AI MOCK INTERVIEW SYSTEM
With Real-Time Voice Interaction and Intelligent Feedback Platform

[1] M.Abhishek [2] Dr.Meerasharif Sheik
[1]Student, [2]Professor
[1]Computer Science and Engineering,
[1]BVCEC, Odalarevu, India

***Abstract*:** The recruitment process in the software industry increasingly emphasizes communication skills, technical clarity, and real-time problem-solving ability. Traditional mock interview preparation methods often lack real-time feedback, conversational realism, and performance analytics. This paper presents an AI Mock Interview System developed using modern web technologies and artificial intelligence tools to simulate realistic interview experiences.

The system integrates real-time speech recognition, intelligent response generation, voice synthesis, and conversational backchanneling to create a human-like interview environment. The application uses Next.js for the frontend, Node.js with Socket.IO for real-time communication, speech-to-text conversion and text-to-speech for interviewer voice output, and a Large Language Model (LLM) to generate context-aware responses.

The proposed system enhances interview preparation by providing interactive sessions, performance analysis, transcript storage, and adaptive questioning strategies. This platform improves student confidence, communication skills, and technical readiness through realistic AI-driven interviews.

Index Terms - Mock Interview, Speech Recognition, Real-Time Communication, LLM, Voice Interaction

1. INTRODUCTION

Preparing for technical interviews is one of the biggest challenges faced by students and job seekers. While theoretical knowledge can be acquired through books and online courses, real-time communication and structured interview practice are difficult to simulate without a mentor.

Conventional mock interview systems rely on static question banks or pre-recorded responses, which lack interactivity and personalization. To address this issue, this project introduces an AI-powered Mock Interview System that simulates a real interviewer using speech recognition, natural language processing, and voice synthesis technologies.

The system provides a fully interactive experience where the student speaks through a microphone, receives AI-generated interviewer responses, and continues the conversation dynamically.

2. NEED FOR THE STUDY

The need for an AI Mock Interview System arises due to:

- Limited access to experienced mentors
- Lack of real-time conversational feedback
- Absence of voice-based interview simulation
- No structured performance analytics
- High cost of professional interview coaching

This study focuses on developing an intelligent, scalable, and cost-effective solution that enables students to practice interviews anytime with realistic AI interaction.

3. LITERATURE SURVEY

Several existing interview preparation platforms use AI and chatbot-based systems to simulate interview questions and evaluate candidate responses. Most of these systems are limited to text-based interaction or pre-recorded video analysis, lacking real-time conversational flow and dynamic follow-up questioning.

Although recent advancements in speech recognition and large language models have improved automation, many platforms do not integrate real-time voice interaction, Voice Activity Detection (VAD), or human-like backchanneling. The proposed AI Mock Interview System addresses these limitations by combining speech recognition, LLM-based response generation, and text-to-speech technologies to create a realistic and interactive interview experience.

4. PROPOSED SYSTEM

The proposed AI Mock Interview System is a real-time web-based application designed to simulate realistic technical interviews using artificial intelligence. The system enables candidates to interact with an AI interviewer through voice-based communication, creating a natural and engaging interview experience.

The system is built using a scalable architecture with Next.js for the frontend and Node.js with Socket.IO for real-time communication. Voice Activity Detection (VAD) ensures smooth conversational flow by detecting when the candidate starts and stops speaking. The platform also stores interview transcripts and provides structured feedback, allowing users to analyze their performance and improve their communication and technical skills effectively.

5. ARCHITECTURE OVERVIEW

The system follows a three-tier architecture:

- Presentation Layer – Web interface developed using Next.js and modern front-end technologies to provide an interactive interview dashboard, microphone access, and real-time audio playback.

- Application Layer – Backend logic implemented using Node.js, handling session management, LLM prompt processing, speech-text integration, and interview flow control.

- Database Layer – Used MySQL to store interview transcripts, session history, performance feedback, and user data.

User requests are processed through RESTful APIs, ensuring smooth communication between the front end and back end.

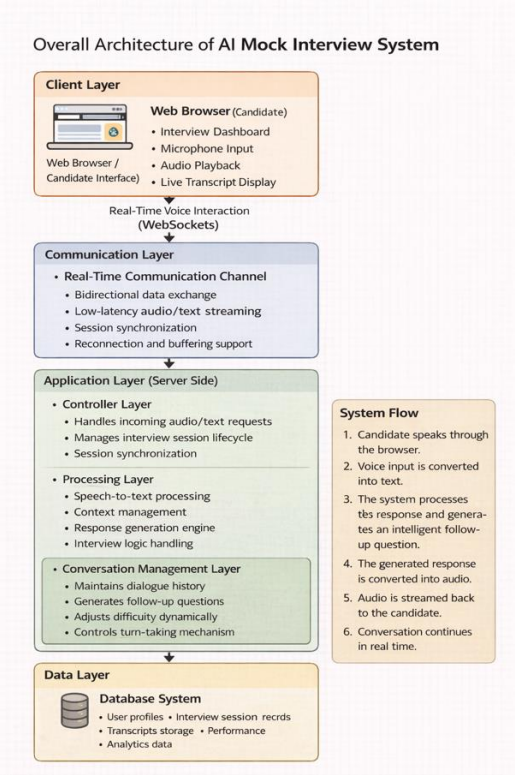

Fig 5. System architecture

6. MODULES EXPLANATION

The proposed system is divided into two functional modules to ensure efficient management, security, and scalability. Each module performs a specific task within the system.

6.1 Interview Session Module

- Start Interview Session
- Controls conversation flow
- Maintains context history

6.2 Speech Recognition and Synthesis Module

- Converts student speech into text using Deepgram
- Uses VAD to detect when the student stops speaking
- Operates in real time while maintaining low latency.

6.3 LLM Response Module

- Generates interviewer questions and provides follow-up questions dynamically.
- Determines the type of next question (technical, behavioural, or coding) based on candidate responses.
- Adjusts difficulty based on answers.

6.4 Feedback & Analytics Module

- Evaluates communication clarity and problem solving skills.
- Provides structured performance feedback including question-wise report.
- Gives total feedback of overall performance in the interview.

7. IMPLEMENTATION

The implementation of the AI Mock Interview System was carried out through multiple development iterations to ensure smooth real-time interaction and conversational accuracy. The system was designed to simulate a realistic interview environment where candidates can communicate naturally through voice. Secure authentication mechanisms were implemented to create protected user sessions, and role-based routing ensured controlled access to interview dashboards and stored performance data.

During the development of the candidate interface, special focus was given to user experience and conversational flow. The dashboard provides easy access to starting interviews, viewing previous sessions, and reviewing performance feedback. The microphone integration allows seamless voice input, while live transcript display enables users to monitor their responses in real time. The system dynamically processes spoken responses, maintains conversational context, and generates intelligent follow-up questions to simulate a human interviewer. Voice Activity Detection ensures smooth turn-taking, preventing interruptions during the interaction.

On the server side, session management and conversation control mechanisms were implemented to maintain dialogue continuity. The system stores transcripts, session history, and feedback data for performance analysis. Real-time communication handling ensures minimal latency and continuous bidirectional data flow between client and server. Proper buffering and reconnection mechanisms were implemented to handle temporary network disruptions without breaking the interview session.

- **Front End:** Next.js, TailwindCSS, ShadcnUI

- **Back End:** Node.js, Express.js, REST APIs

- **Database:** MySQL, MongoDB

- **Communication Protocol :** Real-time bidirectional communication

- **Development Tools**: VS Code, IntelliJ IDEA, Postman

- **Functionalities:** Real-time voice interview, dynamic questioning, performance analytics

The application was tested locally using a browser-based environment under multiple simulated interview scenarios to ensure stability, responsiveness, and accuracy of conversational processing.

8. RESULTS & ADVANTAGES

8.1 Results

After successful implementation and testing, the AI Mock Interview System demonstrated reliable and consistent performance across all functional modules. The system was able to conduct real-time voice-based interview sessions with secure authentication and proper session management, ensuring that only authorized users could access interview features and stored performance data. The real-time communication mechanism enabled smooth interaction between the candidate and the AI interviewer without noticeable interruptions.

Candidates were able to speak naturally through the browser, and their responses were accurately converted into text and processed dynamically. The system successfully generated context-aware follow-up questions, maintaining conversational continuity throughout the session. Transcripts were stored automatically, and performance feedback was recorded for later analysis. The implementation of turn-taking control ensured natural conversation flow, preventing overlap between user speech and AI responses.

Overall, the system performed as intended, maintaining low latency during interactions, accurate conversational processing, and stable session handling. The project successfully achieved its objective of providing a realistic, AI-driven mock interview platform that enhances interview preparation through interactive voice-based simulation.

8.2 Advantages

The AI Mock Interview System offers several practical and technological advantages over traditional interview preparation methods. One of the major benefits is the ability to simulate real-time interview conversations using voice interaction, which helps candidates improve both technical knowledge and communication skills. Unlike static question banks, the system provides dynamic and context-aware follow-up questions, creating a more realistic and engaging experience.

Secure authentication and session management ensure that user data, interview transcripts, and feedback reports are protected from unauthorized access. The centralized storage of session history allows candidates to review past performances and track improvement over time. The user-friendly interface makes the platform easy to navigate, even for users with minimal technical background.

Additionally, the system is scalable and can be extended with advanced features such as AI-based scoring, emotion analysis, coding assessment modules, and cloud deployment. This makes it a long-term,

intelligent solution for interview preparation that can adapt to evolving recruitment standards and technological advancements.

9. CONCLUSION

The AI Mock Interview System presents an innovative approach to interview preparation by integrating real-time voice interaction, intelligent response generation, and structured performance analysis into a single platform. By simulating a realistic interview environment, the system bridges the gap between theoretical learning and practical communication skills. The implementation of conversational flow control and adaptive questioning enhances user engagement and creates a human-like interview experience.

10. FUTURE SCOPE

The AI Mock Interview System can be further enhanced by integrating advanced intelligent features to improve realism and evaluation accuracy. Future improvements may include automated scoring mechanisms that assess communication clarity, technical accuracy, confidence level, and response structure. The system can also incorporate emotion and tone analysis to evaluate candidate confidence and stress levels during interviews.

Additionally, the platform can be expanded to support coding interviews with real-time code execution and automated evaluation. Multi-language interview support can be introduced to make the system accessible to a wider audience. Cloud deployment and scalable infrastructure can further improve performance and allow large-scale concurrent interview sessions. These enhancements would make the system more robust, intelligent, and suitable for real-world recruitment preparation at an advanced level.

11. REFERENCES

[1] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017).
Attention Is All You Need. Advances in Neural Information Processing Systems (NeurIPS).

[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019).
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT.

[3] Radford, A., et al. (2019).
Language Models are Unsupervised Multitask Learners. OpenAI Research Report.

[4] Jurafsky, D., & Martin, J. H. (2023).
Speech and Language Processing (3rd ed.). Pearson.

[5] Graves, A., Mohamed, A., & Hinton, G. (2013).
Speech Recognition with Deep Recurrent Neural Networks. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

[6] Young, S., Gašić, M., Thomson, B., & Williams, J. D. (2013).
POMDP-based Statistical Spoken Dialogue Systems: A Review. Proceedings of the IEEE, 101(5), 1160-1179.

[7] Myers, B. A., et al. (2018).
The Design and Evaluation of Conversational Agents. ACM Computing Surveys.

[8] Fielding, R. T. (2000).
Architectural Styles and the Design of Network-based Software Architectures. Doctoral Dissertation, University of California, Irvine.

Copyright & License:

© Authors retain the copyright of this article. This work is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.