An AI-Powered Integrated Platform for Job Interview Coaching System

Kumari Kajal
Dept. of ECE
CMR Institute of Technology
Bengaluru, India
kuka22ece@cmrit.ac.in

Kamlesh Jain
Dept. of ECE
CMR Institute of Technology
Bengaluru, India
kaja22ece@cmrit.ac.in

Mahesh Kumar Jha
Dept. of ECE
CMR Institute of Technology
Bengaluru, India
mahesh.j@cmrit.ac.in

***Abstract*—Campus recruitment has grown rapidly, increasing the need for structured interview preparation among engineering students. Existing methods are not organized, relying on manual resume review, informal mock interviews, and separate coding platforms, and lack personalization and real-time feedback. This paper presents an Artificial Intelligence (AI) -Powered Job Interview Coach, a single integrated platform that combines resume analysis, mock interview practice, and coding assessment. The system uses Natural Language Processing (NLP) for resume–job description matching, speech processing, and machine learning techniques for interview evaluation, and automated code execution frameworks for coding assessment. Quick, data-driven feedback enables skill gap identification and performance improvement.The Test results show improved communication skills, coding proficiency, and overall job readiness.**

***Index Terms*—Job Interview Coach, Campus Recruitment, Artificial Intelligence (AI), Natural Language Processing (NLP).**

I. Introduction

Campus placements play an important role in shaping the professional careers of engineering students. With increasing competition and growing industry expectations, proper and structured interview preparation has become necessary. However, despite the availability of many preparation resources, the overall process remains scattered and inefficient. Students are often required to use separate platforms for resume review, mock interviews, and coding practice, resulting in a lack of continuity and clarity in preparation.

In typical preparation scenarios, resumes are reviewed manually with limited structured or objective feedback. Mock interviews conducted with peers or mentors provide subjective evaluation and rarely simulate real interview conditions, while coding practice is usually carried out on independent platforms disconnected from interview performance. These disconnected approaches lack personalization and real-time feedback, leaving students uncertain about their strengths, weaknesses, and general readiness for placement interviews. Consequently, students experience increased stress, inefficient use of time, and inconsistent performance during recruitment drives.

Recent advances in Artificial Intelligence (AI), Natural Language Processing (NLP), and machine learning offer an effective solution to these challenges. AI-driven systems are capable of objectively analyzing resumes, evaluating interview responses, and assessing coding performance with accuracy and consistency. By simulating real interview scenarios and providing datadriven feedback, such systems significantly enhance the quality and reliability of interview preparation.

This paper presents an AI-Powered Job Interview Coach, a single integrated platform for placement preparation that supports resume analysis, mock interviews, and coding assessments. Using NLP for resume–job description matching, speech processing, and machine learning for interview evaluation, and automated code execution for programming assessment, the system provides personalized, real-time feedback that improves technical skills, communication ability, and overall interview readiness.

The remainder of the paper is organized as: Relative work is discussed in Section II and Problem statement in Section III. Methodology of the work is discussed in section IV and Section V discuss about implementation. Section VI explains results and finally Section VII conclude the work.

II. RELATED WORK

Several studies and platforms have explored interview preparation using AI, primarily focusing on individual components such as resume screening, mock interviews, or coding assessments. However, most existing solutions address these aspects independently and do not provide a unified placement preparation framework. AI-based resume screening systems employ NLP to extract keywords and match resumes with job descriptions [1-3]. While effective for shortlisting, these approaches often rely on keyword similarity and provide limited personalized feedback, making them more recruiter-oriented than student-centric.

Mock interview systems utilize predefined question sets, chatbots, and speech-to-text techniques to evaluate responses [4-5]. Although some methods analyze sentiment or basic speech features, they lack comprehensive assessment of communication parameters such as fluency, hesitation, and clarity. Additionally, these tools typically function as standalone applications without integration with resume or coding evaluation.

Automated coding assessment platforms evaluate programming skills using test case execution and complexity analysis [6-7]. While effective for measuring code correctness, these

platforms are isolated from interview preparation workflows and offer limited guidance for overall performance improvement. Recent research[7-10] has explored AI-driven interview coaching using conversational agents and machine learning models.

III. PROBLEM STATEMENT

Many engineering students preparing for campus placements are unsure whether they are fully ready for interviews or suitable for a specific job role. Although students may have academic knowledge and technical skills, they often lack a clear understanding of their actual interview readiness, communication ability, coding proficiency, and resume relevance to industry requirements. This uncertainty affects their confidence and performance during placement interviews. Most existing placement preparation resources provide static learning material, common practice questions, or isolated evaluation tools. These methods do not offer clear and personalized assessment of a student's readiness for a specific job role. In addition, they fail to simulate real interview situations or provide meaningful feedback that highlights individual strengths, weaknesses, and skill gaps. As a result, students find it difficult to accurately evaluate their job readiness or track their improvement in a structured manner. Therefore, there is a need for an intelligent system that can objectively assess a student's placement readiness and job-role suitability. The proposed AI-Powered Job Interview Coach aims to address this challenge by using AI and NLP to analyze resumes, conduct mock interviews, evaluate coding skills, and provide personalized feedback, enabling students to prepare effectively and confidently for placements.

The proposed system acknowledges the risk of bias in AI-based evaluation of interview responses. To address this, multiple strategies are adopted using diverse and balanced training datasets. The system focuses on quality, clarity and relevance of content rather than demographic or linguistic variations. In addition, continuous human-in-the-loop validation and updates to the model are incorporated to minimize unintended bias and ensure fair, transparent, and reliable assessment results.

IV. METHODOLOGY

The proposed AI-Powered Job Interview Coach is designed as a modular system to assist students in evaluating their placement readiness through resume analysis, mock interview practice, and coding assessment. The methodology as shown in Fig.1 follows a structured pipeline consisting of data collection, preprocessing, AI-based analysis, and section-wise feedback generation.

Initially, the student interacts with the system through a webbased interface and selects the required preparation module. Based on the selected module, relevant input data such as resume documents, audio responses, or source code is collected. Each module operates independently, allowing focused assessment of specific placement skills. In the Resume Analysis Module, the student uploads a resume along with a target job description. Text extraction techniques are applied

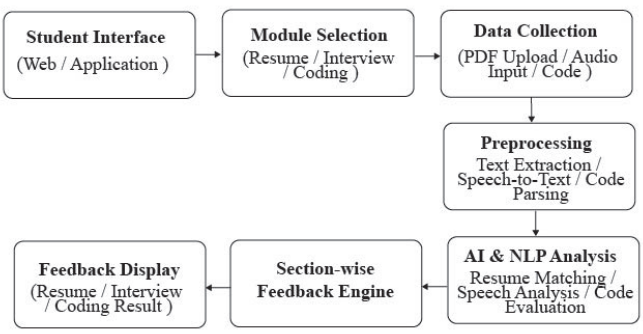

Fig. 1. Proposed Methodology

to parse the documents, followed by NLP methods to identify skills, keywords, and role-specific requirements. A semantic matching process evaluates the alignment between the resume and the job description, and feedback is generated highlighting missing skills, keyword gaps, and improvement suggestions. The Mock Interview Module simulates a real interview environment by presenting predefined interview questions. The student records audio responses for each question within a specified time limit. Speech-to-text conversion and NLP-based analysis are performed to evaluate clarity, fluency, hesitation, confidence, and relevance of the responses. Based on these parameters, interview-specific feedback is provided to improve communication and interview performance. In the Coding Assessment Module, students are given placement-oriented programming problems with time constraints. Submitted solutions are executed using an automated evaluation engine to verify correctness against test cases. Additional analysis is performed to assess logical accuracy and code quality. The system generates feedback indicating errors, optimization suggestions, and areas for improvement. Finally, the feedback generated by each module is presented separately to the student through the user interface. This section-wise feedback approach enables focused evaluation and improvement of resume quality, interview performance, and coding skills. The modular architecture ensures clarity in assessment, simplifies system design, and supports efficient processing of each component independently.

V. IMPLEMENTATION

The proposed AI-Powered Job Interview Coach is designed as a modular system to assist students in evaluating their placement readiness through resume analysis, mock interview practice, and coding assessment (as shown in fig 2). The methodology follows a structured pipeline consisting of data collection, preprocessing, AI-based analysis, and section-wise feedback generation.

A. User Interface and Application Flow

The frontend of the system is developed using HTML, CSS, and JavaScript, providing an interactive and user-friendly interface. User authentication is handled through a login and

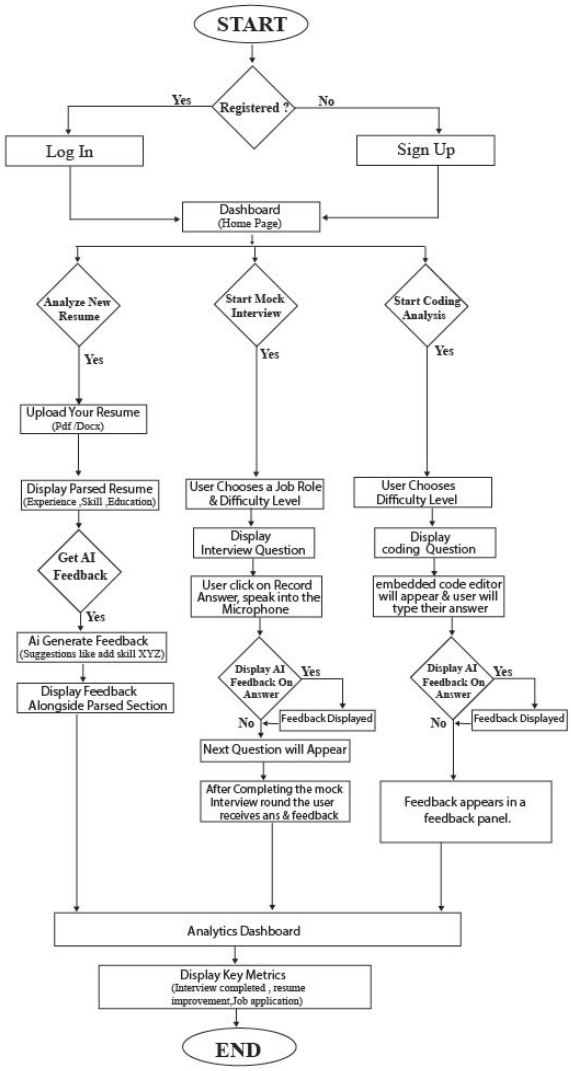

Fig. 2. Flow Chart

registration mechanism. After successful authentication, users are redirected to a dashboard where they can choose between Resume Analysis, Mock Interview, or Coding Assessment modules. The application flow follows the sequence shown in the system flowchart.

B. Resume Analysis Module Implementation

The resume analysis module allows users to upload resumes in PDF or DOCX format. Text extraction from resumes is performed using PyPDF2 and document parsing libraries. The extracted content is categorized into sections such as skills, education, and experience, and displayed on the user interface. NLP techniques are applied to analyze the resume content and generate AI-based feedback, such as identifying missing skills or suggesting improvements aligned with placement requirements. The feedback is displayed alongside the parsed resume sections for better understanding.

C. Mock Interview Module Implementation

The mock interview module is implemented using JavaScript-based audio recording in the frontend. Users select a job role and difficulty level, after which predefined interview questions are displayed. Audio responses are recorded through the system's microphone interface and processed using speech-to-text models from Hugging Face. The converted text is analyzed using NLP techniques and Gemini API to evaluate parameters such as clarity, fluency, hesitation, and relevance of answers. Feedback is generated for each response and displayed immediately to help users improve their interview performance.

D. Coding Assessment Module Implementation

The coding assessment module uses an embedded code editor implemented through frontend scripting. Users select a difficulty level and are presented with coding problems commonly asked in placement tests. Submitted code is executed and evaluated using the Judge0 API, which supports multiple programming languages. The system verifies correctness through test cases and analyzes solution logic. AI-based feedback is generated using the Gemini API, providing suggestions for optimization and improvement. Feedback is displayed in a dedicated panel to ensure clear separation from the coding interface.

E. Backend, Data Handling, and Feedback Display

The backend of the application is developed using Flask and Node.js, which handle API requests, file uploads, audio processing, and code evaluation workflows. Supabase is used for storing user data, feedback history, and performance records. Each module generates feedback independently, and the results are displayed section-wise to the user. An analytics dashboard summarizes key metrics such as completed interviews, resume improvements, and coding attempts, enabling students to track their placement preparation progress effectively.

VI. RESULTS AND DISCUSSION

The analysis demonstrates that the proposed AI-Powered Job Interview Coach provides an effective and structured approach for placement preparation by integrating resume analysis, mock interview evaluation, and coding assessment using Large Language Model (LLM). The system successfully addresses key challenges faced by students, such as uncertainty in interview readiness and job-role suitability, by delivering section-wise, AI-driven feedback. The modular design enables efficient processing of user inputs, including resume documents, audio responses, and coding solutions, ensuring smooth execution without noticeable latency.

The resume analysis module (as shown in fig 3) accurately identifies skill gaps and keyword mismatches, allowing students to better align their resumes with industry requirements.

The mock interview module (as shown in fig 4) effectively evaluates communication parameters such as clarity, fluency, hesitation, and relevance, simulating real interview conditions and improving student confidence. The coding practice dash-

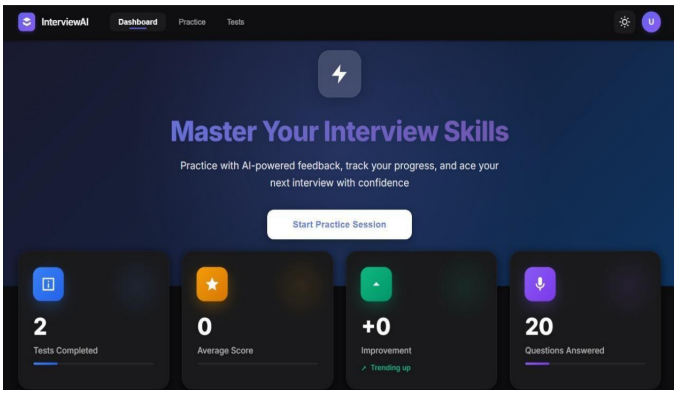

Fig. 3. Voice Interview Dashboard

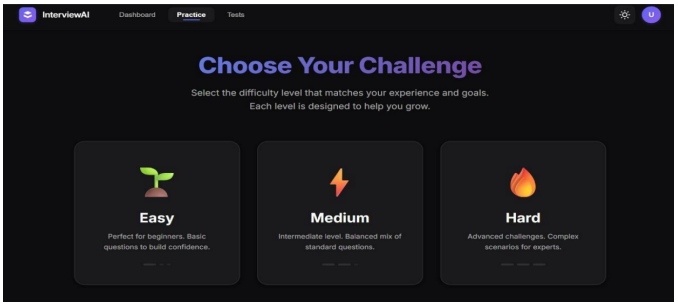

Fig. 4. Difficulty Level for Voice Interview

board (as shown in Fig. 5) provides an intuitive interface for students to access timed tests, review past attempts, analyze AI-based feedback, and track performance progress in a structured manner. The coding assessment module (as shown in fig 6) reliably verifies solution correctness and provides meaningful feedback on logic and code quality.

By combining automation with AI-based evaluation, the system minimizes manual effort while maintaining consistent and objective assessment. Overall, the integrated architecture offers a scalable and user-friendly solution for placement preparation. It enhances transparency in evaluation, supports focused skill improvement, and reduces reliance on traditional, generic preparation methods. The system proves to be a practical and efficient tool for helping students assess and improve their readiness for interviews and specific job roles.

The proposed system prioritizes the security and privacy of

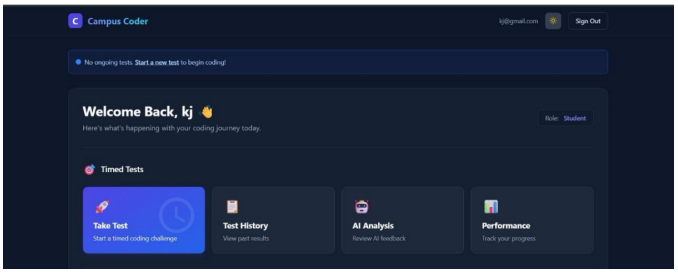

Fig. 5. Coding Practice Dashboard

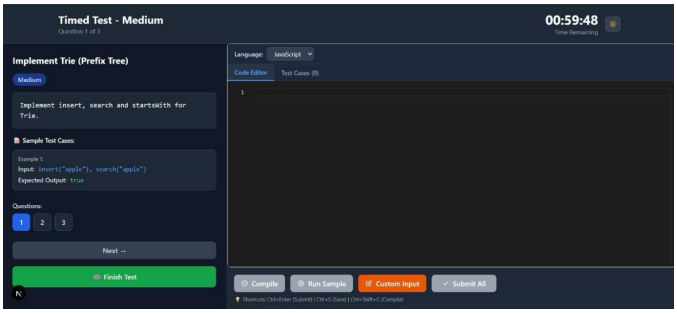

Fig. 6. Coding Practice Test

user data which may include resume or audio using secure protocols (HTTPS/TLS) and stored in encrypted form (AES-based encryption) to prevent unauthorized access. Implementation of Access control mechanisms, such as authentication and role-based authorization, can ensure that only authorized entities can access sensitive information.

VII. CONCLUSION AND FUTURE SCOPE

To address the challenges associated with uncertainty in interview readiness, lack of personalized evaluation, and limitations of traditional placement preparation methods, this paper present an AI-Powered Job Interview Coach for Engineering students. The proposed framework integrates resume analysis, mock interview evaluation, and coding assessment into a unified, modular system. The results indicate that the proposed approach improves transparency and consistency in placement preparation by identifying skill gaps, evaluating communication performance, and assessing coding proficiency. The system offers a scalable, user-centric, and efficient solution that supports focused improvement across multiple placement-related competencies. In the future, the framework can be enhanced by incorporating security measures such as secure user authentication, encrypted storage of resumes and audio data, role-based access control, and protection against unauthorized access. Additional extensions may include unified readiness scoring, advanced speech analysis, and long-term performance tracking to further strengthen system reliability, data privacy, and student employability.

REFERENCES

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in Proc. Int. Conf. Learn. Representations (ICLR), 2013.
[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, pp. 4171–4186, 2019.
[3] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," arXiv preprint arXiv:1907.11692, 2019.
[4] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. Pearson, 2023.
[5] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), pp. 6645–6649, 2013.
[6] OpenAI, "Large language models and conversational AI," OpenAI Tech. Doc., 2023. [Online]. Available: https://ai.google.dev
[7] Hugging Face, "Transformers: State-of-the-art natural language processing," 2023. [Online]. Available: https://huggingface.co/docs.

[8] M. Fowler, Patterns of Enterprise Application Architecture. Addison-Wesley, 2002.
[9] I. Sommerville, Software Engineering, 10th ed. Pearson, 2016.
[10] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Pearson, 2021. Assessment Systems: A Survey," IEEE Access, vol. 10, pp. 55678–55690, 2022.