**Project Title**

**AI-powered Resume Screening and Ranking System**

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

**Name of Student, Email id**

**Name of Student: Ampolu Raj Kumar**

**Email id: ampolurajkumar96@gmail.com**

Under the Guidance of

**Name of Guide**

**Saomya choudhar**

**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, … **Saomya choudhar…………..,** 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.

…We would also like to thank AICTE, Microsoft, and SAP for providing us with this wonderful opportunity to work on such an innovative project. …..

#### This Acknowledgement should be written by students in your own language (Do not copy and Paste)

#### …..

……

….

……

#### **ABSTRACT**

The **AI-powered Resume Screening and Ranking System** is designed to automate and streamline the recruitment process by leveraging artificial intelligence and machine learning techniques. The system aims to address the challenges faced by recruiters in manually screening and ranking resumes, which is time-consuming and prone to human bias. The project focuses on developing a system that can efficiently parse resumes, extract relevant information, and rank candidates based on their skills, experience, and job requirements.

The key objectives of the project include:

1. Developing an AI model to extract and analyze information from resumes.
2. Implementing a ranking algorithm to prioritize candidates based on job requirements.
3. Creating a user-friendly interface for recruiters to interact with the system.

The proposed methodology involves the use of Natural Language Processing (NLP) techniques for text extraction, machine learning models for candidate ranking, and a web-based interface for user interaction. The system was implemented using Python, with libraries such as SpaCy, Scikit-learn, and Flask for backend development. The results demonstrate that the system can significantly reduce the time taken for resume screening while improving the accuracy and fairness of the recruitment process.

In conclusion, the AI-powered Resume Screening and Ranking System has the potential to revolutionize the recruitment industry by automating repetitive tasks and enabling recruiters to focus on more strategic aspects of hiring.

**TABLE OF CONTENT**

**Abstract I**

**Chapter 1.**  **Introduction 1**

1.1 Problem Statement 1

1.2 Motivation 1

1.3 Objectives 2

1.4. Scope of the Project 2

**Chapter 2.**  **Literature Survey 3**

**Chapter 3.**  **Proposed Methodology**

**Chapter 4.**  **Implementation and Results**

**Chapter 5. Discussion and Conclusion**

**References**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Caption** | **Page No.** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table. No.** | **Table Caption** | **Page No.** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**CHAPTER 1**

**Introduction**

* 1. **Problem Statement:**

The recruitment process is often time-consuming and labor-intensive, especially when it comes to screening and ranking resumes. Recruiters have to manually go through hundreds of resumes to shortlist candidates, which is prone to human bias and errors. This project aims to address this problem by developing an AI-powered system that can automate the resume screening and ranking process.

* 1. **Motivation:**

The motivation behind this project is to reduce the time and effort required for resume screening while improving the accuracy and fairness of the recruitment process. By leveraging AI and machine learning, the system can help recruiters focus on more strategic tasks, such as interviewing and candidate engagement, rather than spending hours on manual resume screening

* 1. **Objective:**
* To develop an AI model that can extract and analyze information from resumes.
* To implement a ranking algorithm that prioritizes candidates based on job requirements.
* To create a user-friendly interface for recruiters to interact with the system.
  1. **Scope of the Project:**

The project focuses on developing a system that can parse resumes in PDF and Word formats, extract relevant information such as skills, experience, and education, and rank candidates based on their suitability for a given job role. The system is designed to be scalable and can be integrated with existing recruitment software.

**CHAPTER 2**

**Literature Survey**

* 1. **Review relevant literature or previous work in this domain.**

**Review of Relevant Literature:**

Several studies have been conducted on resume parsing and candidate ranking using AI and machine learning. Existing solutions often rely on keyword matching and rule-based systems, which have limitations in handling diverse resume formats and extracting contextual information. Recent advancements in NLP and deep learning have enabled more accurate and efficient resume parsing techniques.

* 1. **Mention any existing models, techniques, or methodologies related to the problem.**
* Keyword-based Matching: Traditional methods that rely on keyword matching to rank resumes.
* Machine Learning Models: Supervised learning models that use labeled data to predict candidate suitability.
* Deep Learning Approaches: Neural networks that can extract complex patterns from resumes**.**
  1. **Highlight the gaps or limitations in existing solutions and how your project will address them.**

**Gaps in Existing Solutions:**

* Limited ability to handle diverse resume formats.
* Lack of contextual understanding in keyword-based systems.
* Difficulty in ranking candidates based on soft skills and experience

**CHAPTER 3**

**Proposed Methodology**

* 1. **System Design**

Provide the diagram of your Proposed Solution and explain the diagram in detail.

The system architecture consists of three main components:

* Resume Parser: Extracts information from resumes using NLP techniques.
* Ranking Algorithm: Ranks candidates based on their skills, experience, and job requirements.
* User Interface: A web-based interface for recruiters to upload resumes and view ranked candidates**.**

**Requirement Specification**

Mention the tools and technologies required to implement the solution.

* Hardware Requirements:
* Processor: Intel i5 or higher
* RAM: 8GB or higher
* Storage: 500GB HDD or SSD
* Software Requirement
* Python 3.7 or higher
* Libraries: SpaCy, Scikit-learn, Flask
  + 1. **Hardware Requirements:**

Processing Power (CPU/GPU)

* CPU: A multi-core processor (e.g., Intel Xeon, AMD Ryzen) with high clock speed is essential for handling general-purpose tasks, data preprocessing, and running lighter machine learning models.
* GPU: For deep learning models, especially those involving natural language processing (NLP) and large-scale data processing, a high-performance GPU (e.g., NVIDIA Tesla, A100, or RTX 3090) is recommended. GPUs accelerate the training and inference of AI models.

2. Memory (RAM)

* Minimum: 16 GB RAM for small-scale systems or prototyping.
* Recommended: 32 GB to 128 GB or more for handling large datasets and running complex AI models efficiently.

3. Storage

* SSD: Fast storage (NVMe SSDs) is recommended for quick data access and processing.
* Capacity:
  + At least 500 GB for small-scale systems.
  + 1 TB or more for larger systems with extensive resume databases and model storage.
* Scalability: Consider cloud-based storage solutions (e.g., AWS S3, Google Cloud Storage) for scalability and backup.

4. Networking

* High-Speed Internet: Essential for cloud-based systems or hybrid setups.
* Bandwidth: Sufficient bandwidth to handle data transfer, especially if the system is integrated with external APIs or cloud services.

5. Cloud Infrastructure (Optional)

* If the system is cloud-based, consider using cloud services like AWS, Google Cloud, or Azure. These platforms provide scalable computing resources, including virtual machines, GPUs, and storage.
* Serverless Computing: For cost efficiency, serverless options like AWS Lambda or Google Cloud Functions can be used for specific tasks.

6. Peripheral Devices

* High-Resolution Monitor: For developers and administrators to work efficiently.
* Input Devices: Standard keyboard and mouse.

7. Backup and Redundancy

* Backup Solutions: Regular backups of the database and AI models to prevent data loss.
* Redundant Power Supply: For on-premise systems, ensure uninterrupted power supply (UPS) to avoid downtime.

8. Scalability Considerations

* Horizontal Scaling: Ability to add more machines or nodes to handle increased load.
* Vertical Scaling: Ability to upgrade existing hardware (e.g., adding more RAM or GPUs).

Example Configurations:

* Small-Scale System:
  + CPU: Intel Core i7 or AMD Ryzen 7
  + GPU: NVIDIA GTX 1660 or equivalent
  + RAM: 16 GB
  + Storage: 500 GB SSD
* Large-Scale System:
  + CPU: Dual Intel Xeon or AMD EPYC
  + GPU: NVIDIA A100 or equivalent
  + RAM: 128 GB or more
  + Storage: 2 TB NVMe SSD + Cloud Storage
    1. **Software Requirements:**

1.Operating System

* Options:
  + Linux: Ubuntu, CentOS, or other distributions (preferred for AI/ML development due to compatibility with most frameworks).
  + Windows: Windows 10/11 or Windows Server (for .NET-based systems or user-friendly environments).
  + macOS: For development purposes (if preferred by developers).
* Considerations: Choose based on compatibility with AI frameworks, ease of deployment, and team expertise.

2. AI/ML Frameworks and Libraries

* Deep Learning Frameworks:
  + TensorFlow
  + PyTorch
  + Keras
* Natural Language Processing (NLP) Libraries:
  + Hugging Face Transformers
  + spaCy
  + NLTK
  + Gensim
* Machine Learning Libraries:
  + Scikit-learn
  + XGBoost
  + LightGBM
* Computer Vision Libraries (if needed for parsing resumes with images):
  + OpenCV
  + PIL (Python Imaging Library)

3. Programming Languages

* Primary Language:
  + Python (most commonly used for AI/ML development due to extensive libraries and community support).
* Secondary Languages (optional):
  + JavaScript/Node.js (for backend or frontend development).
  + Java or C# (for enterprise-level integrations).

4. Database Management System (DBMS)

* Relational Databases:
  + PostgreSQL
  + MySQL
  + Microsoft SQL Server
* NoSQL Databases (for unstructured data):
  + MongoDB
  + Elasticsearch (for text-based search and indexing).
* Cloud-Based Solutions:
  + AWS DynamoDB
  + Google Firestore

5. Development Tools

* Integrated Development Environment (IDE):
  + PyCharm
  + Visual Studio Code
  + Jupyter Notebook (for prototyping and experimentation).
* Version Control:
  + Git (with platforms like GitHub, GitLab, or Bitbucket).
* Containerization:
  + Docker (for creating consistent development and deployment environments).
* Orchestration:
  + Kubernetes (for managing containerized applications at scale).

6. Cloud Services (Optional)

* AI/ML Platforms:
  + Google Cloud AI Platform
  + AWS SageMaker
  + Azure Machine Learning
* Storage:
  + AWS S3
  + Google Cloud Storage
  + Azure Blob Storage
* Serverless Computing:
  + AWS Lambda
  + Google Cloud Functions
  + Azure Functions

7. APIs and Integration Tools

* Resume Parsing APIs:
  + Hireability
  + Sovren
  + Affinda
* Third-Party Integrations:
  + LinkedIn API (for fetching candidate profiles).
  + Job Board APIs (for integrating with platforms like Indeed or Glassdoor).
* Authentication and Authorization:
  + OAuth 2.0
  + JWT (JSON Web Tokens)

8. Web Development Frameworks (for UI/UX)

* Frontend:
  + React.js
  + Angular
  + Vue.js
* Backend:
  + Django (Python)
  + Flask (Python)
  + Node.js (JavaScript)

9. Monitoring and Logging Tools

* Monitoring:
  + Prometheus
  + Grafana
* Logging:
  + ELK Stack (Elasticsearch, Logstash, Kibana)
  + Splunk

10. Security Software

* Data Encryption:
  + SSL/TLS for secure data transmission.
  + AES for data encryption at rest.
* Authentication:
  + Multi-factor authentication (MFA) tools.
* Vulnerability Scanning:
  + Tools like Nessus or OpenVAS.

11. Backup and Recovery Tools

* Backup Solutions:
  + Veeam
  + Acronis
  + Cloud-based backups (AWS Backup, Google Cloud Backup).

12. Collaboration and Project Management Tools

* Collaboration:
  + Slack
  + Microsoft Teams
* Project Management:
  + Jira
  + Trello
  + Asana

**CHAPTER 4**

**Implementation and Result**

* 1. **Snap Shots of Result:**
* Snapshot 1: Resume parsing output showing extracted skills, experience, and education.
* Snapshot 2: Ranking algorithm output showing the top 5 candidates for a given job role.
* Snapshot 3: User interface screenshot showing the upload and ranking features.
  1. **GitHub Link for Code:**

**CHAPTER 5**

**Discussion and Conclusion**

* 1. **Future Work:**

Provide suggestions for improving the model or addressing any unresolved issues in future work.

* + Improve the system's ability to handle more diverse resume formats.
  + Incorporate sentiment analysis to evaluate soft skills from resumes.
  + Integrate the system with popular recruitment platforms like LinkedIn and Naukri.
  1. **Conclusion:**

The AI-powered Resume Screening and Ranking System successfully automates the resume screening process, reducing the time and effort required by recruiters. The system's ability to accurately parse and rank resumes makes it a valuable tool for modern recruitment processes. Future work will focus on enhancing the system's capabilities and integrating it with existing recruitment platforms.

**REFERENCES**

1. Ming-Hsuan Yang, David J. Kriegman, Narendra Ahuja, “Detecting Faces in Images: A Survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume. 24, No. 1, 2002.