

# AI Mock Interview: An Intelligent Voice-Driven Interview Simulation System using Gemini AI and Whisper

Abdul Wahid, Aditya Jha, Meerhan Munshi, Sumit Sonwane, Prof. Amit Chakrawarti  
Department of Artificial Intelligence & Machine Learning  
Dilkap Research Institute of Engineering and Management Studies, University of Mumbai  
Village Mamdapur, Post-Neral, Tal: Karjat, Maharashtra, India - 410101

**Abstract** - This paper presents the design and implementation of an AI Mock Interview system, an advanced voice-interactive web application developed using Streamlit. The system integrates Google Gemini for intelligent, resume-aware question generation and real-time performance evaluation, and OpenAI Whisper for speech-to-text transcription, enabling a fully conversational interview experience. Upon uploading a resume in PDF format and selecting a technical domain, the candidate engages in a dynamic interview session where spoken responses are transcribed, analyzed, and followed up intelligently. Post-session, a comprehensive performance report covering technical depth, communication skills, problem-solving ability, and domain suitability is auto-generated as a downloadable PDF using ReportLab. The system targets students, job seekers, and professionals seeking personalized, on-demand technical interview preparation. Experimental evaluation demonstrates the system's ability to generate contextually relevant questions, provide meaningful feedback, and simulate a realistic interview environment superior to existing platforms.

**Keywords** - AI Mock Interview; Google Gemini; OpenAI Whisper; Speech-to-Text; Resume Parsing; Natural Language Processing; Streamlit; ReportLab; Technical Interview Preparation; Generative AI

## I. INTRODUCTION

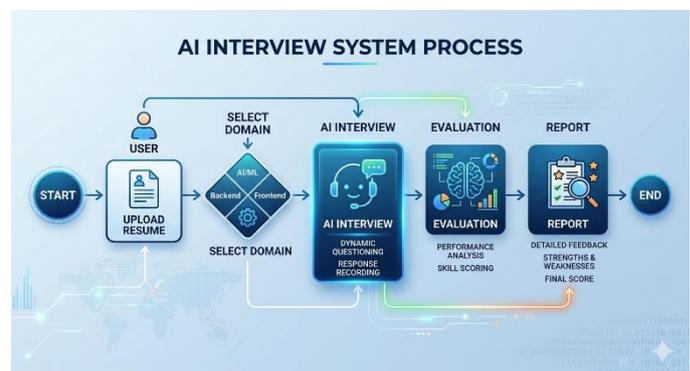
In today's competitive job market, technical interview preparation is a critical yet often inadequately addressed challenge for students and early-career professionals. Traditional preparation methods such as peer mock interviews, textbook review, or platform-based coding challenges offer fragmented solutions that fail to replicate the holistic dynamics of a real interview, particularly the interplay of technical knowledge, verbal communication, and adaptive questioning.

Artificial Intelligence has transformed numerous facets of human-computer interaction, and its application to interview simulation presents a compelling opportunity. Large language models (LLMs) such as Google Gemini can generate contextually rich, domain-specific questions, while OpenAI's Whisper provides state-of-the-art automatic speech recognition (ASR) across diverse accents and acoustic environments. The convergence of these technologies enables the construction of a fully autonomous, conversational interview coach.

This paper presents the AI Mock Interview system — a Streamlit-based web application that orchestrates resume analysis, real-time voice interaction, intelligent follow-up

questioning, and automated performance evaluation within a single, unified platform. The system addresses five core deficiencies identified in the existing landscape: (1) absence of voice-based interaction, (2) lack of resume-personalized questioning, (3) no automated holistic evaluation, (4) poor accessibility for individual learners, and (5) absence of structured, downloadable performance reports.

The remainder of this paper is organized as follows: Section II surveys related work; Section III defines the problem and objectives; Section IV describes the proposed system architecture; Section V details the methodology; Section VI presents system design; Section VII states hardware and software requirements; and Section VIII concludes with future directions.



## II. LITERATURE SURVEY

### A. Review of Existing Interview Platforms

Several AI-assisted platforms have been developed for interview preparation and recruitment automation. HireVue [1] and Talview [2] are enterprise-grade video interviewing solutions that leverage computer vision and NLP to assess facial expressions, tone, and verbal content. While highly effective for large-scale recruitment, these systems are designed for employers rather than self-directed learners and lack transparency in their scoring rubrics.

Pramp [3] and Interviewing.io [4] offer peer-to-peer mock interview experiences with real-time feedback from industry professionals. These platforms effectively simulate the social dynamics of interviews but are constrained by human

availability and provide no automated, data-driven evaluation or downloadable reports.

HackerRank [5] and LeetCode [6] have established themselves as the de facto standards for coding skill assessment, offering automated code evaluation across thousands of algorithmic problems. However, they are narrowly scoped to programming challenges and entirely omit communication assessment, behavioral dimensions, and open-ended technical discourse.

### B. Identified Research Gaps

A critical survey of these systems reveals the following persistent gaps, summarized in Table I:

Platform	Voice-Based	Resume-Aware	Auto-Feedback	PDF Report	Free/Learner
HireVue	Partial	No	No	No	No
Pramp	Yes	No	No	No	Yes
HackerRank	No	No	No	No	Yes
Proposed System	Yes	Yes	Yes	Yes	Yes

TABLE I. Comparison of Existing Systems vs. Proposed System

Beyond the features listed, existing systems also lack adaptive difficulty calibration based on real-time performance, domain suitability analysis, and alternative career path recommendations — capabilities integrated into the proposed system.

## III. PROBLEM DEFINITION

### A. Problem Statement

Candidates preparing for technical interviews face a fragmented ecosystem: coding platforms that ignore communication, peer networks constrained by availability, and enterprise recruitment tools inaccessible to individual learners. No single platform combines voice-based interaction, resume-personalized questioning, automated holistic evaluation, and structured reporting into a coherent, learner-centered experience.

### B. Objectives

The primary objectives of the AI Mock Interview system are:

- To develop an AI-powered, voice-interactive platform for realistic technical interview simulation.
- To extract and analyze candidate resume data for generating personalized, domain-specific questions.
- To transcribe verbal responses using ASR and evaluate them for technical depth, communication clarity, and problem-solving approach.
- To provide a comprehensive performance analysis covering strengths, weaknesses, and improvement recommendations.
- To auto-generate a downloadable PDF report encapsulating the full interview analysis.
- To ensure a clean, accessible, browser-based interface requiring no specialized hardware beyond a standard microphone.

### C. Scope

The system supports six technical domains: Software Development, Data Science, Machine Learning, Cloud Computing, Cybersecurity, and Web Development. It is designed for individual self-assessment and iterative improvement, and can be extended toward institutional

deployment for placement preparation or educator-led assessments.

## IV. PROPOSED SYSTEM ARCHITECTURE

The AI Mock Interview system is structured into five loosely coupled, independently maintainable modules operating under a client-server architecture.

### A. System Overview

As illustrated in Fig. 1, the system follows a linear pipeline: Resume Upload → Domain Selection → AI Question Generation → Voice Response Capture → ASR Transcription → AI Evaluation → Performance Analysis → PDF Report Generation. Each stage feeds structured data to the next, maintaining a coherent conversation history in session state.

### B. Module Descriptions

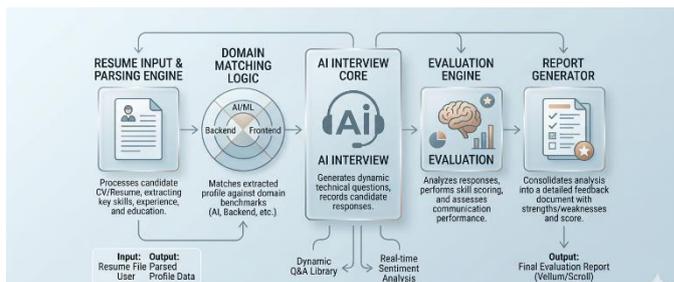
1) Resume Analysis Module: The candidate uploads a resume in PDF format. PyPDF2 extracts raw text, which is then formatted as a structured context prompt for the Gemini API. The module isolates skills, educational qualifications, prior experience, and project history to inform personalized questioning.

2) AI Question Generation Module: Leveraging Google Gemini's large-scale generative capabilities, the module constructs an initial prompt combining the parsed resume, the selected domain, and an instruction to generate progressively challenging, contextually appropriate questions. Follow-up questions are generated dynamically based on the quality and content of the candidate's prior responses, simulating the adaptive logic of a skilled human interviewer.

3) Audio Processing Module: Candidate responses are captured via the SoundDevice library, stored as temporary .wav files using SoundFile, and passed to OpenAI's Whisper model for transcription. Whisper's multi-lingual, accent-robust architecture ensures high transcription accuracy. The resulting text is appended to the session conversation history.

4) Performance Evaluation Module: At interview conclusion, the full conversation history is submitted to Gemini with a structured evaluation prompt requesting: (a) an overall score (0-100), (b) communication skill rating, (c) technical depth assessment, (d) key strengths and improvement areas, (e) problem-solving approach rating, and (f) domain suitability analysis with upskilling recommendations.

5) Report Generation Module: ReportLab compiles the structured evaluation into a multi-section PDF report with consistent typography, section headers, and score visualizations. The report is made available for immediate download from the Streamlit interface.



## V. METHODOLOGY

The development methodology follows a waterfall model, ensuring rigorous sequential validation at each stage: Requirement Analysis → System Design → Implementation → Testing → Deployment.

### A. Data Input and Resume Parsing

The process begins when the candidate uploads a resume (PDF) and selects a domain. PyPDF2's PdfReader extracts text page-by-page. The concatenated text is cleaned of formatting artifacts and embedded in a Gemini prompt template that instructs the model to identify key skills and experience markers relevant to the chosen domain.

### B. Dynamic Question Generation Workflow

The initial question prompt is structured as: "Given the following resume context: {resume\_text}, generate an opening technical interview question for the domain: {domain}. The question should be appropriate to the candidate's stated experience level." After each response, the conversation history (question-answer pairs) is appended to a follow-up prompt: "Based on the candidate's answer: {answer}, generate the next interview question that probes deeper or transitions to a related topic." This chain produces a coherent, contextually evolving interview session typically comprising 8-12 questions.

### C. Audio Transcription Pipeline

Audio is sampled at 16 kHz (optimal for Whisper's acoustic models), stored as 16-bit mono .wav, and passed to `whisper.load_model("base").transcribe()`. The base model balances speed and accuracy for interactive use. The transcribed text undergoes basic cleaning (removal of filler words, punctuation normalization) before evaluation.

### D. Evaluation Prompt Engineering

The evaluation prompt explicitly specifies the desired JSON output schema, requesting keys for `overall_score`, `communication_rating`, `technical_depth`, `strengths` (list), `improvement_areas` (list), `problem_solving_rating`, and `alternative_domains` (list with rationale). Gemini's structured output mode ensures reliable parsing without regular-expression post-processing.

### E. Performance Report Structure

The generated PDF report comprises: (1) Candidate profile summary, (2) Interview transcript, (3) Quantitative performance dashboard (radar chart of skill dimensions), (4) Narrative strengths and improvement analysis, (5) Domain suitability matrix, and (6) Recommended learning resources and upskilling directions.



## VI. SYSTEM DESIGN

The system adopts a modular client-server design. The Streamlit application serves as both frontend and lightweight server, hosting all processing logic within the Python runtime. External API calls to Google Gemini and Whisper inference are the primary I/O bottlenecks; these are handled asynchronously where supported.

Session state management in Streamlit maintains conversation history, uploaded file buffers, and evaluation results across widget interactions without requiring a persistent database for single-session use. For multi-session tracking, SQLite integration is provided as an optional configuration, enabling performance trend analysis across multiple practice sessions.

The UI presents a chat-style interface where AI-generated questions appear as interviewer messages and transcribed candidate responses appear as candidate messages. Real-time transcription status and recording indicators provide clear feedback, minimizing user uncertainty during voice capture phases.

## VII. SYSTEM REQUIREMENTS

### A. Hardware Requirements

Minimum specifications for reliable operation include: Intel Core i5 / AMD Ryzen 5 or equivalent (4+ cores, 2.0 GHz+); 8 GB RAM (16 GB recommended for concurrent AI model execution); 500 MB free storage for temporary audio and generated reports; a quality microphone for clear audio capture; and a stable broadband internet connection (10+ Mbps) for real-time API calls to Google Gemini. An NVIDIA GPU accelerates local Whisper inference but is optional when using the cloud API variant.

### B. Software Requirements

Component	Tool/Technology	Purpose
Language	Python 3.10+	Core development
UI Framework	Streamlit	Web interface
AI Model	Google Gemini API	Q&A generation & eval
Speech-to-Text	OpenAI Whisper	Audio transcription
PDF Parsing	PyPDF2	Resume extraction
Report Gen.	ReportLab	PDF report creation
Audio I/O	SoundDevice & SoundFile	Recording & saving

TABLE II. Software Stack

The application is compatible with Windows 10/11, macOS 10.15+, and major Linux distributions. Deployment can be containerized using Docker for institutional use, with environment variables managing API key injection securely.

### VIII. CONCLUSION

This paper presented the AI Mock Interview system, a comprehensive, voice-driven interview preparation platform that integrates Google Gemini for adaptive question generation and performance evaluation with OpenAI Whisper for accurate speech-to-text transcription. The system addresses critical gaps in existing solutions by offering resume-personalized questioning, real-time conversational interaction, holistic automated evaluation, and downloadable PDF reporting within a single accessible web application.

The modular architecture ensures that individual components — resume parsing, question generation, audio transcription, evaluation, and report generation — can be independently updated as underlying AI models evolve. The system's flexibility across six technical domains and its self-paced, iterative learning model make it suitable for students, job seekers, and professionals at various experience levels.

Future work will focus on: (1) incorporating real-time emotion and sentiment analysis for richer communication feedback; (2) implementing adaptive difficulty calibration using reinforcement learning from user performance history; (3) expanding domain support to include behavioral and HR interview simulation; and (4) developing a multi-modal evaluation pipeline that integrates video analysis for posture, eye contact, and non-verbal communication assessment. These enhancements will further bridge the gap between AI simulation and the full complexity of real-world interview environments.

### ACKNOWLEDGMENT

The authors express sincere gratitude to Prof. Amit Chakrawarti (Head of Department, AIML) and Prof. Priyanka Khanke for their invaluable guidance, encouragement, and sustained mentorship throughout this project. Special thanks are extended to the staff of the AIML Laboratory at Dilkap Research Institute of Engineering and Management Studies for providing the resources and environment necessary to conduct this research.

### REFERENCES

- [1] HireVue, "AI-Powered Video Interviewing and Assessment Platform," HireVue Inc., 2024. [Online]. Available: <https://www.hirevue.com>
- [2] Talview, "AI-Driven Remote Hiring and Proctoring Platform," Talview Inc., 2024. [Online]. Available: <https://www.talview.com>
- [3] Pramp, "Free Peer-to-Peer Mock Interview Platform," Pramp by Exponent, 2024. [Online]. Available: <https://www.pramp.com>
- [4] Interviewing.io, "Anonymous Technical Mock Interviews with Professionals," 2024. [Online]. Available: <https://interviewing.io>
- [5] HackerRank, "Technical Assessment and Remote Interview Solution," HackerRank Inc., 2024. [Online]. Available: <https://www.hackerrank.com>
- [6] LeetCode, "Online Judge and Technical Interview Preparation Platform," LeetCode LLC, 2024. [Online]. Available: <https://leetcode.com>
- [7] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," in Proc. Int. Conf. Machine Learning (ICML), 2023, pp. 28492-28518.
- [8] Google DeepMind, "Gemini: A Family of Highly Capable Multimodal Models," Google LLC, Technical Report, 2023.
- [9] A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.
- [10] B. Chen and R. Cardie, "Multinomial Adversarial Networks for Multi-Domain Text Classification," in Proc. NAACL-HLT, 2018, pp. 1226-1240.
- [11] M. Maslej-Kresnakova, P. Butka, and M. Sarnovský, "Comparison of Different Approaches for Sentiment Analysis," in IEEE 16th Int. Symp. Intelligent Systems and Informatics, 2018.
- [12] ReportLab Inc., "ReportLab PDF Library User Guide," ReportLab, 2023. [Online]. Available: <https://www.reportlab.com/docs/reportlab-userguide.pdf>