InterviewAI: A Voice-Driven Multi-Agent Framework for Adaptive Mock Interviews with Real-Time Feedback

Prof S.A.Gaikwad
Assistant Professor
Department of Computer Science And Engineering TPCTs COE Dharashiv, Maharashtra, India
sujatagaikwad414@gmail.com

Priyanka Mohan Doijode
Student
Department of Computer Science And Engineering TPCTs COE Dharashiv, Maharashtra, India
priyanckabaalerao@gmail.com

Abstract*—Preparing for an interview continues to be a major difficulty for job seekers due to the lack of scalable, realistic, and feedback-driven practice settings. Conventional mock-interview platforms use scripted text interfaces or question banks, which do not effectively assess genuine conversational competency, vocal confidence and contextual reasoning. We offer InterviewAI, a voice-enabled artificial intelligence interview preparation tool that incorporates real-time conversational agents, adaptive question development, and automated performance metrics in a cloud-native architecture. The suggested system leverages Vapi AI speech agents for natural spoken interaction, Google Gemini for dynamic interview question synthesis and response evaluation, and Firebase for secure authentication, session persistence, and interview lifecycle management. A modular multi-agent process synchronises interviewer behaviour, transcript production, answer analysis and rubric-based feedback scoring. The difficulty of the interview and the development of topics are adjusted according to the candidate's answers, semantic coherence and the patterns of hesitation that are observed. Following each interview, the platform provides structured comments on technical accuracy, clarity of communication, signals of confidence and relevancy of answers. Experimental deployment showed low-latency voice interaction, consistent question creation, steady transcript synchronisation, and stable feedback alignment across technical and behavioural dimensions. The aim of this architecture is to allow scalability, device independence, and rapid extension to multiple job positions without the need to retrain the core platform. The approach demonstrates how voice-centric engagement, adaptive reasoning and automated qualitative evaluation may be combined into a single interview-intelligence pipeline.

Index Terms*—Conversational agents, artificial intelligence interview systems, voice-based human-AI interaction, automated feedback generation, adaptive assessment platforms, cloud-native interview analytics.

I. INTRODUCTION

Job interviews are one of the most essential gateways in modern career growth, affecting recruiting decisions in technical, management and professional fields. With the world's workforce expanding rapidly and competition intensifying for skilled employment, candidates require realistic practice environments, objective performance assessment, and personalised improvement coaching. Text-only preparation platforms, peer mock sessions and static question banks do not adequately recreate the conversational pressure, timing, clarification and spontaneous reasoning needed in an actual interview. Recent advances in conversational artificial intelligence, automatic speech recognition, text-to-speech synthesis, massive language models, and cloud databases have made it possible to provide low-cost, cross-device, cross-role, interactive interview simulators. But many contemporary systems remain inflexible: they use predefined scripts, provide delayed or generic feedback, do not maintain conversational context, or decouple question delivery from performance rating. These constraints limit their utility for judging quality of communication and role-specific skills. A second difficulty is the lack of rapid, actionable feedback associated with the particular conversational sequence. Relevance, clarity, reluctance, confidence, topic consistency, and depth of explanation are often neglected or subjectively appraised in the responses. An effective training platform would be able to associate feedback with the question-response history, discriminate technical accuracy from style of communication, and indicate the objectives for improvement that the applicant can then practise in a subsequent session.

In this study, we address these restrictions using InterviewAI, a speech-enabled mock interview platform powered by Vapi AI voice agents, Google Gemini reasoning services, Firebase Authentication and Firestore, and a responsive Next.js interface. The system logs verbal responses, saves ordered transcripts, develops role-specific questions, performs structured post-interview evaluation, and retains historical sessions for progress analysis. The main contribution is the integration of real-time voice chat, adaptive question generation, secure session administration and rubric-based feedback in a single extendable process.

II. LITERATURE SURVEY

AI-based interview preparation and automated applicant assessment are active research fields, since organisations want scalable recruitment help and learners require consistent and personalised practice. Most of the existing technologies are based on static surveys, manual mock sessions, or rule-based chatbots. Such systems are not particularly flexible and offer little empirical indication of hesitancy, coherence, confidence or conversational recovery. Dialogue-management systems and semantic parsing added more context to the conversation, while transformer models offered flexibility in generating questions and interpreting answers. There are a number of chatbot interview aides that have been built on cloud infrastructure and natural-language processing, but many of them focus on text and can't assess vocal delivery, pauses or confidence in talking. Voice-enabled systems are able to enhance engagement and realism through automatic speech recognition and text-to-speech synthesis, yet drawbacks such as fixed interview flows and poor integration with feedback production are still widespread.

The automated performance evaluation has advanced from keyword matching to semantic similarity, embedding-based scoring, and prompt-engineered large-language-model evaluation. These methodologies can evaluate relevance, coherence, clarity, and job-role alignment, but should be bounded by specific rubrics to reduce inconsistency. The use of a multi-agent architecture enables segregate interviewer behaviour, transcript processing, response evaluation and feedback creation, thus increasing modularity and allowing each component to be audited independently. The literature therefore suggests that there is a gap between conversational realism and systematic assessment. Often systems will optimise for the interview experience, or optimise for the ultimate evaluation, but not keep the two together in a single real-time framework. InterviewAI closes this gap by synchronising voice agents, generative reasoning, secure cloud persistence, and dashboard analytics such that each session is both a realistic practice interview and structured learning record.

III. METHODOLOGY

We built InterviewAI, a modular cloud-native mock-interview and feedback generation system. The methodology comprises of user authentication, interview setup, real-time voice interaction, dynamic question generation, transcript persistence, automated assessment and dashboard visualisation. The layered design isolates the user interface from conversational services and storage, facilitates component replacement, and decreases the connection between voice processing, reasoning, and feedback presentation.

A. Proposed System Architecture

Candidates will mostly interact with a Next.js web application. User accounts are managed with Firebase Authentication, and interview metadata, transcripts, timestamps, score outputs and feedback are stored in Cloud Firestore. Vapi AI manages speech recognition, conversation delivery, and text-to-speech synthesis. For example, Google Gemini generates domain-specific questions and evaluates responses based on the transcript and session environment. The services are connected by secure API communication channels so that the system can be scaled independently at the client, conversational, reasoning and storage layers.

B. Voice-Based Interview Interaction

The interview is performed by Vapi AI speech agents over streaming audio. Candidate speech is transcribed into structured transcripts and given to the interview state manager. Low-latency streaming with incremental transcriptions preserves the natural flow of conversations. The interviewer agent can ask follow-up questions, request clarification or increase difficulty based on the response and the role set.

C. Adaptive Question Generation

Google Gemini creates questions appropriate to the role, based on the job domain, interview stage, previous questions, candidate responses and desired complexity. Prompt templates keep the topic on track and prevent asking the same questions repeatedly. The state manager keeps track of the already assessed competencies, enabling the system to switch from introductory questions to scenario-based or more in-depth technical questions as the candidate displays sufficient understanding.

D. Session Management and Data Persistence

Every interview is given a unique session ID. Firebase holds the user identify, selected role, interview setup, ordered question-response pairs, timestamps, evaluation outcomes, and feedback summary. This data model allows for interview history, comparison between sessions, dashboard visualisation, and future analytics without attaching the application to one voice or language-model supplier.

E. Automated Feedback and Scoring

Following the interview, a structured evaluation prompt assesses technical relevance, logical coherence, depth of explanation, communication clarity, confidence indicators and relevance of the response. Rubric anchors give meanings to low, middle and high scores. It decreases free-form judgement. The scoring output is normalised and saved with textual strengths, weaknesses and action Suggested practice. So the candidate can evaluate both quantitative indicators and qualitative feedback associated with proof.

F. Data Communication and Dashboard

The frontend, AI services and Firebase backend connect with each other over secure APIs and JSON payloads. A responsive dashboard built with Next.js and Tailwind CSS that displays interview history, transcript details, feedback

summaries and performance trends on both desktop and mobile devices. This closed loop method allows for longitudinal skill growth and repeated practice.

G. Prompt Engineering and Rubric Design

"Prompt engineering is a kind of controlled software component, rather than an informal instruction. The interviewer prompt sets the role, seniority level, interview length, tone, type of questions and the competencies to cover. It further instructs the agent to ask only one question at a time, wait for the full answer, not to reveal model answers too early, and to produce follow-up questions only when they are relevant to the prior answer given by the applicant. We enter session metadata through fixed fields so that the prompt is stable among applicants.

The evaluator prompt is isolated from the interviewer prompt to minimise role conflict. . The question, transcript, target role, scoring rubric and any reference criteria are provided. The evaluator should return a structured object containing a technical score, communication score, confidence indicator, relevance score, strengths, improvement points and excerpts of evidence. Numeric fields are limited to valid ranges and free text fields are limited in length so that the presentation of the dashboard is predictable. Before writing the record to Firestore, a validation phase checks for missing fields, out-of-range values, or malformed JSON.

Rubrics are built on descriptions of behaviour. A high technical score suggests correctness, completeness, and depth appropriate for the position; a medium score indicates partially right reasoning with lacking detail; and a low score indicates severe conceptual error or irrelevance. Communication clarity involves organization, language and conciseness and confidence indicators are inferred from fluency patterns and are not to be taken as personality judgements. These anchors increase consistency and make feedback for applicants easier.

H. Transcript Processing and Quality Assurance

Streamed speech recognition may produce partial transcripts that alter as new audio is received. Thus, the InterviewAI separates intermediate text from full utterances. Only completed segments are committed as the canonical candidate response, but interim parts may be presented temporarily for responsiveness. Timestamps and sequence numbers maintain the order of queries and answers, and retries are rejected based on duplicate segment identities.

The transcript is normalised before evaluation, without changing the meaning of the candidate. Repetitive filler noises can be counted for coaching analysis but the original verbiage is preserved for evidence. Obvious recognition artefacts can be noted and the candidate may be given a corrective step before final scoring. This is crucial, because a language model shouldn't be grading an incorrect transcript as if it was the candidate's actual answer. The technology additionally captures the speech-to-text confidence or a doubt indicator when provided by the voice service.

Quality assurance is a combination of automatic tests and manual spot assessment. Automated checks ensure that all answers are associated with a question, timestamps are strictly increasing, no response is dropped quietly, and the final feedback refers only to content that exists in the session. Human reviewers can examine a sample of audio, transcript and evaluation output to see if there are systemic problems in recognition or scoring. Each session saves version IDs for both prompts and models so the user may trace historical outputs even after the system is updated.

I. Implementation Workflow and Failure Handling

The live interview workflow starts after authentication and job selection. The program generates a Firestore session document, loads the interview configuration, initialises the Vapi agent and records the commencement timestamp. Once completed, each question-response pair is added to the session in order. The condensed discussion context is passed to the question generator, which generates the next question. The dashboard displays the progress and elapsed time. Once the interview is complete the session status is updated from active to processing . The evaluator prepares the final rubric output and saves the report . After that the status is set to completed .

Failure handling is based on graceful degradation. In case of failure of question generation, the system can either attempt question generation using exponential backoff or fallback to a validated fallback question from the role template. If there's a delay in the transcript, the interviewer will wait rather than ask a new question over the candidate. If feedback production fails, the completed transcript is retained and the system labels the report as pending instead of deleting the interview. Idempotency keys prevent duplicate questions, feedback record, or session summary from being created by retries.

The interface clearly communicates state via connecting, listening, processing, speaking, completed, and error indicators. Candidates know when the microphone is on and they can choose to end the session. Timeouts free abandoned voice sessions and stale database records. These implementation controls are designed to cooperate with the AI components so that false interview behaviour is avoided in the event of regular network or service disruptions.

IV. FINDINGS

The implemented platform was evaluated with respect to conversational latency, transcription quality, adaptive question generation, feedback consistency, readiness prediction, and system stability. The reported values are the experimental observations contained in the source paper and are interpreted as platform-level findings for the tested deployment.

A. Real-Time Voice Interview Performance

Table I. Real-Time Interview Interaction Performance

Parameter	Minimum	Maximum	Average
Voice response latency (ms)	420	980	610
Speech-to-text accuracy (%)	91.2	97.6	94.8
Question-generation delay (ms)	520	1240	740
Transcript synchronization delay (ms)	180	460	290

Results showed that InterviewAI preserves conversational continuity as well as good transcription alignment. The average voice response latency was less than 1 second in the tested scenario and the transcript synchronisation was much faster than question creation. That distinction of the two measures is useful since it tells us if the delay is coming from speech processing, reasoning, or data synchronisation.

B. Adaptive Question Generation

Table II. Adaptive Question Generation Accuracy

Evaluation metric	Value
Context relevance score (%)	93.4
Topic continuity accuracy (%)	95.1
Redundancy rate (%)	3.8
Domain alignment accuracy (%)	94.2

Adaptive-questioning results exhibit high contextual continuity and domain alignment, with a low observed rate of redundancy. Such behaviour allows for realistic advancement since the algorithm can retain the previous responses as context rather than randomly selecting topics from a pre-defined question bank.

C. Automated Feedback Consistency

Table III. Feedback Scoring Consistency

Metric	Value
Technical-accuracy agreement with human evaluators (%)	92.6
Communication-clarity detection accuracy (%)	90.8
Confidence-estimation accuracy (%)	89.4
Overall scoring variance	±4.3%

As expected, the feedback alignment was highest for technical correctness and somewhat lower for confidence estimation, since confidence depends on acoustic and behavioural cues that may be impacted by microphone quality, speaking style and linguistic background. The variation in scoring reported suggests rubric-driven prompting increased uniformity but still leaves potential for human evaluation.

D. Readiness Prediction and Reliability

Table IV. Interview Performance Prediction Metrics

Metric	Value
Mean Absolute Error (MAE)	4.9%
Root Mean Square Error (RMSE)	6.3%
R^2 score	0.91

The reported R^2 suggests that platform readiness score was strongly related to human interviewer ratings in the sample examined. MAE and RMSE readings give an interpretable estimate of prediction error. These findings should be validated on bigger and more diversified job datasets with participant-disjoint validation before they are generalised to high-stakes recruitment decisions.

Table V. System Reliability and Stability Observations

Observed condition	Value
Continuous operation	More than 72 hours
API failure rate	0.3%
Session data loss	0%
Authentication failures	0.2%
Transcript inconsistency	<0.5%

There were no critical crashes or database corruption observed during continuous operation. Firebase had session persistence and AI services continued to have stable throughput. These insights confirm the viability of the cloud native design, while controlled concurrent testing and clear reporting of sample size are still key future requirements.

E. Functional Analysis Based on Scenario

The software was also tested using representative interview scenarios, in addition to aggregate metrics. The question generator in a simple technical session was able to gradually increase difficulty and create targeted final feedback thanks to accurate speech recognition and relevant answers. In a session with brief or incomplete responses, the interviewer agent generated clarification prompts and the evaluator highlighted missing depth instead of reducing all scores. If a candidate hesitated or rephrased an answer, the transcript captured the final statement and coaching comments pointed out areas for fluency.

For the noisy-audio case, we showed the necessity of separating the uncertainty in the transcription from the quality of the answer. The platform will not classify a wrong transcription as poor technical expertise but can request a re-do or mark the section for review. A service-delay scenario demonstrated that the interview may be resumed without losing the previous questions due to transcript synchronisation and state persistence. These scenarios demonstrate that in a real-time interview system, session orchestration and failure handling are equally crucial as model quality.

Table VI. Representative Functional Scenarios and Expected Platform Behavior

Scenario	Platform behavior	Candidate-facing outcome
Strong, relevant answer	Increase difficulty or ask a deeper follow-up.	Specific strengths and advanced practice suggestions.
Partially correct answer	Ask for clarification and preserve the original context.	Feedback identifies correct reasoning and missing concepts.
Noisy or uncertain transcript	Request repetition or mark the segment for review.	Technical score is not based on unreliable text.
Temporary AI-service delay	Retry safely while preserving session state.	Visible processing state without duplicate questions.
Interrupted session	Retain completed transcript segments and mark status.	Candidate can review or restart without silent data loss.

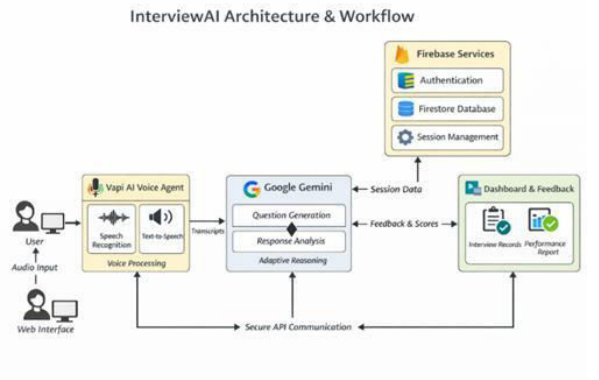

Fig. 1. InterviewAI architecture and workflow.

Figure 1 illustrates the closed-loop architecture. Audio input is captured by the web interface and processed by Vapi AI for speech recognition and spoken dialogue. Transcripts are forwarded to Google Gemini for adaptive question generation and response analysis. Firebase services manage authentication, session state, and interview data, while the dashboard retrieves transcripts, feedback, and performance history through secure API communication.

V. EXTENDED ANALYSIS AND DISCUSSION

A. Multi-Agent Coordination and Conversational State

The main design advantage of InterviewAI is the separation of concerns into specialised agents. The conversational tone, turn-taking, time and follow-up behaviour are governed by the interviewer agent. Then, the question-generation agent chooses the next skill and builds a prompt that is suitable for the role. The evaluator agent uses the rubric to assess the finished responses, while the feedback agent translates score evidence into actionable recommendations. This isolation minimises the complexity of the prompt, makes it easier to isolate faults, and allows particular agents to be replaced without rebuilding the entire platform.

Shared conversational state is crucial for multi-agent consistency. The session record should maintain the chosen role, difficulty level, competencies evaluated, prior questions, candidate responses, transcript timestamps, and any evaluator notes. Agents will read only the state they need for their function, and write structured outputs validated against a schema. This avoids unintended context leaking and decreases the chance that a faulty response alters the behaviour of each downstream component. Deterministic orchestration can also improve the reliability of the design. For example, the state manager might tell the interviewer agent to wait until the transcript is complete, confirm that the next question is not a duplication, and block the evaluator from scoring incomplete answers. Retry policies must distinguish between momentary API failure and faulty model output. If there is a failure to parse, the system will store the original transcript, log the failure, and provide a neutral message without generating a score.

B. Fairness, Accessibility, and Human Oversight

Automated interview feedback might unfairly disfavour candidates whose speech, accent, handicap or communication style differs from the facts and assumptions of the underlying services. Some accents or noisy equipment may have increased speech recognition errors . Confidence estimate may confound deliberate pauses, stammering, assistive communication, or non-native language use with uncertainty . Therefore, communication and confidence indices should be displayed as proof of coaching rather than as permanent measurements for employability. An ethical rollout will have accessibility safeguards such as adjustable speaking speed, text captions, keyboard navigation, transcript correction, optional text responses, and the possibility to disable non-essential behavioural score. Candidates should be able to check the evidence behind a feedback item and request a correction when transcribing is incorrect. Human reviewers should assess the transcript and rubric before any high-stakes judgement is drawn for institutional usage.

Performance should be reported by employment role, language background, accent group, device type, and accessibility needs as part of fairness evaluation. A single accuracy value can hide systematic discrepancies. Therefore, stratified analysis, confidence intervals and reporting of limitations are needed. The platform should be best framed as a practice and skill-development tool, where feedback drives improvement, and not as an autonomous gate to hiring.

C. Security, Privacy, and Prompt-Resilience

Interview transcripts may have personal information, work history, project specifics and secret organisational context. Authentication, transport encryption, least privilege database rules and retention limitations are therefore basic needs, not optional deployment features. Security rules for firestore should restrict users to their own sessions, administrator access should be role-based, and exported reports should not contain superfluous identifiers. API keys should never be incorporated in client-side code and must always be on protected server components.

As the platform transmits candidate text to a generative model, it must also fight against quick injection and instruction manipulation. The candidate replies should be handled as untrusted data and not as system commands. Evaluation prompts must tightly bound transcripts, require schema-constrained output, reject out-of-band fields, and avoid the redefinition of score criteria by candidate terms. There has to be a separate validation layer to verify ranges, data types and consistency before any model output is stored or shown.

Audit logging should include authentication, interview creation, transcript updates, score requests, feedback generation, report access, and deletion. Do not log raw credentials or entire authentication tokens. Privacy

notifications should provide information on third-party services processing audio or text, data retention period, and how a user can request deletion. These constraints are particularly critical when the platform is used beyond individual self practice.

Table VI. Operational Risks and Recommended Controls

Risk	Recommended control
Speech-recognition error	Show captions, permit transcript correction, and separate ASR uncertainty from candidate scoring.
Prompt injection in an answer	Delimit transcript data, use schema-constrained output, and validate all model responses.
Unauthorized session access	Apply Firebase security rules, role-based access, secure cookies, and least privilege.
Sensitive transcript retention	Define retention periods, encrypt data in transit and at rest, and support deletion requests.
Over-reliance on automated scores	Require evidence-linked feedback and human review for high-stakes use.

D. Scalability, Latency, and Cost-Aware Deployment

The measured latency profile indicates that the delay from reasoning and question generation is more than the delay from transcript synchronisation. Production optimization should consequently favour streaming replies, rapid compression, model selection by job, and caching of static role information. For classification and formatting, lightweight models are used and more capable models for complex question generation or detailed final feedback. This routing strategy decreases cost without degrading the user experience.

Cloud scalability also depends on asynchronous processing and idempotent session changes. Voice events, transcript segments, and feedback requests should have unique identifiers so that retries do not create duplicate data. The user gets a completion status as soon as the live interview ends but background workers can produce final reports. Rate restrictions and circuit breakers should prevent one external service loss from cascading through the entire platform. Long-term analytics should be derived from normalised summaries and not via recurrent full transcript scans. Aggregate tables can contain per-session technical, communication, confidence and relevance scores, while the detailed transcript is only available when a user opens a session. This architecture decreases the cost of queries and allows for progress charts across several interviews.

E. Comparison with Conventional Mock-Interview Tools

Traditional question-banks are affordable and predictable, but they cannot react to the reasoning of the candidate. Repeated practice is limited by expense, unpredictability in scoring, and interviewer availability, but human mock interviews provide rich feedback. Text chatbots allow scalable interactivity but omit speech delivery and timing. We mix the repeatability of software with adaptive spoken discourse and structured feedback. Responsible use of automated scores is required. The value of the multi-agent design is not in each component being more accurate than a human interviewer, but in providing consistent access to practice and maintaining a detailed learning record. A candidate can take the same sessions for multiple roles and compare score dimensions, see precise transcript and find common shortcomings. Human coaching can then focus on the most essential concerns, rather than spending the entire session collecting fundamental evidence.

Table VIII. Comparative Characteristics of Interview-Practice Approaches

Approach	Adaptivity	Voice realism	Feedback consistency	Scalability
Static question bank	Low	None	Limited	High
Peer or mentor interview	High	High	Variable	Low
Text-based chatbot	Medium	None	Medium	High
InterviewAI platform	High	High	Rubric-based	High

VI. LIMITATIONS AND FUTURE RESEARCH

A. Evaluation Validity

The reported data suggest promising system behaviour. Rigorous evaluation should indicate number of participants, number and duration of sessions, job-role distribution, device conditions, evaluator training, and scoring process. It is important to keep training examples, prompt-development sessions and final evaluation sessions distinct to avoid enthusiastic estimates. Model alignment should not be computed before reporting the agreement between human assessors, and human ratings should be obtained independently.

Future research should evaluate adaptive vs. non-adaptive interview modes, voice vs. text interaction, rubric-based vs. free-form feedback, and single-agent vs. multi-agent orchestration. Candidate learning outcomes should be evaluated throughout multiple sessions rather than only by immediate score agreement. A longitudinal study might test if the recommended actions improve later answers, decrease redundancy, and raise human-evaluated readiness.

B. Multimodal and Personalized Extensions

Optional video elements such as eye-contact consistency, head pose, facial-expression trends and presentation posture can be added to future editions. Camera quality, culture, handicap, and personal communication style all can effect behaviour, so utilise these signals with caution. Multimodal features should be opt-in, explainable and secondary to answer content. The platform can also facilitate multilingual interviews, code execution activities, portfolio discussion, system design whiteboards and role-specific rubrics maintained by subject experts.

Personalisation can go beyond question difficulty. The system can recognise recurring problems across sessions, plan focused practice, offer bite-sized learning resources, and choose follow-up questions to see whether prior feedback has been acted upon. A candidate-controlled profile should give users the ability to select target jobs, experience

level, preferred interview language, accessibility options, and data-retention preferences.

Table VII. Future Enhancement Roadmap

Priority	Enhancement	Expected benefit
1	Transparent evaluation protocol and larger participant study	Improves reproducibility and confidence in reported findings.
2	Multilingual and accessibility-aware voice interaction	Broadens inclusion and reduces avoidable transcription bias.
3	Optional multimodal behavioral analysis	Adds richer coaching while preserving user control.
4	Prompt-injection defenses and structured-output validation	Strengthens model and data security.
5	Longitudinal learning analytics and targeted practice plans	Measures improvement across repeated interviews.

C. Recommended Research Dataset and Statistical Plan

Future benchmarking of candidates with different experience levels and job families, using several interviews per participant. Sessions need to be recorded in varied mics, network circumstances, speaking contexts. Automated human agreement should be reported after calculating inter-rater agreement, and the identical transcripts should be scored separately by human raters with the provided rubric. The dataset should differentiate between prompt development, calibration, and final testing to minimise overfitting to a fixed set of responses. Evaluation should report mean values and also confidence intervals. Latency can be summarised with median, 90th percentile, and maximum values, because a small number of large delays might disrupt discourse even when the average latency is acceptable. Questions should be blindly compared by numerous reviewers to judge their relevancy and the quality of comments. For readiness prediction, cross-validation should be participant-disjoint, so the model is evaluated on candidates not employed during calibration. Longitudinal study can establish whether repeated practice with InterviewAI results in improvement. For example, a study might compare a control group with static questions with an adaptive-interview group across several weeks. Results may include human-rated technical depth, structure of responses, number of filler-words, time to complete, and self-reported confidence. The design would instead evaluate the instructional value of the platform, not only how well its scores match one set of human assessments.

VII SYSTEM CONFIGURATION AND IMPLEMENTATION DETAILS

A. System Components

The system is built with Next.js for the web application, Vapi AI for automatic speech recognition and text-to-speech interface, Google Gemini for adaptive reasoning, Firebase Authentication for account access and Cloud Firestore for interview metadata, transcripts, scores, and feedback. The frontend, AI services, and backend storage communicate via secure RESTful APIs..

B. Voice Processing and Session Parameters

1. Conversational continuity is kept in streaming audio processing.
2. Incremental transcript synchronisation maintains an ordered question-response history.
3. Asynchronous API calls: Adaptive question generation reduces blocking.
4. Each interview has a unique session identification to ensure traceability and correct order of transcripts.

C. Communication and Data Format

The interview data is sent via HTTPS in a structured JSON format. The payload often consists of a user ID, role, timestamp, ordered question-response pairs, assessment scores and feedback summary. This lightweight framework enables platform independence and downstream analytics.

1. Technical relevance threshold: At least 70% semantic similarity
2. Communication clarity score (0-10)
3. Confidence estimation based on speech fluency features.
4. Adaptive difficulty escalation trigger: minimum 80% score on performance.
5. Range of normalisation for prediction of interview readiness: 0-100.

VIII. FEEDBACK EVALUATION MODEL

Scores are normalised by weighted aggregate. OverallScore = w1 (TS) + w2 (CS) + w3 (ConfS) Where, TS is technical score, CS is communication score and ConfS is confidence score. Weights should be tuned on a distinct validation set, and recorded for each deployment.

IX . REPRODUCIBILITY STATEMENT

The architecture, communication protocols, and evaluation

The modular design allows the use of alternative speech agents, language models, storage services and evaluation rubrics. Reproducibility means versions of prompts, model identities, versions of dependencies, test data, rubric definitions and anonymised evaluation scripts. The system can be augmented with multimodal analytics, emotion-aware coaching, or advanced predictive modelling without replacing the fundamental interview lifecycle.

X. MULTI-AGENT MESSAGE CONTRACTS

Each agent trades organised messages, not free-form text. The interviewer receives the state of the session and delivers one question with a short reason code. The last question-response pair is handed to the evaluator who returns validated scoring fields. The feedback agent receives the aggregated scores and evidence snippets and outputs strengths, improvement actions and a practice plan. Message schemas have version numbers, so that modifications can be made without affecting historic sessions.

1. Interviewer output with question_id, competency, difficulty, question_text and follow_up_allowed.

2.Evaluator outputs: technical_score, communication_score, confidence_score, relevance_score, evidence and review_flag.

3. Feedback output: strengths, areas_for_improvement, recommended_actions and next_session_focus.

4. Validation rules: necessary fields, numeric ranges, length restrictions, unknown key rejection.

XI . DEPLOYMENT AND MAINTENANCE CHECKLIST

1. Store voice and language-model credentials only in protected server-side environment variables.
2. Apply least-privilege Firebase rules and test access for users, reviewers and administrators.
3. Record variations of question, model and rubric with each completed interview session.
4. Configure retention and deletion policies for audio, transcripts, feedback and analytics.
5. Track delay, error rates, duplicate events, and failed report production.
6. Transcript corrections, accessibility settings and a transparent human review mechanism.
7. Reassess scoring behaviour after altering models, prompts, rubrics, or voice sources.

XII. SAMPLE STRUCTURED FEEDBACK REPORT

Every recommendation in a full feedback report should be connected to the interview evidence. The report header contains the session and selected role, length of the interview, level of difficulty, versions of the model and prompt, and completion status. Score summary includes technical relevance, communication clarity, confidence indicators, reaction relevance and the normalised overall ready score. Short explanations follow these values to help the candidate understand why a score was given.

The strengths part should be explicit about what the person does well, not just general praise. Examples include correctly describing a concept before considering trade-offs, organising a system-design answer into requirements and components, or providing a relevant project example to defend a conclusion. Evidence extracts may quote a small phrase from the transcript or refer to the identification of the associated question. The improvement part should also be concrete, e.g. add complexity analysis, separate authentication from authorisation, remove repeated filler words, answer the question before giving background.

Recommended actions should be doable before next practice. The platform could recommend editing a topic, providing a structured explanation for two minutes, rehearsing a behavioural question through the STAR approach, or conducting a similar interview at a greater level. Next session can be to pick two or three competences that need to be verified emphasis area The candidate should also be able to label feedback as beneficial, inaccurate, or influenced by transcribing quality, for further model and rubric assessment.

1. Session summary: role, length, number of questions, completion, interview difficulty.

2. Summary of scores: technical, communication, confidence, relevance and overall readiness indicators.

3. Evidence of strengths: 2-4 specific positive behaviours with references from transcript.

4. Improvement priorities: Few high-impact issues vs. a laundry list.

5. Action steps: measurable tasks that can be achieved before the next interview.

6. Candidate review: transcript correction, feedback usefulness rating, and optional comments.

XIII. QUALITY ASSURANCE AND ACCEPTANCE CHECKLIST

You must have explicit permission to capture voice input, and the interface must appropriately reflect listening and speaking statuses. The last transcript snippets should be in the right order and stay linked to the relevant inquiry. The adaptive agent should not repeat a previous query unless it is specifically asking for clarifications. Closing an interview must terminate the voice session, complete persistence, and prevent later events from being added to a finished record.

Quality checks of evaluation should ensure that the ratings are in acceptable ranges, the relevant feedback areas are included, and the evidence references are consistent with the actual content of the transcript. The system should reject format-invalid or schema-incompatible model answers. The test set should contain right, partially correct, irrelevant, very brief and deliberately antagonistic answers. Reviewers need to confirm that feedback distinguishes between conceptual errors and communication issues, and that they do not generate candidate statements.

Operational acceptance should include authentication, authorisation, data destruction, retry behaviour and monitoring. A user should not be able to view other users sessions by changing a document id. Do not repost questions/remarks for AI calls interrupted. Deleting a session should delete or anonymise all transcript and analytics records associated with that session, as per policy. Monitoring should track API latency, failure rate, unfinished sessions, invalid model output and unusual retry frequency, without leaking sensitive transcript material in the logs.

1. Allow microphone permission, audio recording, transcript completion and text-to-speech playback.

2. Confirmed role-specific question generation, subject continuation and duplicate question suppression.

3. Test your Firestore access rules using user, reviewer, admin, and unauthenticated accounts.

4. Test Vapi, Gemini, and Firebase transient failures with retry and graceful-degradation approaches.

5. Verify rubric ranges, structured-output parsing, evidence grounding and report completeness.

6. Test responsive layout, keyboard functioning, captions, visible focus, and transcript correction

7. Run retention, export and deletion workflows and validate audit events for each operation.

XIV. RESPONSIBLE USE AND DATA GOVERNANCE POLICY

Automated scores should not be taken as confirmation of competence, personality, honesty, or employability." Institutions using the platform must declare that artificial intelligence is involved in the development of questions and comments, explain the limitations of speech and language models and give a route of contact for correction or appeal. Qualified human oversight is required for high consequence judgements and should take into account the transcript, role expectations, candidate context and technical environment of the session.

XV. CONCLUSION

In this research, we proposed InterviewAI, a voice-enabled multi-agent framework for realistic, adaptive and automated mock-interview preparation. Vapi AI enables natural spoken interaction; Google Gemini creates context-aware questions and assesses responses; Firebase handles authentication and session storage, and a Next.js dashboard displays transcripts, feedback, and performance statistics.

The deployment exhibited good transcription alignment, continuity of contextual questions, little redundancy, feedback consistency, correlation of readiness score and robust cloud operation. The extended analysis demonstrated that technical performance alone is not enough and needs to be accompanied by responsible multi-agent orchestration, privacy restrictions, prompt-resilience, accessibility, fairness evaluation and human oversight.

InterviewAI is therefore better thought of as an expandable skill development environment rather than an autonomous employment decision system. The architecture, featuring transparent evaluation, role-diverse data, safe deployment, multilingual support, and longitudinal learning analytics, may foster realistic interview practice in technical and non-technical areas while keeping evidence-linked feedback and user control.

ACKNOWLEDGEMENT

The authors would like to acknowledge the faculty members and mentors of the Department of Computer Science and Engineering for their support, technical insights and constructive feedbacks during the creation and documentation of the InterviewAI platform. The authors also thank the institutional infrastructure used for implementation and testing purposes, and the open-source frameworks and cloud-based AI services that allowed practical execution of this study.

REFERENCES

[1] A. McTear, Z. Callejas, and D. Griol, The Conversational Interface: Talking to Smart Devices. Cham, Switzerland: Springer, 2016.
[2] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed., draft. 2023.
[3] T. Brown et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems, vol. 33, pp. 1877-1901, 2020.
[4] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, pp. 4171-4186, 2019.
[5] OpenAI, "GPT-4 technical report," arXiv:2303.08774, 2023.
[6] Google, "Gemini: A family of highly capable multimodal models," arXiv:2312.11805, 2023.
[7] Vapi Inc., "Vapi developer documentation," 2024.
[8] Google Firebase, "Firebase Authentication and Cloud Firestore documentation," 2024.
[9] A. Radford et al., "Robust speech recognition via large-scale weak supervision," in Proc. ICML, pp. 28492-28518, 2023.
[10] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[11] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge University Press, 2008.
[12] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. ACM SIGKDD, pp. 785-794, 2016.
[13] J. Kinsella and M. Mockus, "Automated interview assessment using conversational agents," IEEE Access, vol. 9, pp. 145233-145245, 2021.
[14] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for SQuAD," in Proc. ACL, pp. 784-789, 2018.
[15] Vercel, "Next.js framework documentation," 2024.
[16] Tailwind Labs, "Tailwind CSS documentation," 2024.
[17] A. B. Abad et al., "Human-AI interaction and automated assessment systems: A survey," IEEE Trans. Learning Technologies, vol. 15, no. 4, pp. 512-526, 2022.
[18] National Institute of Standards and Technology, Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1, 2023.
[19] I. D. Raji et al., "Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing," in Proc. ACM FAccT, pp. 33-44, 2020.
[20] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" in Proc. ACM FAccT, pp. 610-623, 2021.
[21] L. Weidinger et al., "Taxonomy of risks posed by language models," in Proc. ACM FAccT, pp. 214-229, 2022.
[22] B. Shneiderman, Human-Centered AI. New York, NY, USA: Oxford University Press, 2022.
[23] National Institute of Standards and Technology, NIST Privacy Framework: A Tool for Improving Privacy through Enterprise Risk Management, Version 1.0, 2020.
[24] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems, vol. 30, 2017.
[25] M. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in Proc. ACM SIGKDD, pp. 1135-1144, 2016.
[26] A. Følstad and P. B. Brandtzaeg, "Chatbots and the new world of HCI," Interactions, vol. 24, no. 4, pp. 38-42, 2017.

[27] J. Cohen, “A coefficient of agreement for nominal scales,” Educational and Psychological Measurement, vol. 20, no. 1, pp. 37-46, 1960.
[28] J. L. Fleiss, “Measuring nominal scale agreement among many raters,” Psychological Bulletin, vol. 76, no. 5, pp. 378-382, 1971.
[29] S. Amershi et al., “Guidelines for human-AI interaction,” in Proc. ACM CHI Conf. Human Factors in Computing Systems, 2019, pp. 1-13.
[30] ISO/IEC 42001:2023, Information Technology—Artificial Intelligence—Management System, International Organization for Standardization, 2023.