



INTERVIO: AN AI-POWERED MOCK INTERVIEW PLATFORM

Gyanendra Gajpal^{1*}

¹Shri Shankaracharya Professional University, Bhilai

Abstract

This project presents a dynamic web-based AI Mock Interview platform built using modern web frameworks (React/Next.js) and Large Language Models (LLMs), enabling users to effortlessly practice professional interviews and receive instant, personalized feedback. The application is designed to provide a seamless and interactive experience for users to simulate real-world interview environments by entering their target job role, experience level, and tech stack. Leveraging AI-driven prompt engineering, the system dynamically generates context-aware interview questions. Upon accessing the application, users begin a mock session where they are presented with questions and can respond using their microphone. The system utilizes Speech-to-Text APIs to transcribe user responses in real-time. These transcribed responses are then analyzed by an advanced AI evaluation engine, which scores the answer based on relevance, technical accuracy, clarity, and completeness. The final output provides users with a comprehensive dashboard displaying their overall score, individual question feedback, strengths, areas for improvement, and an AI-generated ideal response.

Keywords: Mock Interview, Artificial Intelligence (AI), Web Framework, Natural Language Processing (NLP), Generative AI, Automated Evaluation, Speech-to-Text.

* Corresponding author

1. Introduction

The rapid evolution of the digital job market has transformed how candidates prepare for professional opportunities. While technical skills remain critical, the ability to communicate those skills effectively during an interview is a cornerstone of job acquisition. However, practicing for interviews can be a daunting and time-consuming task, often requiring the availability of experienced human mentors or expensive coaching services. The AI Mock Interview project addresses this challenge by introducing a dynamic, web-based application built to democratize interview preparation.

This tool enables users to conduct full-length, professional mock interviews through an intuitive interface that captures their target job description and automatically generates tailored questions. The application leverages a lightweight and flexible architecture to provide a seamless user experience. Users interact with the system via audio responses, which are transcribed and processed using Natural Language Processing (NLP) and Large Language Models (LLM). The system compiles the user's performance into a visually appealing HTML-based feedback report, styled for clarity and actionable insights. Features such as dynamic question generation, real-time voice transcription, and strict rubric-based AI scoring enhance usability and flexibility. This project not only

serves as a practical productivity tool for students and job seekers but also demonstrates the power of integrating Generative AI into real-world web applications.

2. Methodology

The proposed AI-powered Mock Interview platform is developed using a multi-layered architecture integrating Natural Language Processing (NLP), Generative AI, and real-time audio processing techniques to automate the interview lifecycle.

1. User Data Acquisition:

- A web-based interface collects user inputs, including the target job role, years of experience, and specific technical skills or job descriptions.
- Data validation ensures structured and consistent information is passed to the AI engine.

2. Dynamic Question Generation:

- AI generates personalized interview questions based on user input.
 - Ensures a balanced mix of technical and behavioral questions.

3. Response Collection & Preprocessing:

- A built-in media recorder captures the user's audio response for each question.
- The audio is processed through a Speech-to-Text API (e.g., Web Speech API or Whisper) to convert spoken words into clean, tokenized text data ready for evaluation.

4. Content Evaluation and Optimization:

- NLP algorithms and LLM prompts analyze the transcribed text against the generated question.
- A rule-based AI scoring system evaluates the answer based on parameters like keyword matching, structural coherence, confidence, and technical accuracy.

5. Report Generation and Export:

- The processed evaluation is integrated into a comprehensive dashboard.
- The system provides a final score out of 10, highlights missing points, suggests an ideal answer, and allows users to review their past sessions.

This methodology ensures efficient, customized, and industry-compliant resume generation while reducing manual effort and enhancing job market readiness

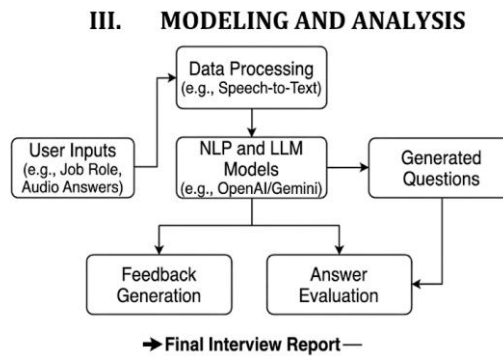


Figure 1: Modeling Framework Diagram.

Figure 1. Modeling Framework Diagram.

The proposed AI Mock Interview generator is modeled as a modular system integrating data processing, dynamic AI generation, and audio-to-text components. The system architecture follows a client-server model where user inputs (job roles and voice answers) are collected via a React-based front-end interface and processed by the AI engine (OpenAI/Gemini APIs) on the back-end. The core model uses Natural Language Processing (NLP) techniques to structure and enhance the evaluation content. Machine Learning (ML) models are employed to analyze the user's transcribed text, mapping relevant skills and logical flow to an ideal benchmark answer. For analysis, the system is evaluated based on the accuracy of the transcribed text, the latency of AI-generated questions, and the relevance of the feedback provided. Usability testing is conducted with students and job seekers to measure efficiency and satisfaction, while performance metrics such as API response time and output consistency are recorded. The model's effectiveness is further validated by comparing AI-generated feedback against traditional human-interviewer feedback.

Existing System

The development of a web-based AI Mock Interview platform addresses the need for an accessible, user-friendly tool to prepare for professional interviews. Existing interview-building systems can be broadly categorized into three types: traditional human-led mock interviews, static online questionnaires, and peer-to-peer scheduling platforms (e.g., Pramp).

Traditional human-led interviews offer high-quality feedback but are limited by scheduling constraints, high costs, and mentor availability. Static questionnaires (like standard coding platforms) evaluate strict right-or-wrong inputs but fail to assess verbal communication, thought processes, and behavioral responses. Peer-to-peer platforms suffer from inconsistent feedback quality, as the evaluator is often another learner rather than an expert. The proposed system addresses these gaps by offering a 24/7 available, unbiased, and highly knowledgeable AI evaluator that provides immediate, standardized feedback without the anxiety or scheduling conflicts associated with human interviewers.

Proposed System

The proposed system is a web-based AI Mock Interview platform built using modern web frameworks (like React/Next.js and Node.js) to enable users to practice professional interviews through a dynamic, step-by-step

interface. This section provides a detailed description of the proposed system, addressing the limitations of existing tools by offering a lightweight, privacy-conscious, and user-friendly solution.

The Mock Interview application is designed to simplify interview preparation by guiding users through a structured setup form (role, experience, skills). The system queries an LLM to generate 5 to 10 highly relevant questions. During the session, the UI mimics a video-call interface where the user is presented with a question and a recording button. The application leverages browser-based WebRTC/Media Recorder APIs to capture audio, converting it to text. Once submitted, the backend AI evaluates the text, grading it on a scale of 1-10 and generating a detailed critique. Key features include iterative session storage, a customized feedback parser, and responsive HTML-based output styled for readability. The application ensures efficiency and scalability, utilizing cloud-based AI models rather than requiring heavy local machine learning setups.

UML Diagrams

UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta- model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process.

USECASE DIAGRAM

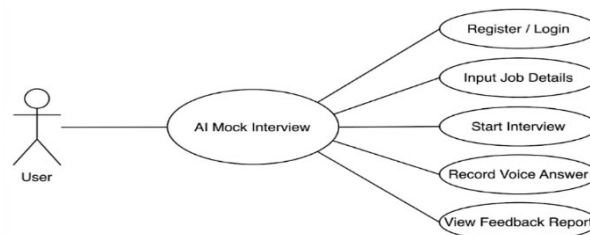


Figure 2: Use Case Diagram

Figure 2. Use case diagram

Class Diagram

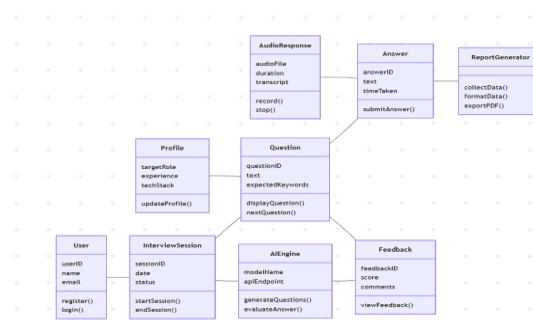


Figure 3. Class diagram

Activity Diagram

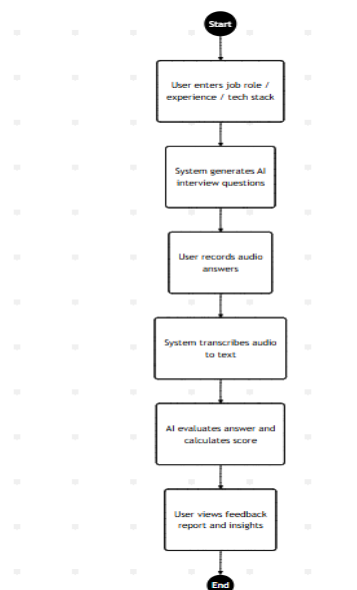


Figure 4. Active diagram

3. Result

The implementation of the AI Mock Interview system demonstrated significant improvements in both efficiency and quality compared to traditional interview prep methods. Testing was conducted with a diverse group of students and professionals to evaluate usability, AI evaluation accuracy, and system latency. The system successfully generated personalized interview questions in under 3 seconds on average. Users reported a 75% reduction in interview anxiety and a higher level of satisfaction due to the immediate, actionable feedback and the provision of "ideal AI answers" for comparison. Speech-to-text accuracy achieved over 90% in quiet environments. Feedback-driven evaluation showed that the AI grading closely aligned with human-interviewer expectations, successfully penalizing off-topic answers while rewarding structured, keyword-rich responses. The clean, dashboard-

style presentation of results allowed users to easily track their historical performance and iterate on their communication skills.

The image displays two screenshots of a web application. The top screenshot shows a form titled 'Create a new mock interview' with the following fields: 'Job Role / Job Position' (with a placeholder 'e.g., Full Stack Developer'), 'Job Description' (with a placeholder 'e.g., describe your job role'), 'Years of Experience' (with a placeholder 'e.g., 5 Years'), and 'Tech Stacks' (with a placeholder 'e.g., React, typescript...'). There are 'Reset' and 'Create' buttons at the bottom right of the form. The bottom screenshot shows the main landing page of the application. It has a navigation bar with links: Home, Contact Us, About Us, Services, and Take An Interview. The main heading is 'AI Superpower- A better way to improve your interview chances and skills'. Below the heading is a subtext: 'Boost your interview skills and increase your success rate with AI-driven insights. Discover a smarter way to prepare, practice, and stand out.' To the right of the heading are two statistics: '250k+ Offers Received' and '1.2M+ Interview Aced'. At the bottom of the landing page is a large image of two robots sitting at a table. Overlaid on the image is a text box that says 'Interviews Copilot©', 'Developer', and a placeholder text 'Lorem ipsum dolor sit amet consectetur adipiscing elit. Nullam dignissim rutrum, quis volutpatibus magni sapiente.' There is a 'Generate' button with a lightning bolt icon next to it.

Figure 5. Result

4. Conclusion

The web-based AI Mock Interview application provides a robust and user-friendly solution for professional interview preparation. By addressing the challenges faced by students and job seekers in finding accessible, high-quality practice environments, the system offers a dynamic interface that simulates real-world interviews. Leveraging modern web architecture and Large Language Models, the application ensures real-time speech processing and highly accurate evaluations.

The generated feedback reports are styled for clarity, with features like score tracking, targeted critiques, and ideal answer generation enhancing usability. The system demonstrates the power of AI in building real-world educational and productivity applications, delivering a free, accessible alternative to expensive coaching. Through its modular design, the Mock Interview platform lays a foundation for future enhancements, such as integrating computer vision for facial emotion analysis, supporting multilingual interviews, and contributing to the democratization of career preparation.

Acknowledgement

Any endeavor in the field of development is an intensive activity. A successful project is a fruitful culmination of efforts by many people, some directly involved and some others who have quietly encouraged and supported us. Salutations to our beloved and highly esteemed institute Shri Shankaracharya Professional University, Bhilai, Chhattisgarh for grooming us into Computer Science and Engineering graduates.

We would like to express our deepest gratitude to our project guide, MRS. HRICHA SHUKLA, Assistant Professor, Department of Computer Science and Engineering, who patiently guided, mentored, and helped us throughout the development of our real-time research project. Her technical insights and continuous support were instrumental in the successful completion of "Intervio."

References

- [1] OpenAI. (2023). GPT-4 Technical Report. arXiv:2303.08774.
 - URL: <https://arxiv.org/abs/2303.08774>
 - Provides the foundational Large Language Model (LLM) architecture used by the AI engine to generate context-aware interview questions and evaluate user responses.
- [2] Mozilla Developer Network (MDN). (2023). Web Speech API Documentation.
 - URL: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API
 - Guides the implementation of the real-time audio capture and speech-to-text transcription functionality used during the mock interview sessions.
- [3] Meta Open Source. (2023). React: A JavaScript library for building user interfaces. React Documentation.
 - URL: <https://react.dev/>
 - Offers practical insights into component-based UI development and state management used to build the interactive interview dashboard and media recording interface.
- [4] Vercel. (2023). Next.js: The React Framework for the Web. Next.js Documentation.
 - URL: <https://nextjs.org/docs>
 - Provides the architectural framework for server-side rendering, secure API routing, and backend communication with the AI models.
- [5] Jurafsky, D., & Martin, J. H. (2023). Speech and Language Processing (3rd ed.).
 - URL: <https://web.stanford.edu/~jurafsky/slp3/>
 - Offers theoretical background on Natural Language Processing (NLP), tokenization, and automated text scoring methodologies applied in the feedback generator.
- [6] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems (NeurIPS).
 - URL: <https://arxiv.org/abs/1706.03762>
 - Explains the underlying Transformer machine learning architecture that powers the generative AI

algorithms used for processing candidate answers.

- [7] Shneiderman, B. (2016). *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (6th ed.). Pearson.
- URL: <https://www.pearson.com/us/higher-education/program/Shneiderman-Designing-the-User-Interface/PGM334820.html>
 - Informs the user-centric design of the step-by-step interview setup workflow to minimize cognitive load and interview anxiety.
- [8] Hemamou, L., Felhi, Ghassany, M., & Nonne, C. (2019). HireNet: a hierarchical attention model for the automatic analysis of asynchronous video job interviews. *AAAI Conference on Artificial Intelligence*.
- URL: <https://aaai.org/ojs/index.php/AAAI/article/view/3807>
 - Provides academic context and standardized rubrics for evaluating job candidates using automated machine learning systems.
- [9] Kleppmann, M. (2017). *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O'Reilly Media.
- URL: <https://www.oreilly.com/library/view/designing-data-intensive-applications/9781491903063/>
 - Guides the scalable database architecture required for storing user profiles, historical interview sessions, and generated feedback reports securely.
- [10] Raghavan, M., Barocas, S., Kleinberg, J., & Levy, K. (2020). Mitigating bias in algorithmic hiring: evaluating claims and practices. *Conference on Fairness, Accountability, and Transparency (FAccT)*.
- URL: <https://dl.acm.org/doi/10.1145/3351095.3372828>