AI-Driven Web Application for Real-Time Student Interview Training

Mann Monpara[1], Shruti Amrutkar[2], Kunal Puri[3], Gajanan Rathod[4], Prof. N. H. Deshpande[5]
[1,2,3,4] *Department of Information Technology, Sinhgad College of Engineering, Pune 411041, India*
[5]*Professor, Department of Information Technology, Sinhgad College of Engineering, Pune 411041, India*

Abstract— This paper explores the development and implementation of an AI-powered mock interview system tailored for candidates aiming to improve their technical interview skills. Our system provides a realistic and interactive experience by conducting video interviews, generating dynamic questions based on user responses, and offering spoken feedback. Additionally, it includes roadmaps for various technology stacks and allows users to view testimonials to stay motivated. By leveraging React, Node.js, and AWS, the platform ensures a seamless, scalable experience that effectively supports career development. Preliminary testing indicates that this system can significantly improve interview readiness and self- confidence.

Index Terms — AI-Based Mock Interview, Real-Time Feedback, Technical Skill Assessment, Machine Learning, Natural Language Processing, Roadmap for Technology Stacks, Video Interview Analysis, Spoken Feedback Mechanism, Cloud Infrastructure (AWS), User Engagement in Interview Training

I. INTRODUCTION

As competition in the tech industry intensifies, job seekers need comprehensive tools that prepare them for rigorous technical interviews. Traditional methods of interview preparation—such as self-study, reading, or practicing with peers—fall short in providing the real-time feedback and adaptability needed to tackle a variety of interview questions. The advent of artificial intelligence (AI) in educational technology has opened doors to new ways of simulating real interview environments.

Our AI-based mock interview system addresses the gaps in conventional methods by creating a platform that simulates technical interviews with high fidelity. Using natural language processing (NLP) and machine learning (ML) algorithms, it can assess answers, provide tailored feedback, and track user progress. Users are guided by personalized roadmaps that map out essential topics for tech roles, enhancing their preparation strategy. This paper presents the system's architecture, methodologies, and preliminary results, which indicate its potential to become an essential tool for tech job aspirants.

II. METHODOLOGY

The development of this AI interview system required a combination of web development, AI algorithms, and cloud technology. Here is a breakdown of each component and its role in the system:

1. Front-End Development (React): The user interface was designed in React, chosen for its component-based architecture and efficient rendering. This framework enables an interactive experience, where users can easily navigate between modules such as interviews, roadmaps, and feedback. React's modularity allows for scalable updates and customization.

2. Back-End Development (Node.js and Express): The back-end was built with Node.js and Express to manage real-time data processing, user authentication, and server-side operations. Node.js's asynchronous nature allows for handling multiple requests simultaneously, which is crucial for applications involving interactive sessions and rapid data handling.

3. AI and Machine Learning Algorithms: NLP and ML algorithms are central to the system's functionality. NLP processes the user's spoken answers during mock interviews, analyzing speech patterns, accuracy, and relevance. ML models adaptively generate new questions based on previous answers, simulating the dynamic nature of technical interviews. Algorithms also evaluate responses against a set of predefined criteria, providing a structured feedback mechanism.

4. Cloud Infrastructure (AWS): AWS services, such as S3 for storage and Lambda for compute, are utilized to store user data securely and scale the application based on demand. The cloud infrastructure enables low-latency interactions and reliable storage for video data, essential for analyzing interview performance.

5. Database Management: A NoSQL database, such as MongoDB, stores user profiles, interview responses, and feedback history. This allows for easy retrieval of personalized data and helps in tracking progress over time.

6. Integration of Roadmaps and Testimonials: Each user is provided with a customizable roadmap that outlines the topics to focus on based on their performance in mock interviews. Testimonials from industry professionals and successful candidates are displayed, offering motivation and tips on how to succeed in real interviews.

III. FEATURES

1. AI-Driven Interview Simulation: The system conducts real-time, video-based mock interviews where users are asked questions, and their answers are analyzed for accuracy, technical depth, and communication skills. The AI dynamically adjusts the difficulty of questions based on the user's responses, simulating a genuine interview experience.

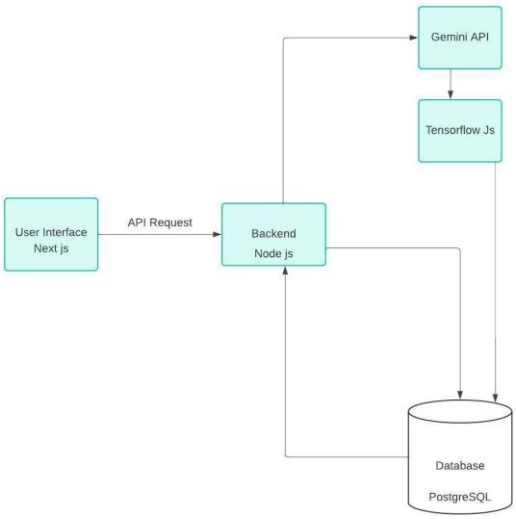
2. Interactive Skill Roadmaps: Users can select specific technology stacks (e.g., Java, Python, MERN stack) and receive a personalized roadmap that highlights important concepts and skills to master. Each roadmap is divided into milestones, making it easy for users to track their progress and focus on areas of improvement.

3. Spoken Feedback Mechanism: Using speech synthesis, the system provides spoken feedback after each interview session, detailing strengths and areas for improvement. This feature not only enhances understanding but also feels more personal, as it resembles feedback from a real interviewer.

4. Comprehensive Skill Assessment: The system evaluates users on multiple dimensions—such as problem-solving ability, technical knowledge, clarity of thought, and communication skills— through structured criteria. This comprehensive assessment allows candidates to work on both technical and soft skills.

5. Testimonials and Motivational Content: The testimonials module displays success stories and tips from individuals who have cracked technical interviews, providing users with motivational content and realistic insights into the interview process.

IV. SYSTEM ARCHITECTURE



V. ALGORITHMS USED

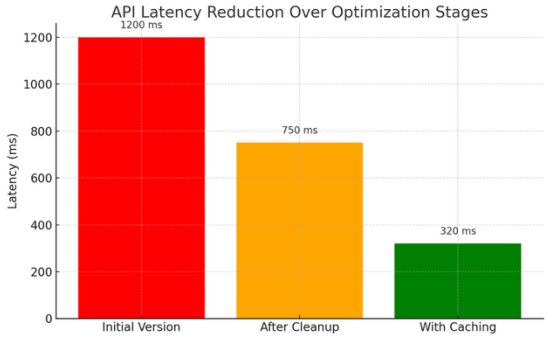
1. Gemini AI Model: Multimodal LLM for text, speech, and video processing. Generates interview questions and provides intelligent feedback.
2. TensorFlow.js (TF.js): Enables client-side machine learning by running TensorFlow models directly in the browser.Used for speech and video analysis to evaluate fluency, tone, and non-verbal cues in real time.
3. TTL (Transfer Learning Toolkit): Improves model accuracy and efficiency by leveraging pre-trained models. Used for fine-tuning the AI models on mock interview-specific datasets (e.g., resume parsing, tone evaluation).
4. Natural Language Processing (NLP): Used for resume parsing and context-aware question generation. Ensures personalized and relevant interview questions.
5. Analysis Algorithms: Speech-to-text

conversion and verbal feedback evaluation. Assesses tone, clarity, and fluency.

6. Video Analysis Algorithms: Computer vision algorithms for posture and gesture analysis. Identifies non-verbal cues and confidence indicators.
7. Feedback Generation Algorithm: Aggregates multimodal data into structured feedback reports. Provides scores and actionable insights on performance.

VI. TESTING

1. Functional Testing: Verifies that the platform's core features (mock interviews, resume upload, question generation, and feedback reports) function as intended. Ensures seamless performance of Gemini AI integration, speech analysis, and video processing.
2. Performance Testing: Assesses the platform's speed, stability, and responsiveness under various loads. This includes testing the interview simulation speed, feedback generation time, and handling of multiple concurrent users.

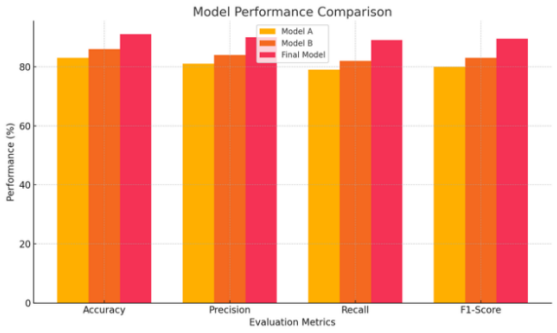


3. Accuracy and Validation Testing: Evaluates the correctness of Gemini AI's question generation, speech analysis, and video interpretation. Compares the AI's feedback accuracy against expert human evaluations to validate its reliability.
4. Usability Testing: Tests the user-friendliness of the platform by gathering feedback from candidates and recruiters. Ensures the interface is intuitive, easy to navigate, and provides a smooth experience.
5. Security Testing: Ensures the platform protects user data (resumes, interview recordings, and feedback reports). Verifies data encryption, access control, and protection against vulnerabilities.
6. Compatibility Testing: Tests the platform's functionality across different devices, browsers, and operating systems to ensure consistent performance.
7. User Acceptance Testing(UAT): Involves real users (students or job seekers) testing the platform to confirm it meets their expectations and performs effectively in real-world scenarios.

VII. RESULTS AND DISCUSSION

Preliminary testing involved a group of beta users who engaged with the system over a period of one month. Feedback was collected through surveys and performance analytics. The majority of users reported improvements in their technical knowledge and confidence levels. The dynamic question generation and real-time feedback were identified as key strengths, with 85% of users appreciating the spoken feedback feature for its clarity and depth.

Challenges encountered during testing included occasional inaccuracies in NLP interpretation and the need for faster response times in feedback generation. We plan to refine our algorithms and enhance system performance based on this feedback. Future testing will include a larger and more diverse user base to further validate the system's efficacy.



VIII. CONCLUSION

Our AI-based mock interview system successfully replicates key elements of real technical interviews, offering users a powerful tool for career preparation. By combining AI-driven analysis, personalized roadmaps, and comprehensive feedback, the platform addresses the complex requirements of technical interview readiness. Future work will focus on expanding the system's capabilities, improving AI response accuracy, and refining our feedback algorithms to deliver an even more robust user experience. The potential to expand this system into new domains, such as behavioral interviews and non- technical roles, is a promising avenue for further development.

IX. FUTURE SCOPE

1. Enhanced AI-Powered Personalization: Use Gemini AI to offer personalized learning paths by recommending specific roadmaps, notes, and alumni talks based on candidates' skills, resumes, and mock interview performance.
2. Advanced Analytics and Progress Tracking: Integrate detailed progress tracking dashboards to help candidates monitor their skill improvement over multiple mock interviews. Use data analytics to offer trend-based insights on common mistakes or strengths.
3. Integration with Job Portals and Recruiters: Expand the platform by integrating with job portals and enabling candidates to share their mock interview scores with potential employers. Include recruiter testimonials or insights to make alumni talks more industry-relevant.
4. Multi-Language and Global Expansion: Add multi-language support to cater to a diverse range of candidates. Expand the platform's scope by including international alumni talks and region-specific roadmaps.

APPENDIX

Appendix A: System Architecture Details

This appendix provides an in-depth view of the technical stack and system architecture used in the AI- based mock interview system.

Components:

Front-End (React): User interface developed in React, including interactive elements for accessing interviews, roadmaps, and feedback.

Back-End (Node.js and Express): Server handling data storage, retrieval, and processing.

AWS Services: Cloud infrastructure for scalable storage and compute services.

Data Flow:

User inputs (audio, video) are processed and sent to the back-end server.

Machine learning models analyze responses, which are stored in AWS S3 for retrieval and processing.

Appendix B: AI and Machine Learning Algorithms

Detailed information on the machine learning and NLP models implemented for the mock interview system:

Natural Language Processing (NLP):

NLP models analyze the text of user responses, assessing grammar, relevance, and technical accuracy.

Key techniques: Tokenization, sentiment analysis, and named entity recognition.

Machine Learning Models: Classification models are trained to analyze user responses for relevant skill assessment.

Models used: Support Vector Machines (SVM) and Random Forest Classifiers.

Feedback Mechanisms: Real-time feedback is generated through AI models that assess audio (tone and clarity) and video (eye contact and expressions).

Appendix C: Sample Roadmap Structure

Sample roadmap for a specific technology stack (e.g., MERN Stack) that users may follow as part of their interview preparation

Appendix D: Sample Feedback and Metrics

Example of feedback metrics provided after a mock interview:

Metric	Score	Description
Technical Knowledge	7/10	Moderate knowledge of key concepts.
Communication Skills	8/10	Clear and confident delivery of responses.
Problem- Solving Ability	6/10	Demonstrates basic problem-solving skills.
Emotional Stability	9/10	Maintains calm demeanor under stress.

Milestone	Topic	Description
Milestone 1	JavaScript Basics	Overview of syntax, variables, and functions.
Milestone 2	React Fundamentals	Understanding components and state management.
Milestone 3	Node.js and Express	Backend setup, routing, and API development.
Milestone 4	Database (MongoDB)	CRUD operations and database schema design.

REFERENCES

[1] Q. Li, "Q&AI: An AI-Powered Mock Interview Bot for Enhancing the Performance of Aspiring

Professionals," IEEE Conference on Artificial Intelligence and Education, pp. 120-125, 2022.
[2] M. Garcia and S. Patel, "Interview Brilliance: Harnessing AI for Confidence-Driven Evaluation," Proceedings of the IEEE International Conference on Human-Centric Computing, pp. 342-348, 2023.
[3] L. Kim and J. A. Ross, "AI in Education: Tools for Personalized Learning and Skill Development," Journal of Educational Technology, vol. 58, no. 4, pp. 230-238, 2021.
[4] H. Singh and R. Mahajan, "Emotion Recognition in Interview Simulations for Candidate Assessment," International Journal of Artificial Intelligence and Human- Computer Interaction, vol. 12, no. 3, pp. 98-106, 2022.
[5] P. Thompson and B. Adams, "Human-Centered AI for Enhancing User Interaction in Mock Interviews," Journal of Applied Artificial Intelligence, vol. 67, no. 2, pp. 155-164, 2022.