Volume 10, Issue 2, February – 2025 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14987553

Smart Hiring System

Srinidhi K S[1]; Nischal V U[2]; U Karthik[3]; Manoj B P[4];

[1,2,3,4]SJB Institute of Technology Bangalore, India

Publication Date: 2025/03/11

Abstract: Selecting the ideal candidate is often a challenging endeavor time consuming, inefficient, and often influenced by biases. The Smart Hiring System is designed to change that. This platform streamlines recruitment by automating the most tedious parts of the process, from resume screening to candidate evaluations. Using machine learning, the system scans resume, picks out key qualifications, and matches them with job requirements ensuring the best-fit candidates move forward. The hiring journey is structured to be fair and efficient. Candidates first take an MCQ-based assessment, followed by a coding round and a descriptive evaluation, allowing recruiters to assess their real-world problem-solving skills. A standout feature is real-time speech-to-text transcription during HR interviews, making conversations smoother and evaluations more precise. To ensure fairness, bias mitigation techniques are integrated, reducing the impact of unconscious prejudices in hiring decisions. Secure email-based authentication safeguards user data, ensuring privacy and trust. By taking over repetitive tasks, the Smart Hiring System lets recruiters focus on what truly matters finding the right talent. The automated workflow not only reduces hiring time but also improves accuracy in matching candidates with roles. This research demonstrates how technology can transform hiring, making it faster, fairer, and more effective. In today's competitive job market, a system like this isn't just a convenience it's a game changer for both recruiters and job seekers.

Keywords: *Resume Parsing, Candidate Matching, Skill Extraction, MCQ Assessment, Technical Evaluation, Coding Test, HR Interview, Speech-to-Text Processing, Bias Mitigation, Secure Authentication, Automated Hiring, Job Recommendation, Recruitment Platform, Performance Evaluation, Data-Driven Hiring, Interview Transcription, Job Listings Management, Candidate Assessment.*

How to Cite: Srinidhi K S; Nischal V U; U Karthik; Manoj B P; (2025). Smart Hiring System. *International Journal of Innovative Science and Research Technology,* 10(2), 1969-1976. https://doi.org/ 10.5281/zenodo.14987553

I. INTRODUCTION

In today's competitive job market, finding the right candidate isn't just a numbers game—it's a challenge that can feel overwhelming. Many organizations find themselves buried under a pile of resumes, running the same repetitive tests, and conducting interviews that sometimes let personal biases seep in. These hurdles not only slow down the process but also make it harder to spot the perfect fit. The Smart Hiring System was developed to change all that by automating and fine-tuning every step of the hiring journey.

Think of it as a trusted partner that handles the heavy lifting. The Smart Hiring System simplifies recruitment by using smart technology to boost both speed and fairness. At its core, the system automatically parses resumes, pulling out essential details like qualifications, work experience, and key skills. This crucial information is then used to match candidates with job openings that suit their profiles, ensuring that only the most fitting applicants move forward. This approach not only relieves HR teams from tedious manual screening but also accelerates the entire process.

Once potential candidates are identified, they move through a carefully structured evaluation process. They begin with an MCQ-based assessment to gauge their theoretical understanding, followed by a technical evaluation that includes coding challenges and descriptive questions to showcase their problem-solving abilities. One of the standout features of the system is its real-time speech-to-text capability during HR interviews. This tool captures every word accurately, ensuring that nothing important is missed during discussions.

Security and fairness are at the heart of the system. It uses secure, email-based authentication to safeguard sensitive information, while built-in bias mitigation techniques help ensure that evaluations remain objective and fair. By automating repetitive tasks and relying on data-driven insights, the Smart Hiring System lets recruiters focus on what truly matters—finding the right talent.

Overall, the Smart Hiring System transforms the hiring process into a more efficient, human-centric experience, helping organizations make quicker, more precise hiring decisions and build stronger, more dynamic teams.

II. LITERATURE SURVEY

In recent years, the focus of research in sentiment analysis has been on developing specific models for tasks such as emotion detection, sentiment classification, opinion mining, and context-aware sentiment prediction. Though this project integrates all these models into a single interface,

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14987553

which had not been developed earlier, it offers a more comprehensive approach to sentiment analysis. Some of the existing works in this field have been described below:

Kinger et al., [1] introduced a novel method that marries visual and textual analysis to enhance resume evaluation. The approach utilizes YOLOv5 for the visual parsing of resumes, effectively identifying and extracting structured elements such as layout components, headings, and sections. In parallel, DistilBERT processes the textual content to extract key details like skills, qualifications, and work experiences. By combining these two powerful techniques, the system can automatically rank candidates based on their relevance to specific job profiles. The researchers employed a diverse dataset of 500 resumes with varying formats, which revealed challenges such as high computational requirements and limited adaptability across industries. Future research is directed toward optimizing resource usage and refining ranking algorithms to improve generalization across different sectors.

Chenguang Gan et al., [2] proposed a framework that leverages Large Language Models (LLMs) to automate and enhance the resume screening process. Their method dissects resumes into critical sections—skills, experience, education—and applies classification techniques to convert unstructured data into actionable insights for recruiters. The framework uses annotated datasets containing thousands of resumes and over 78,000 sentences, demonstrating its effectiveness in handling complex linguistic structures. However, the study also highlights challenges such as data sensitivity, privacy concerns, and variability in resume formats. Future directions include incorporating context-aware models to further boost accuracy, expanding datasets for broader industry applications, and addressing scalability issues in large-scale recruitment scenarios.

Ching et al., [3] developed MCQGen, a system designed to generate customized multiple-choice questions (MCQs) using advanced LLMs. This tool harnesses sophisticated prompt engineering techniques to create MCQs that are closely aligned with specific learning objectives, dynamically adjusting question complexity based on the learner's performance. The approach is supported by dual datasets—one crafted by instructors and another derived from student contributions—to ensure a rich variety of questions. Despite its promising results, the system faces limitations in terms of subject coverage and the generation of complex mathematical problems. Future enhancements aim to broaden the range of subjects, improve adaptive learning capabilities, and integrate real-time performance analytics.

Pedram et al., [4] tackled the challenge of generating subjective, open-ended questions by fine-tuning LLMs. The system, dubbed Opinerium, uses a dataset of 40,000 news articles covering diverse topics such as politics, health, and education to train models capable of creating nuanced questions that encourage deeper analytical responses. The methodology combines advanced prompt engineering with open-source tools to ensure that the generated questions maintain semantic relevance and reduce inherent biases. Key challenges noted include handling nuanced contexts and ensuring the questions accurately capture the intended complexity. Future work will focus on refining semantic understanding, reducing bias, and expanding interdisciplinary applications.

Feng et al., [5] proposed an adaptive multi-task learning framework to improve speech-to-text translation performance. This approach dynamically balances the contributions of both speech and text inputs, using adaptive weighting techniques that enhance translation quality across different languages and resource conditions. The model leverages shared representations to integrate multimodal data effectively and has been evaluated on datasets including Tibetan-Chinese, English-German, and English-French pairs. Despite promising performance improvements, the study identifies challenges such as fine-tuning adaptive weights for low-resource languages and bridging modality gaps. Future research is anticipated to focus on refining these adaptive mechanisms and expanding language support to ensure robust performance in diverse linguistic settings.

Paiva et al., [6] proposed AsanasCluster, a tool for clustering programming assignments using semantic features derived from Control Flow Graphs (CFGs) and Data Flow Graphs (DFGs). The system groups code submissions based on algorithmic strategies to support automated feedback, plagiarism detection, and learning analytics. Unlike syntax-based methods, it avoids code execution and pairwise comparisons, relying instead on incremental k-means clustering for real-time efficiency. Evaluated on the PROGpedia dataset (9,117 submissions), it achieved near real-time clustering (<7 seconds) with high precision. Challenges include handling diverse coding styles and syntax variations. Future work focuses on integrating the tool into program repair workflows and extending its use to software fault localization and performance optimization.

Zhou et al., [7] introduced a Text-to-Speech (TTS)-guided accent conversion framework that eliminates the need for parallel speech data. The method trains a TTS model on the target accent to generate pronunciation embeddings, which guide a speech encoder to map source-accented speech to the target accent. Tested on the L2-ARCTIC dataset, the system improved Word Error Rate (WER) and accentedness for conversions such as Chinese/Indian to American/British accents. Key challenges include preserving speaker identity and managing cross-linguistic pronunciation differences. Future enhancements aim to expand language support, refine pronunciation mapping, and improve speaker consistency.

Mishra et al., [8] developed JediCode, a gamified competitive coding platform integrating real-time leaderboards, synchronized challenges, and random matchmaking. Built with NestJS (backend), ReactJS (frontend), and Judge0 (code evaluation), the platform emphasizes scalability and user engagement. Challenges include maintaining real-time synchronization, ensuring secure code execution, and balancing gamification with educational outcomes. Evaluations demonstrated robust performance in handling concurrent users and dynamic

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14987553

updates. Future work targets collaborative problem-solving features, expanded language support, and enhanced learning analytics to foster community-driven skill development.

Praveen et al., [9] designed a hybrid framework for automated Multiple Choice Question (MCQ) generation using semantic ontologies and machine learning. For Wh-type questions, ontology modeling with SWRL and DL rules ensures alignment with Bloom's Taxonomy. For Cloze questions, ML techniques process technical text (e.g., operating systems textbooks) to identify keywords and generate distractors. A dataset of 4,500 sentences yielded 2,900 viable questions, addressing challenges like grammatical accuracy and ontology construction. Future directions include integrating advanced NLP models, automating ontology creation, and expanding to non-technical domains for broader educational applications.

Yuan Lai et al., [10] proposed BiMuF, a bi-directional recommender system for recruitment that matches candidates to jobs and vice versa. The framework employs multi-task learning, knowledge graphs, and LSTM-based text encoding to address data sparsity and interaction asymmetry. By sharing embeddings across tasks, it improves contextual understanding and generalization. Evaluated on the SCTCC dataset, BiMuF demonstrated effectiveness in dual recommendations. Challenges include managing sparse interaction data and optimizing semantic representations. Future work focuses on privacy-preserving computation, scalability via deep learning, and extending the model to cross-domain recruitment scenarios.

III. METHODOLOGY

The platform focuses on optimizing candidate selection by automating the initial screening stage and guiding candidates through a series of evaluation modules. It begins with resume parsing, where the system extracts critical information—such as educational background, work experience, and technical skills—from resumes. This data is then used to match candidates with suitable job listings based on predefined criteria. In addition, the system conducts a multi-phase evaluation that includes MCQ assessments to gauge theoretical knowledge, coding tests to evaluate practical skills, and HR interviews enhanced with real-time speech-to-text processing for accurate documentation of responses. Machine learning techniques, including classification models and clustering algorithms, are employed to enhance candidate ranking and identify trends in candidate performance.

➢ *Datasets*

For this project, we have compiled a diverse dataset comprising resumes and candidate profiles sourced from various recruitment channels. The dataset includes detailed candidate information such as personal details, academic qualifications, professional experience, and skill sets. In total, the dataset consists of approximately 10,000 resumes, each providing rich textual data and structured information crucial for the evaluation process.

Preprocessing involved collecting raw resume data, cleaning extraneous information, and standardizing content for analysis. Irrelevant columns were removed, and missing values were appropriately handled to ensure data integrity. Tokenization and normalization were applied to process textual content efficiently, while numerical data was formatted to facilitate model training. Once the data was cleaned and structured, feature extraction techniques—such as word embeddings and TF-IDF—were used to convert textual information into numerical representations. These features, along with structured candidate attributes, were then employed to train classification and ranking models, ensuring that every aspect of candidate information contributed effectively to the overall hiring process.

This integrated approach enables organizations to make more informed, fair, and efficient hiring decisions while significantly reducing manual workload.

➢ *Model Selection*

Model selection involves choosing the best machine learning or recommendation algorithm for a given problem based on various performance metrics. It takes into account factors like accuracy, interpretability, computational complexity, and scalability. In the context of BiMuF, models such as Large Language Models (LLMs) for MCQ generation and graph-based models like Control Flow Graphs (CFGs) and Data Flow Graphs (DFGs) for coding assessments are considered. The selected models must effectively capture both semantic understanding and the complex relationships between candidates and job roles. The final choice ensures optimal performance in candidate-job matching and assessment.

➢ *Resume Parsing*

A dual-algorithm approach is implemented. YOLOv5, a convolutional neural network (CNN)-based object detection algorithm, is used to visually segment resumes. It accurately detects and isolates key regions such as headings, sections, and layouts. Complementing this, DistilBERT—a transformer-based model that leverages attention mechanisms—is used for textual analysis. DistilBERT processes the extracted text to identify critical details, such as skills, work experiences, and qualifications, converting unstructured information into actionable data.

➢ *Algorithms Used:*

- YOLOv5: YOLOv5 is a state-of-the-art object detection algorithm that processes images in real time. In our system, it is used to visually parse resumes. YOLOv5 scans the resume image and quickly identifies distinct regions—such as headers, sections, and logos—by drawing bounding boxes around them. This detection helps segregate different parts of a resume, facilitating subsequent extraction of structured information. Its speed and accuracy make it ideal for handling diverse resume layouts.
- Distilbert: Distilbert is a transformer-based model optimized for efficient sentiment regression. It assigns sentiment scores to words based on contextual

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14987553

embeddings, capturing nuanced sentiment variations. The model is fine-tuned on sentiment datasets to improve accuracy and robustness. The performance of DistilBERT is evaluated using MSE and R-squared metrics.

➢ *Candidate Evaluation*

Candidate evaluation in the Smart Hiring System involves automated resume parsing, skill assessments, and HR interviews. The system uses machine learning algorithms to classify candidates as suitable or unsuitable for roles. Metrics such as accuracy, precision, recall, and F1 score evaluate model performance. The confusion matrix further breaks down prediction results into true positives, false positives, true negatives, and false negatives. This ensures a data-driven, objective, and efficient recruitment process.

➢ *Algorithms Used:*

- Bi-directional Recommender System (BiMuF): A Bi-directional Recommender System (BiMuF) uses multi-semantic filters to match candidates to job roles and vice versa, ensuring the system considers various attributes like skills, experience, and preferences from both perspectives. This allows for a more balanced and effective recommendation process, as it tailors both the candidate's suitability for a role and the role's suitability for the candidate.
- Confusion Matrix: The Confusion Matrix is a performance evaluation tool used to assess the accuracy of candidate assessments. It compares predicted outcomes (e.g., whether a candidate is a good fit for a role) against actual results. The matrix is structured with four key elements: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN), which help calculate metrics like accuracy, precision, recall, and F1-score, providing insights into the effectiveness of the recommendation system in matching candidates to roles accurately.

➢ *Skill-Based Assessments:*

It leverages multiple-choice question (MCQ) generation based on job requirements, while coding assessments are conducted through Control Flow Graphs (CFGs) and Data Flow Graphs (DFGs) to ensure precise evaluation of coding skills.

- MCQ Generation Algorithm: Large Language Models (LLMs) like GPT are employed to generate personalized MCQs based on job descriptions, skills required, and other specific criteria. The LLM analyzes job requirements and formulates questions that test knowledge relevant to the role. This ensures that the MCQs are tailored to assess the most critical skills and knowledge areas.
- Coding Assessment Algorithm: Control Flow Graphs (CFGs) represent the flow of execution in a program, while Data Flow Graphs (DFGs) track the flow of data through variables and functions. By analyzing these graphs, the system evaluates the semantic correctness of code submissions. It ensures that the logic is sound, that data flows properly, and that the code adheres to specified requirements, allowing for a more accurate assessment of a candidate's coding skills.

➢ *Model Training*

The input parameters are processed as features, and the selected sentiment classification model is trained using labeled datasets. Performance metrics such as accuracy, precision, recall, and F1-score are used for evaluation. The model's generalization capability is validated using holdout validation or cross-validation, ensuring its effectiveness across unseen data.

- Bidirectional LSTM (BiLSTM): A deep learning-based model that captures context by processing text in both forward and backward directions. This allows it to better understand sentiment nuances, making it suitable for analyzing complex sentence structures.
- DistilBERT: A transformer-based model optimized for fast and efficient candidate-job matching. It processes entire text inputs simultaneously, using contextual embeddings to extract meaningful insights, ensuring high accuracy while maintaining computational efficiency.

➢ *Proposed Architecture*

The Smart Hiring System is designed to streamline recruitment through an efficient combination of user interfaces, backend operations, and algorithm-driven functionalities. The user interface layer provides interactive portals for candidates and administrators, enabling job searches, application tracking, and profile management. Personalized dashboards offer real-time updates and easy navigation.

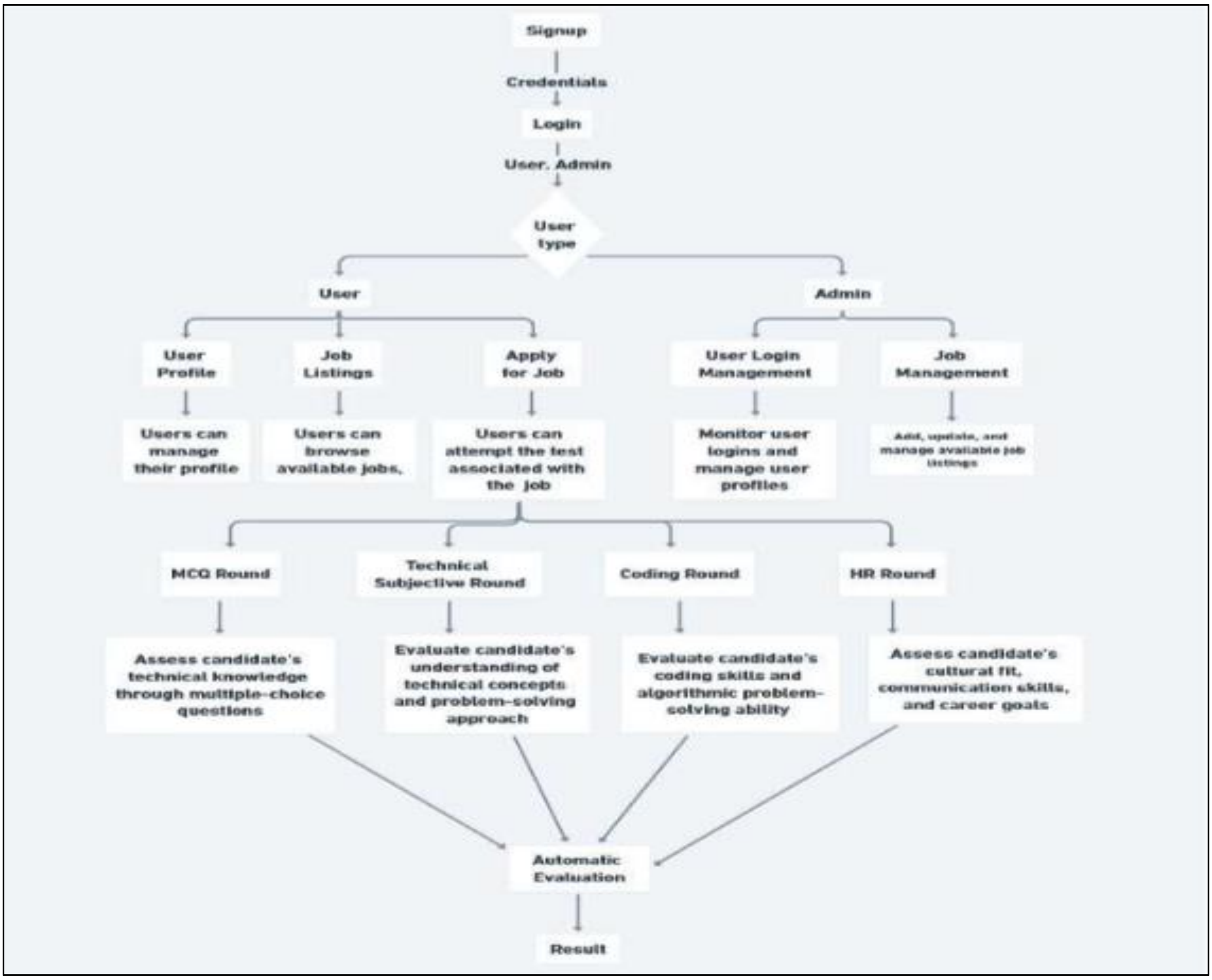

Fig1 Proposed Architecture

The backend layer ensures data security and structured management, housing user profiles, job listings, candidate assessments, and feedback records. The system manages job postings, interview schedules, and performance evaluations, offering a centralized approach to recruitment. The interview management module tracks candidate progress and facilitates decision-making for recruiters.

The algorithm layer enhances automation by incorporating resume parsing, skill-matching algorithms, and automated evaluation mechanisms. Various assessment rounds, including MCQs, technical interviews, coding challenges, and HR evaluations, optimize candidate selection. Additionally, security measures ensure proper validation of submitted files and compliance with data protection policies, making the system efficient, transparent, and data-driven.

➢ *Evaluation*

The Smart Hiring System underwent rigorous testing through various strategies, ensuring both functionality and efficiency:

- Unit Testing: Unit tests validated individual components like resume parsing, candidate assessments, and authentication. These tests confirmed that each module returned the expected outputs for predefined sample inputs.
- Integration Testing: Integration testing assessed the smooth data flow and interaction between modules, ensuring seamless communication between the frontend and backend systems.
- System Testing: System testing simulated real-world recruitment scenarios, ensuring smooth operation from login to candidate selection.
- Performance Testing: Performance testing evaluated system stability under high concurrent user loads, ensuring the system could handle peak traffic without compromising response times.
- Usability Testing: Feedback from recruiters and candidates was collected to improve the system’s user interface and experience.

To further quantify the Smart Hiring System's performance, we employed standard classification metrics, including:

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14987553

- Precision: Precision assessed the reliability of the system in identifying positive candidates:

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall(Sensitivity): Recall measured how effectively the system identified all actual positive candidates:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- F1 Score: The F1 score provided a balanced measure of precision and recall, which is especially useful in cases where the dataset might be imbalanced:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

IV. RESULTS

The results highlight significant trends and patterns that provide valuable insights into the subject under study. Notable variations and correlations in the data are discussed, offering a clear understanding of the underlying factors at play. The following visualizations provide a more in-depth representation of these results, facilitating a clearer interpretation of the data. The subsequent graphs illustrate the primary metrics and their variations over time.

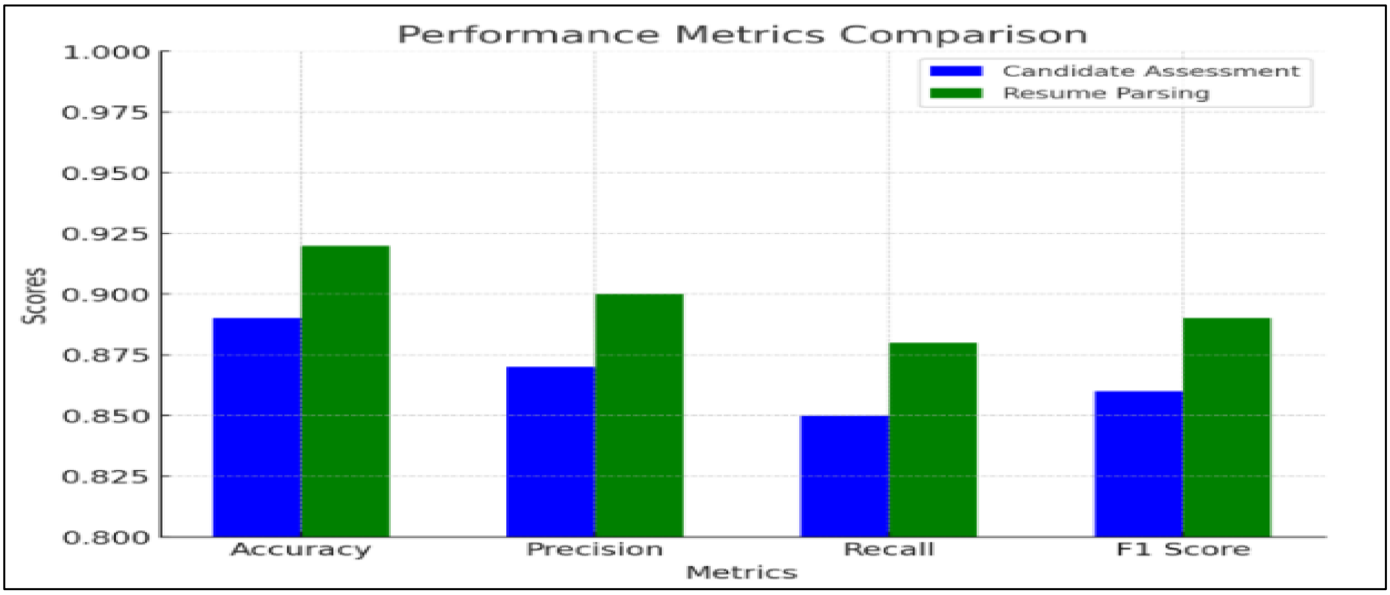

Fig 2 Performance Comparison of Candidate Assessment and Resume Parsing

Figure 2 demonstrates the system's accuracy in evaluating candidate qualifications based on predefined criteria such as skills, experience, and other relevant factors. It compares the system's automated assessment with human-reviewed results, showing the improvements in evaluation speed and consistency.

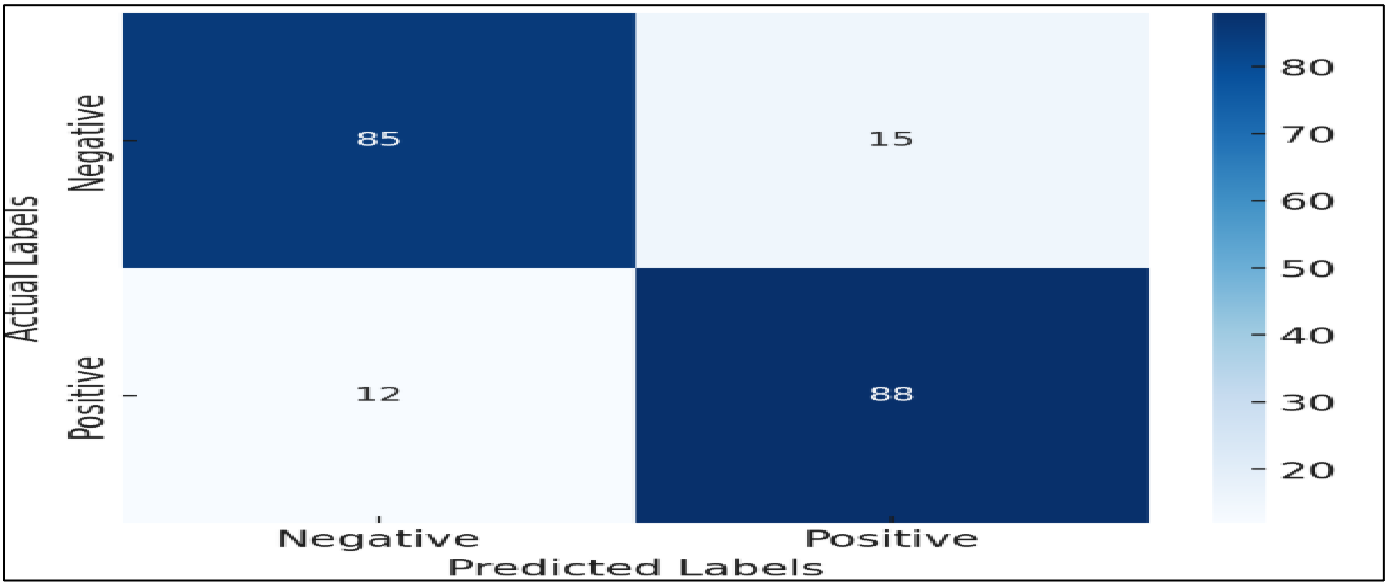

Fig 3 Confusion Matrix Representation for Candidate Assessment

This graph tracks the system's effectiveness in promoting diversity and inclusion during the hiring process. It visualizes the gender, ethnic, and experience diversity of shortlisted candidates, offering insights into how well the system aligns with inclusive hiring practices.

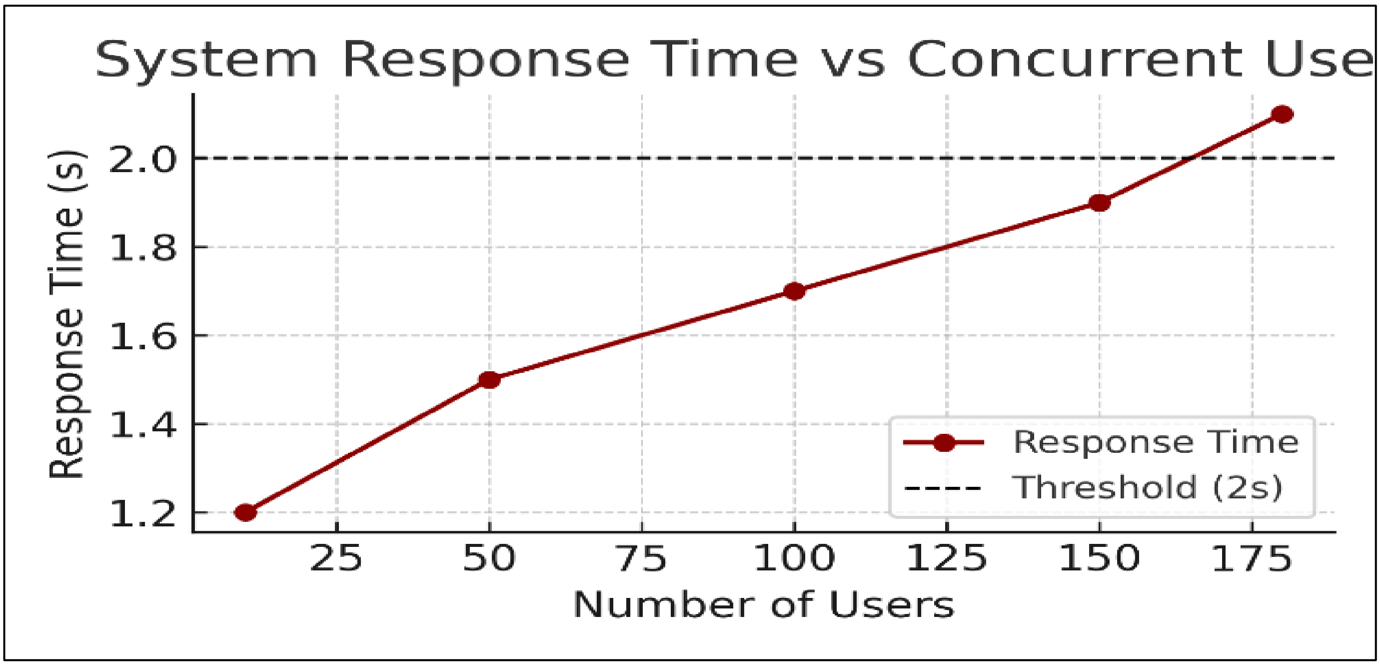

Fig 4 Performance Testing Line Graph System Response Time vs. Users

This graph showcases the time taken to shortlist candidates compared to traditional manual processes. It demonstrates the time savings made possible by automation and the NLP-based analysis, highlighting the system's potential for accelerating the hiring process while maintaining quality.

V. FUTURE SCOPE

The Smart Hiring System has the potential for significant future enhancements, including the integration of AI-powered interview bots to conduct initial candidate interviews, improving the efficiency of the hiring process. Additionally, advanced bias mitigation techniques could be implemented using machine learning and NLP to further reduce subtle biases and promote diversity. Future versions could integrate with video interview platforms to capture real-time speech-to-text data and analyze candidates' soft skills and body language. Predictive analytics could be used to assess the likelihood of candidate success based on historical data, while AI-driven job description optimization could improve candidate matching. Moreover, cross-platform integration with external HR tools like ATS and LMS would streamline data flow, further enhancing the system's capabilities and its role in a fully integrated recruitment process.

VI. CONCLUSION

The Smart Hiring System, developed using Node.js, SQL, and Express, efficiently automates the hiring process by enabling candidates to create profiles, upload resumes, and proceed through multiple assessment stages, including MCQs, subjective rounds, coding challenges, and HR interviews. The system features a responsive, user-friendly interface, with Node.js and Express managing authentication, data submission, and database interactions. Performance tests confirmed effective data processing and real-time interactions, though optimization for large datasets is needed. Challenges included real-time data flow, database optimization, and UI refinement.

Future improvements will focus on enhancing prediction accuracy, scalability, and AI integration for clearer result interpretation, reinforcing the system's potential as a robust automated hiring solution.

REFERENCES

[1]. Kinger, S., Kinger, D., Thakkar, S. et al. Towards smarter hiring: resume parsing and ranking with YOLOv5 and DistilBERT. Multimed Tools Appl 83, 82069–82087 (2024).,doi.org/10.1007/s11042-024-18778-9.

[2]. Application of LLM Agents in Recruitment: A Novel Framework for Resume Screening [https://arxiv.org/abs/2401.08315].

[3]. C. N. Hang, C. Wei Tan and P. -D. Yu, "MCQGen: A Large Language Model-Driven MCQ Generator for Personalized Learning," in IEEE Access, vol. 12, pp. 102261-102273, 2024, doi: 10.1109/ACCESS.2024.3420709.

[4]. P. Babakhani, A. Lommatzsch, T. Brodt, D. Sacker, F. Sivrikaya and S. Albayrak, "Opinerium: Subjective Question Generation Using Large Language Models," in IEEE Access, vol. 12, pp. 66085-66099, 2024, doi: 10.1109/ACCESS.2024.3398553.

ISSN No:-2456-2165 https://doi.org/10.5281/zenodo.14987553

[5]. Feng, X., Zhao, Y., Zong, W. et al. Adaptive multi-task learning for speech to text translation. J AUDIO SPEECH MUSIC PROC. 2024, 36 (2024),doi.org/10.1186/s13636-024 00359-1.
[6]. Paiva, J.C., Leal, J.P. & Figueira, Á. Clustering source code from automated assessment of programming assignments. Int J Data Sci Anal (2024).https://doi.org/10.1007/s41060-024 00554.
[7]. Y. Zhou, Z. Wu, M. Zhang, X. Tian and H. Li, "TTS-Guided Training for Accent Conversion Without Parallel Data," in IEEE Signal Processing Letters, vol. 30, pp. 533-537, 2023, doi: 10.1109/LSP.2023.3270079.
[8]. Mishra, A., & Pokalwar, S. (2023). JediCode--A Gamefied Approach to Competitive Coding. arXiv preprint arXiv:2311.10244.
[9]. Kumar, A.P., Nayak, A., K, M.S. et al. A Novel Framework for the Generation of Multiple Choice Question Stems Using Semantic and Machine-Learning Techniques. Int J Artif Intell Educ 34, 332–375 (2024). https://doi.org/10.1007/s40593-023-00333-6 .
[10]. Lai, PY., Yang, ZR., Dai, QY. et al. BiMuF: a bi-directional recommender system with multi-semantic filter for online recruitment. Knowl Inf Syst 66, 1751–1776 (2024). https://doi.org/10.1007/s10115-023-01997-1.