**1. Aim**

The primary aim of this project is to develop an automated system that ranks resumes based on job descriptions using Large Language Models (LLMs). The system helps recruiters efficiently screen candidates by automating resume analysis, thereby reducing manual effort, increasing accuracy, and minimizing biases in the recruitment process.

**2. Introduction**

The recruitment process often involves handling a large number of resumes, making manual screening time-consuming and error-prone. This project introduces a solution to automate resume ranking using LLMs. By analyzing resumes and comparing them with job descriptions, the system can quickly rank candidates, providing recruiters with a more efficient way to manage applications and make data-driven decisions.

**3. Literature Survey**

* **Resume Analyzer: An Automated Solution to Recruitment**  
  Automates resume analysis using keyword matching to extract qualifications and experience, improving the speed and efficiency of interview scheduling.
* **Machine Learning-Based Resume Ranking System**  
  Uses Gensim and KNN to match resumes with job descriptions, improving precision in candidate selection.
* **Resume Classification Using NLP and ML**  
  Employs TF-IDF vectorization for classifying resumes by determining word relevance to job descriptions, improving matching accuracy.
* **NLP-Based Resume Parser Analysis**  
  Automates resume parsing and ranking, extracting key candidate details like skills and experience, ideal for campus recruitment.
* **Online Resume Parsing Using Text Analytics**  
  Uses text analytics to extract and rank candidates based on their expertise, particularly in technical fields such as programming.
* **Resume Parser Using Machine Learning and NLP**  
  Analyzes resumes with keyword matching, automatically scheduling interviews to streamline the hiring process.

**4. Motivation and Challenges**

**Motivation**:  
Recruitment often involves sifting through hundreds or even thousands of resumes. This manual process is slow, prone to biases, and can result in missing qualified candidates. Automating resume ranking with LLMs solves this problem by making the process faster, more objective, and more scalable.

**Challenges**:

* Integrating LLMs for accurate resume analysis.
* Ensuring scalability to handle large volumes of resumes and job descriptions.
* Managing data privacy and security when dealing with sensitive personal information.

**5. Objectives and Scope of the Project**

**Objectives**:

* Automate the ranking of resumes by comparing them with job descriptions.
* Provide recruiters with a ranked list of candidates based on relevance.
* Offer personalized recommendations for users on skills and courses to improve their resumes.

**Scope**: This project targets mid-to-large-scale organizations that handle large volumes of job applications. The system is designed to be scalable, adaptable to various industries, and user-friendly for both recruiters and applicants.

**6. Proposed Methodology (System Architecture and Module Details)**

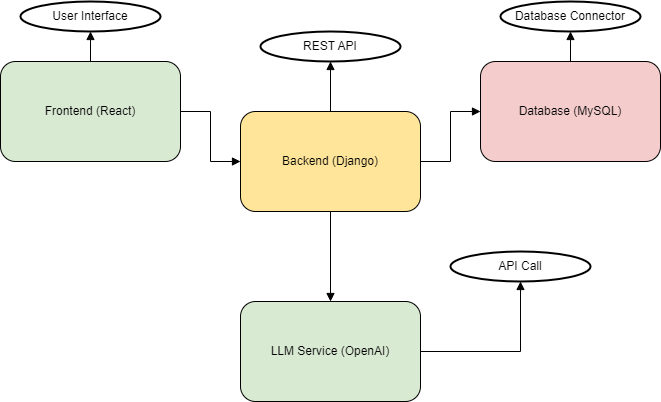
**System Architecture**:

* **Frontend**: A user-friendly web interface where users upload resumes and recruiters manage job vacancies.
* **Backend**: Django-based server handling user requests, job postings, and integrating with LLMs for resume analysis.
* **LLM API**: Used to extract key information from resumes and match them with job descriptions.

**Modules**:

1. **User Registration & Login**: Handles authentication for recruiters and job applicants.
2. **Job Vacancy Creation**: Allows recruiters to create job posts and receive resumes.
3. **Resume Upload & Analysis**: Applicants upload resumes, which are analyzed using LLMs.
4. **Recommendation Engine**: Suggests skills and courses to applicants based on their resumes.
5. **Ranked List Generation**: Ranks applicants based on resume-job relevance.

**Block Diagram**:



**7. Software and Hardware Requirements**

**Software Requirements**:

| **Sr. No.** | **Component** | **Specifications** |
| --- | --- | --- |
| 1 | Operating System | Ubuntu 20.04 / Windows Server 2019 |
| 2 | Technology | Django (Backend), React (Frontend) |
| 3 | Tool | OpenAI API (LLM Integration) |
| 4 | Database | MySQL |
| 5 | Server | AWS / Google Cloud for deployment |

**Hardware Requirements**:

| **Sr. No.** | **Component** | **Specifications** |
| --- | --- | --- |
| 1 | Processor | Quad-Core (Intel i5 or higher) |
| 2 | Memory | 16 GB RAM |
| 3 | Network Interface | High-speed internet |
| 4 | Other | SSD storage (500 GB) |

**8. Contribution of the Project to the Field of Technology and Society**

This project brings several contributions to the recruitment industry and society:

* **Efficiency**: Automates the resume screening process, saving recruiters time and effort.
* **Bias Reduction**: Reduces human biases in candidate selection by leveraging objective LLM analysis.
* **Scalability**: Handles large volumes of applications, making it suitable for organizations of all sizes.
* **Guidance for Job Seekers**: Helps applicants improve their profiles by providing personalized skill and course recommendations.
* **Advanced Recruitment Tools**: Lays the groundwork for future AI-driven recruitment innovations, such as automated interview scheduling and applicant tracking systems.

**9. Conclusion**

This project successfully automates the resume ranking process using LLMs, providing a scalable, efficient, and objective solution to the challenges of modern recruitment. By implementing LLMs for resume analysis, the project improves accuracy in candidate selection while reducing manual effort and biases. Future work can expand the system's capabilities to include interview scheduling, deeper applicant analysis, and integrations with industry-specific recruitment tools.

**10. References**

1. Ankita Vaidya, "Resume Analyzer: An Automated Solution to Recruitment," Vol. 03, Issue 08, August 2020.
2. Tejaswini K, et al., "Machine Learning-Based Resume Ranking System," Vol. 03, Issue 2, November 2022.
3. Irfan Ali, et al., "Resume Classification Using NLP and ML," Vol. 41, Issue 1, January 2022.
4. Mhaske Harshada, "NLP-Based Resume Parser Analysis," Vol. 10, Issue 5, May 2023.
5. Divyanshu Chandola, "Online Resume Parsing Using Text Analytics," Vol. 09, Issue 01, July 2015.
6. Pratham Rasal, "Resume Parser Using Machine Learning and NLP," Vol. 11, Issue 5, May 2023.