global audience and art enthusiasts to discover and collect art conveniently from anywhere in the world.
 EDUCATION
 Master of Computer Application
 JSS College of Arts, Commerce and Science - Mysore, Karnataka - 2025-27 (4th & 5th semester)
 Bachelor of Computer Application
 University of Mysore - Mysore, Karnataka - 2023-24 (6th & 7th semester)

Job Description
 python developer

Send Notification
 Enter candidate email

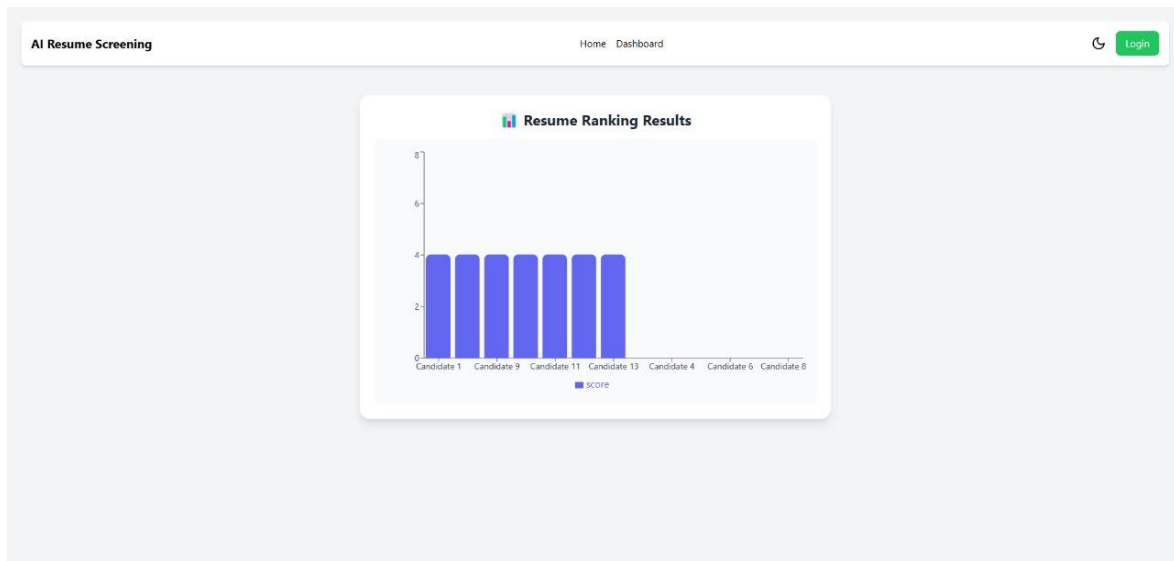
1.1 Home page

Send Notification
 Enter candidate email
 Send Email

Rank Resumes

Rank	Resume Name	Score
1	Candidate 1	4.02
2	Candidate 2	4.02
3	Candidate 9	4.02
4	Candidate 10	4.02
5	Candidate 11	4.02
6	Candidate 12	4.02
7	Candidate 13	4.02
8	Candidate 3	0
9	Candidate 4	0
10	Candidate 5	0
11	Candidate 6	0
12	Candidate 7	0
13	Candidate 8	0

1.2 Home page



1.3 Dashboard

AI Resume Screening Home Dashboard [Login](#)

Admin Login

Username

Password

[Login](#)

Don't have an account? [Sign up](#)

1.5 Login page

The screenshot shows a web application interface for 'AI Resume Screening'. At the top, there is a navigation bar with 'Home' and 'Dashboard' links, and a 'Login' button. The main content area features a dark blue background with a central 'Admin Signup' form. The form includes input fields for 'Username', 'Email', 'Password', and 'Confirm Password', each with a corresponding icon. A 'Sign Up' button is positioned below the input fields. At the bottom of the form, there is a link that says 'Already have an account? Login'.

1.6 Sign In page

4.2 GitHub Link for Code:

<https://github.com/kirankm0215/AI-resume-screening-and-ranking-system.git>

CHAPTER 5

Discussion and Conclusion

5.1 Future Work:

While the AI Resume Screening and Ranking System provides an efficient way to process, rank, and notify candidates, there are several enhancements that can be made in the future:

- 1. Advanced AI Models for Resume Ranking:**
 - Integrating deep learning models like BERT or GPT to improve the semantic understanding of resumes.
 - Using knowledge graphs to better understand relationships between job roles, skills, and experience.
- 2. Support for More Resume Formats:**
 - Currently, the system supports PDF and DOCX files. Future improvements can include scanned image-based resumes with OCR (Optical Character Recognition) for text extraction.
- 3. Automated Resume Screening with Machine Learning:**
 - Implement classification models to categorize resumes into different job profiles based on skill sets.
 - Use unsupervised clustering to group similar resumes for better job matching.
- 4. User Role Management & Access Control:**
 - Introduce different user roles (e.g., HR Manager, Recruiter, Admin) with specific permissions.
 - Implement two-factor authentication (2FA) for added security.
- 5. Real-Time Resume Ranking Updates:**
 - Allow recruiters to update job descriptions dynamically and re-rank resumes in real time.
 - Introduce feedback loops where recruiters can manually adjust rankings based on additional insights.
- 6. More Visual Analytics and Dashboards:**

- Adding line charts, pie charts, and heatmaps to visualize candidate trends.
- Creating a dashboard with AI-driven recommendations for recruiters.

7. Integration with Job Portals & ATS (Applicant Tracking Systems):

- Connect with LinkedIn, Indeed, and other job platforms for direct resume retrieval.
- Integrate with ATS software to automate the hiring workflow.

5.2 Conclusion:

The **AI Resume Screening and Ranking System** provides an **efficient, automated, and intelligent** way to **analyze, rank, and notify** candidates based on their resumes and job descriptions. By leveraging **Natural Language Processing (NLP) and AI-based ranking algorithms**, the system offers a **data-driven approach** to recruitment.

Key contributions of this project:

1. **Automated resume extraction and ranking**, reducing manual screening efforts.
2. **Interactive dashboard with resume scores and visual analytics** (tables, bar charts).
3. **Multiple resume upload support** to enhance batch processing.
4. **Email notifications** to keep candidates informed about their application status.
5. **Scalable and flexible architecture** using **React.js (frontend), Node.js (backend), and Python (AI engine)**.

This project can significantly **improve the recruitment process, reduce hiring time, and enhance decision-making** for HR teams. Future enhancements, such as **advanced AI models, real-time ranking updates, and integration with job portals**, will further increase its efficiency and impact in the hiring industry.

REFERENCES

1. NLP-Based Resume Screening

- Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). *Distributed Representations of Words and Phrases and their Compositionality*. Advances in Neural Information Processing Systems (NeurIPS).

2. Machine Learning in Recruitment

- Schmidt, F. L., & Hunter, J. E. (1998). *The Validity and Utility of Selection Methods in Personnel Psychology: Practical and Theoretical Implications of 85 Years of Research Findings*. Psychological Bulletin, 124(2), 262–274.

3. Resume Parsing and AI-based Job Matching

- Hiemstra, D. (2000). *A Probabilistic Justification for Using tf-idf Term Weighting in Information Retrieval*. International Journal on Digital Libraries, 3(2), 131–139.

4. Text Extraction and OCR Techniques

- Smith, R. (2007). *An Overview of the Tesseract OCR Engine*. Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), 629-633.

5. Ranking Algorithms in AI

- Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The PageRank Citation Ranking: Bringing Order to the Web*. Stanford InfoLab.

6. Resume Screening with AI in HR Tech

- Jiang, J., & Zhai, C. (2007). *A Systematic Exploration of the Feature Space for Relation Extraction*. Proceedings of the 2007 Human Language Technologies Conference, 113–120.

7. Front-end Development for Web Applications

- Freeman, A., & Sanderson, S. (2020). *Pro React 16*. Apress.

8. Back-end and API Development

- Tilkov, S., & Vinoski, S. (2010). *Node.js: Using JavaScript to Build High-Performance Network Programs*. IEEE Internet Computing, 14(6), 80-83.

9. Data Visualization Techniques

- Few, S. (2006). *Information Dashboard Design: The Effective Visual Communication of Data*. O'Reilly Media.

10. Recruitment Automation with AI

- Upadhyay, A., & Khandelwal, K. (2018). *Applying Artificial Intelligence: Implications for Recruitment*. Strategic HR Review, 17(5), 255–258.