**QuickRecruit: AI-Driven System for Automated Resume Ranking and Hiring**

**1. Introduction**

Companies are flooded with resumes for every job opening they fill in today’s fast-paced hiring environment. Screening resumes by hand takes enormous time, and is prone to bias and inconsistency. **QuickRecruit** is an AI resume analysis system that aims to improve recruitment processes by making intelligent, automated candidate profile-job description matching. Using machine learning/ML, it compares the applicants resume to job requirements to help recruiters best understand applicants in a data-driven and objective manner. The challenge we face is high-volume recruitment and this project proposes an automated and accurate response to high-volume recruitment while aiming to provide a high-confidence prediction about candidate-job fit and a significantly faster recruitment pipeline.

**2. Problem Statement**

Recruiters face several challenges when dealing with large volumes of resumes:

* Time-consuming manual screening
* Inconsistencies and bias in evaluation
* Difficulty in identifying the most suitable candidates based on job descriptions
* Lack of standardized scoring mechanisms

These issues lead to inefficiencies in hiring processes and potential loss of top talent. There is a clear need for an automated, intelligent system that can analyse resumes quickly and fairly.

**3. Objectives**

* To develop a web-based application that allows uploading of resumes and job descriptions.
* To use AI techniques for extracting and comparing key resume information with job requirements.
* To provide a match score and analysis report for each candidate.
* To enhance hiring accuracy, speed, and fairness.

**4. Methodology**

**a. Requirement Identification**

**i. Study of Existing System / Literature Review**

* Analysis of existing ATS (Applicant Tracking Systems)
* Review of research papers on resume parsing and NLP techniques
* Study of machine learning algorithms used for document similarity

Further references include:

* Empirical comparisons between cosine similarity and BERT-based encodings
* Performance evaluation of parsing accuracy across various resume formats
* Psychological studies on bias reduction in automated candidate screening

**ii. Requirement Analysis**

**Functional Requirements:**

* Resume parsing
* Job description input
* Scoring algorithm
* User interface
* Result dashboard

**Non-Functional Requirements:**

* Scalability
* Accuracy
* Usability
* Security

**b. Feasibility Study**

**i. Technical**

* Feasible using Python-based ML libraries (e.g., spaCy, scikit-learn, BERT models)
* Frontend with Flutter or React
* Backend using Node.js or Django
* Cloud deployment via AWS or Firebase

**ii. Operational**

* Easy to integrate into existing HR systems
* User-friendly interface for non-technical users

**iii. Economic**

* Open-source libraries reduce software costs
* Hosting on cloud platforms with minimal infrastructure investment

**iv. Schedule (Gantt Chart)**

* Week 1-2: Requirement Gathering & Literature Review
* Week 3-4: Design of System Architecture
* Week 5-7: Development of Resume Parser & Job Matcher
* Week 8-9: Frontend and Backend Integration
* Week 10: Testing and Debugging
* Week 11: Deployment and Documentation

**c. High Level Design of System**

**System Flow:**

1. Upload Resume and Job Description
2. Parse and Extract Key Features
3. Compare Resume Features to Job Requirements
4. Generate Match Score and Analysis Report

**Algorithm Description:**

* Use of NLP for keyword extraction and sentence similarity
* Cosine similarity or transformer-based embeddings for semantic comparison
* Scoring mechanism based on skill match, experience relevance, and education fit

Additional steps will include:

* Static testing of similarity models across controlled data sets
* Dynamic performance evaluation using progressive resume-job matching scenarios

**5. Expected Outcome**

* A working prototype of Matchfolio capable of analysing and scoring resumes
* Enhanced recruiter productivity and decision-making
* Reduced bias and improved fairness in hiring
* Scalable system ready for integration with enterprise platforms
* An empirical report comparing parsing precision and matching accuracy of different models (e.g., TF-IDF, BERT, USE)

**6. References**

* Jurafsky, D., & Martin, J. H. (2021). *Speech and Language Processing*
* Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*
* GitHub repositories on resume parsers and NLP models
* Research articles from IEEE Xplore and Springer on AI in recruitment
* Official documentation of BERT, spaCy, and scikit-learn
* Abbas, A., et al. (2019). Behavior of reinforced concrete sandwiched panels under blast load — for methodological inspiration
* Cervenka, V., et al. (2002). ATENA — A tool for engineering analysis of fracture in concrete — relevant to simulation fidelity