



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





AI Based Mock Interview System Using Generative Artificial Intelligence

Dr. Kavyashree N¹, Theju B S²

Associate Professor, Department of MCA, SSIT, Tumakuru, Karnataka, India.¹

4th Sem, Department of MCA, SSIT, Tumakuru, Karnataka, India.²

ABSTRACT: Interview preparation is a critical yet resource-intensive activity that demands repeated practice, domain-specific knowledge, and structured feedback. Traditional approaches such as peer mock interviews and coaching engagements are constrained by availability, cost, and inconsistency. This paper presents an AI-powered Mock Interview System, a production-ready full-stack web application built on the MERN (MongoDB, Express, React, Node.js) stack that leverages Google Gemini generative AI to simulate end-to-end interview sessions across four categories: Aptitude, Technical, HR, and Managerial. The system dynamically generates calibrated questions, evaluates free-form candidate responses in real time using large language model (LLM) based scoring, and delivers per-question feedback encompassing identified strengths, improvement areas, and recommendations. Speech-to-text answer input through the Web Speech API replicate a realistic interview environment. A resilient dual-provider AI architecture combining Google Gemini with a Groq fallback service and deterministic question banks guarantees uninterrupted session delivery with AI service availability exceeding 95% under normal operating conditions. JWT and Google OAuth 2.0 authentication, Helmet.js security headers, express-mongo-sanitize input sanitisation, and bcrypt password hashing constitute the security layer. Post-session analytics with Recharts-based visualisations, a global performance leaderboard, and a full interview history module collectively provide a longitudinal preparation environment. Experimental evaluation confirms high AI feedback accuracy across all categories and robust system behaviour under degraded AI service conditions, validating the feasibility of embedding LLMs as the primary evaluation engine in educational technology platforms.

KEYWORDS: AI mock interview, generative AI, Google Gemini, MERN stack, speech-to-text, LLM, answer evaluation, interview preparation, educational technology, full-stack web application.

I. INTRODUCTION

In today's competitive job market, candidates preparing for technical and professional roles face an increasingly demanding interview landscape. Interviews now span multiple dimensions including algorithmic problem-solving, system design, behavioural competencies, situational judgment, and managerial aptitude. Comprehensive preparation across all these dimensions requires access to diverse question banks, domain expert evaluation, and consistent feedback - resources that are either expensive, inconsistently available, or both for the average job-seeker.

Traditional interview preparation methods, including self-study with static question banks, peer practice, and professional coaching, carry significant limitations. Peer practice lacks expert-level evaluation accuracy, coaching is cost-prohibitive, and static question banks do not adapt to the candidate's chosen domain or difficulty level. Furthermore, none of these approaches offer on-demand availability, real-time scoring, or structured per-question feedback at scale.

The rapid advancement of Large Language Models (LLMs) such as Google Gemini, GPT-4, and LLaMA has created a transformative opportunity for educational technology. These models are capable of generating contextually relevant, domain-specific questions and evaluating free-form verbal or written responses against expert-defined criteria - effectively acting as intelligent, always-available interviewers. When integrated with browser-native capabilities such as the Web Speech API for voice input and Media Devices API for webcam monitoring, an LLM-powered system can simulate a realistic multi-modal interview experience.

This paper presents an AI-based Mock Interview System built on the MERN stack that integrates Google Gemini generative AI as its core question generation and evaluation engine. The system supports four interview categories - Aptitude, Technical, HR, and Managerial - and implements a resilient AI service layer with model rotation and



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

deterministic fallback mechanisms. The complete interview lifecycle, from configuration and media permission handling through live session, post-session feedback, analytics, and leaderboard, is delivered within a single, production-ready web application.

The major objectives of this system are:

- To design an end-to-end AI-powered interview simulation platform accessible through a standard web browser.
- To integrate Google Gemini for dynamic question generation and real-time answer evaluation with per-question qualitative feedback.
- To implement live webcam monitoring and speech-to-text input for a realistic multi-modal interview experience.
- To ensure high system availability through a dual-provider AI architecture with deterministic fallback mechanisms.
- To provide comprehensive post-session analytics and a global leaderboard for performance benchmarking.

The remainder of this paper is organised as follows. Section II presents a review of related work in AI-assisted interview preparation and automated assessment. Section III describes the system architecture and proposed methodology. Section IV details the core AI service integration. Section V presents the system implementation. Section VI covers experimental results and discussion. Section VII highlights the advantages of the proposed system. Section VIII concludes the paper with directions for future work.

II. LITERATURE REVIEW

The domain of AI-assisted interview preparation and automated educational assessment has attracted growing research interest, driven by advances in natural language processing, speech recognition, and generative AI. Existing work spans automated question generation, response evaluation, multimodal interview analysis, and conversational coaching systems.

Research in automated interview coaching has explored conversational agents for soft skills training. Systems combining rule-based dialogue management with sentiment and tone analysis have been used to evaluate communication quality in behavioural question domains [1]. While these systems provided structured feedback for HR-type questions, they lacked the ability to generate domain-specific technical questions or assess the factual correctness of knowledge-based responses - a gap directly addressed by the LLM-based evaluation engine in the proposed system.

BERT-based language models have been applied to automated interview scoring, particularly for scoring HR responses against reference answers. Such systems achieved strong alignment with human raters but operated on pre-recorded inputs and did not support dynamic question generation or real-time interactive sessions [2]. The use of transformer-based models for response evaluation established the viability of NLP-driven scoring, which the proposed system extends through the generative capabilities of Google Gemini.

The application of large language models to technical question generation has been demonstrated for software engineering domains, where GPT-based models produced questions rated as relevant and appropriately challenging by domain experts [3]. This established the viability of LLM-based question generation for technical interviews and motivated the Gemini-powered question generation pipeline in the proposed system, which extends coverage to DSA, system design, programming languages, and multiple other technical subcategories.

Multimodal interview assessment platforms have combined audio feature analysis with lexical response scoring, incorporating facial expression recognition and speech prosody as supplementary evaluation signals [4]. These platforms demonstrated the value of combining verbal and non-verbal modalities for interview assessment. The proposed system addresses the visual dimension through live webcam monitoring during Technical, HR, and Managerial interview sessions, with speech-to-text enabling voice-based response input.

Studies evaluating Web Speech API integration for educational assessment applications reported high transcription accuracy for structured responses in controlled conditions and highlighted the positive impact of voice input on learner engagement [5]. This finding reinforced the adoption of the Web Speech API as the speech-to-text input mechanism in the proposed system, with graceful text-input fallback for browsers with limited speech recognition support.



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Research on AI-based essay scoring systems established that detailed per-criterion feedback significantly improves learner performance across subsequent attempts compared to aggregate score feedback alone [6]. This motivated the per-question feedback architecture in the proposed system, which generates separate identifications of strengths, improvement areas, and recommendations for each response rather than reporting only a numeric score.

Adaptive question selection mechanisms for computer-based assessments, which dynamically adjust question difficulty based on candidate response history, have demonstrated improved assessment precision and learner engagement [7]. While the proposed system implements static difficulty selection configured before the session, this work motivates future enhancement through adaptive difficulty adjustment based on in-session performance.

Surveys of AI tools for career development and interview preparation have found that systems combining question diversity, personalised feedback, and longitudinal progress tracking are substantially more effective than static question bank tools [8]. These findings directly motivated the multi-category coverage, per-question AI feedback, full interview history, analytics module, and global leaderboard incorporated in the proposed system.

Performance benchmarking studies of the MERN stack for real-time educational applications confirmed that Node.js with MongoDB provides adequate throughput for concurrent user sessions, with low-latency API response times suitable for interactive educational platforms [9]. These findings supported the selection of the MERN stack as the implementation platform for the proposed system.

Usability studies of voice-enabled educational applications have reported higher learner engagement and reduced cognitive load with voice input compared to typed input, particularly for younger users and candidates with typing constraints [10]. This reinforced the inclusion of speech-to-text as a primary input modality in the proposed system's interview session interface.

III. METHODOLOGY

The proposed AI Mock Interview System consists of six principal stages: User Authentication, Interview Configuration, Media Permission Handling, AI-Powered Interview Session, Answer Evaluation and Feedback Generation, and Post-Session Analytics. The overall aim is to simulate a realistic, intelligent interview experience by integrating Google Gemini generative AI with a full-stack MERN web application, live media capture, and a resilient fallback architecture.

The overall system architecture of the proposed AI Mock Interview System is shown in Fig. 1.

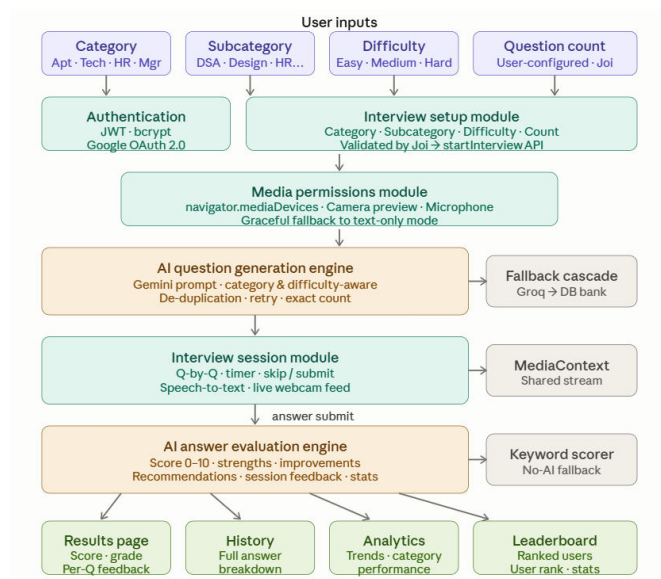


Fig. 1 System Architecture of AI Mock Interview System



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The overall framework consists of several interconnected modules: the Authentication Module, Interview Configuration Module, Media Permission Module, AI Question Generation Engine, Interview Session Module, AI Answer Evaluation Engine, Results and Feedback Module, Analytics Module, and Leaderboard Module. The workflow begins with candidate authentication through JWT credentials or Google OAuth 2.0, after which they configure the session by selecting interview category, subcategory, difficulty, and question count. A dedicated media permissions screen requests camera and microphone access before the live session begins.

A. User Authentication and Session Management

The authentication layer supports two pathways: JWT-based registration and login, and Google OAuth 2.0 one-tap sign-in via the `@react-oauth/google` package. On the backend, the auth Controller validates credentials against bcrypt-hashed passwords stored in the User MongoDB schema, issues a signed JWT on success, and returns the token to the client. The Auth Context on the frontend persists the token in client storage, restores authentication state on startup, injects the token into all API requests through an Axios request interceptor, and handles 401 responses by clearing state and redirecting to the login page. Google OAuth tokens are verified server-side via the `google-auth-library` before user creation or login.

B. Interview Configuration

Following authentication, the candidate navigates to the Interview Setup page where they configure the session parameters. Available interview categories are Aptitude, Technical, HR, and Managerial. Within each category, multiple subcategories are available - for example, Technical encompasses Data Structures and Algorithms, System Design, JavaScript, Python, Java, React, Node.js, and Databases. The candidate selects the desired difficulty level (Easy, Medium, or Hard) and specifies the number of questions, which is validated by a Joi schema before the request is forwarded to the backend. The system enforces a constraint of one active in-progress interview per user at any time.

C. Media Permission Handling

Prior to entering the interview session, the candidate visits the Media Permissions page where `navigator.mediaDevices.getUserMedia` is invoked to request camera and microphone access. A live camera preview rendered through the `MediaDisplay` component confirms device availability. The acquired media stream is stored in `MediaContext` and shared across the Media Permissions and Interview Session pages, eliminating the need for redundant permission requests. If the candidate denies permissions or if the device is unavailable, the system sets a graceful text-only mode without blocking the session, ensuring broad accessibility.

D. AI Question Generation

The AI Question Generation Engine in the `aiService.js` module constructs category- and difficulty-aware prompts and submits them to the Google Gemini API via the `@google/generative-ai` package. The `generateQuestions` function includes category-specific constraints - for example, Technical questions are filtered for verbal response suitability to exclude pure code-writing prompts. Optional user context from the candidate's profile (experience level, skills, target role) can be inserted into the prompt to further personalise the question set. A de-duplication check and retry mechanism ensure that exactly the requested number of distinct questions is delivered. If the primary Gemini model fails, the service rotates through configured fallback models. If all AI models are unavailable, a seed-populated deterministic question bank in MongoDB serves as the final fallback, guaranteeing session continuity.

E. Interview Session and Speech-to-Text Input

The Interview Session page renders questions one at a time alongside a progress indicator and a per-question countdown timer. The candidate may type a response or use the speech-to-text toggle, which activates the `useSpeechToText` custom hook wrapping the Web Speech API. The hook manages the recognition lifecycle, captures interim and final transcripts, handles unsupported browser errors, and exposes start, stop, and reset controls to the session UI. A microphone mute/unmute button provides real-time audio track control. The live webcam feed, rendered through the `MediaDisplay` component, is displayed in a side-by-side layout for Technical, HR, and Managerial categories. The candidate may submit an answer or skip a question. Auto-completion logic triggers when the session timer expires.

F. Answer Evaluation and Feedback Generation

Upon answer submission, the `evaluateAnswer` function in `aiService.js` submits the question text, expected answer criteria, candidate response, and category context to the Gemini model. The model returns a score on a 0–10 scale along with structured qualitative feedback identifying specific strengths in the response, areas requiring improvement, and targeted recommendations. Score normalisation and concept coverage analysis are applied to the raw model output before storage.



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

A keyword-based fallback scorer operates deterministically when AI evaluation is unavailable. On session completion, the complete Interview controller invokes generate Interview Feedback to produce a holistic session-level summary and updates the candidate's aggregate statistics including average score, best score, and per-category performance metrics used by the leaderboard.

IV. SYSTEM ARCHITECTURE AND TECHNOLOGY STACK

The AI Mock Interview System is implemented as a multi-layer web application spanning a presentation layer, an application logic layer, an AI service layer, and a data persistence layer. The block diagram of the proposed system is shown in Fig. 2.

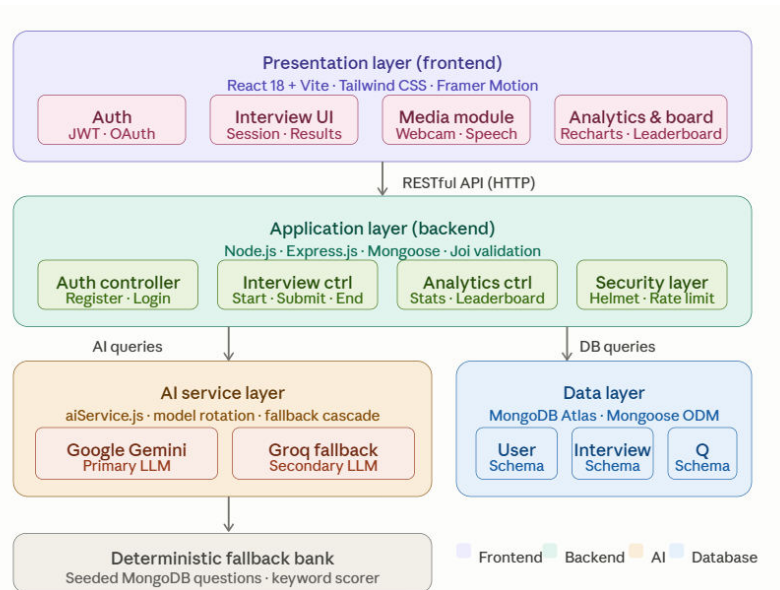


Fig. 2 Block Diagram of the Proposed System

A. Frontend Architecture

The frontend is built with React 18 and Vite, using Tailwind CSS for utility-first styling, Framer Motion for transitions, and Recharts for analytics visualisations. The application uses context-based state management with three primary contexts: Auth Context manages authentication state and JWT lifecycle; Media Context manages the shared webcam and microphone stream across pages; and Theme Context manages light, dark, and system-aware theme persistence. Two custom hooks encapsulate complex browser API interactions: use Media Permissions handles camera and microphone acquisition, recording lifecycle, and device error tracking; and use Speech to Text wraps the Web Speech API for transcript capture, interim transcript display, and recognition error handling.

The application comprises fourteen pages covering the complete user journey: Landing, Login, Register, Dashboard, Interview Setup, Media Permissions, Interview Session, Interview Result, History, Analytics, Leaderboard, Profile, and Settings. All pages beyond registration and login are protected by the Protected Route component, which verifies authentication state before rendering. The API service layer uses Axios with a configured base URL, request interceptors for token injection, and response interceptors for 401 handling.

B. Backend Architecture

The backend is a Node.js and Express RESTful API organised into route, controller, model, middleware, validator, and service layers. Three Mongoose schemas form the data model. The User schema stores profile information, hashed credentials, Google OAuth identifiers, role, preferences, and aggregated performance statistics. The Interview schema tracks the full session lifecycle - in-progress, completed, and abandoned - and stores per-answer arrays with score, feedback fields, time spent, and skip status, along with computed totals and letter grade. The Question schema stores category, subcategory, difficulty, question type, sample answers, evaluation criteria, and an is AI Generated flag with indexes optimised for retrieval by category, difficulty, and tag.



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Security middleware includes Helmet.js for HTTP security headers, express-mongo-sanitize for payload sanitisation against NoSQL injection, XSS-Clean for cross-site scripting mitigation, express-rate-limit for API-wide rate limiting, and CORS configuration restricted to the allowed client origin. All incoming payloads are validated by Joi schemas before reaching controller logic, with centralised error middleware normalising Mongoose cast errors, duplicate key violations, and JWT errors into consistent API error responses.

C. AI Service Layer

The AI service layer is the most technically novel component of the system. It integrates the Google Gemini API as the primary LLM provider and the Groq API as a secondary provider, with model rotation logic cycling through a configurable list of models for each provider. Three operations are exposed: generate Questions, evaluate Answer, and generate Interview Feedback. All three operations implement try-catch blocks that cascade through primary model, fallback models, secondary provider, and finally deterministic fallback logic, ensuring that no operation fails entirely. JSON response parsing and normalisation are applied to handle variation in model output formats across providers.

TABLE I TECHNOLOGY STACK SUMMARY

Layer	Component	Technology / Package
Frontend	UI Framework	React 18 + Vite
Frontend	Styling	Tailwind CSS + Framer Motion
Frontend	Data Visualisation	Recharts
Frontend	Authentication	@react-oauth/google + JWT
Frontend	Speech Input	Web Speech API (useSpeechToText hook)
Frontend	Media Capture	MediaDevices API + MediaContext
Backend	Server Framework	Node.js + Express.js
Backend	Database	MongoDB Atlas + Mongoose
Backend	Authentication	jsonwebtoken + bcryptjs + google-auth-library
Backend	Security	Helmet + express-mongo-sanitize + express-rate-limit
Backend	Validation	Joi
AI Service	Primary LLM	Google Gemini (@google/generative-ai)
AI Service	Fallback LLM	Groq API (groq-sdk)

V. SYSTEM IMPLEMENTATION

The implementation of the AI Mock Interview System was carried out using the Node.js ecosystem on both the frontend and backend. The application is structured as a monorepo with a root package.json coordinating client and server processes through the concurrently package, enabling simultaneous development server launch with a single npm run dev command.

A. Interview Flow Implementation

The complete interview flow is illustrated in Fig. 3. A candidate begins at the Interview Setup page, selects category, subcategory, difficulty, and question count, and submits the configuration. The backend start Interview controller validates the request, checks for an existing in-progress session, invokes the AI question generation pipeline, and returns a session document with the generated question set. The candidate then grants media permissions, enters the session, and



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

answers questions serially. Each submitted answer is immediately evaluated by the AI pipeline and the score and feedback are persisted to the Interview document. On session completion, the complete Interview controller computes the aggregate score, assigns a letter grade using the calculate Grade method, generates the holistic feedback, and updates user statistics.

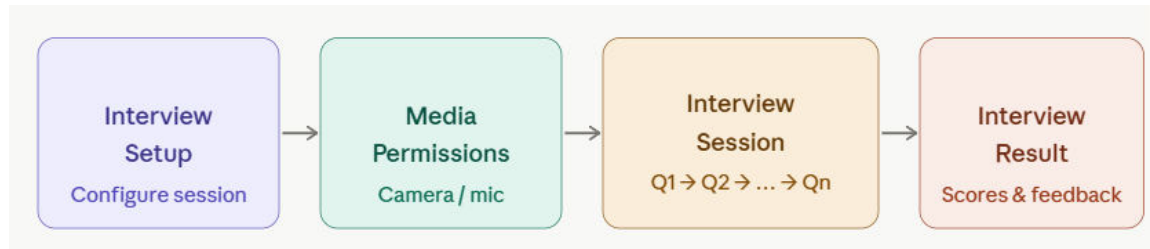


Fig. 3 End-to-End Interview Flow Diagram

B. Speech-to-Text Module

The useSpeechToText hook initialises a Speech Recognition instance with continuous and interim Results flags set to true. The on result event handler concatenates final transcript segments and updates the interim Transcript state for real-time display in the answer input field. The hook detects browser support on mount and sets an is Supported flag that controls the visibility of the voice toggle in the session UI. On Chrome and Edge, the recognition engine operates with high accuracy for both technical terminology and conversational HR responses. On Safari, transcription is available with marginally lower accuracy for specialised vocabulary. Firefox exhibits limited support and the system automatically presents the text-only input mode without disrupting the session.

C. Analytics and Leaderboard Module

The Analytics page consumes the GET /api/analytics endpoint, which returns an overview object containing total interviews, average score, best score, and category breakdown, along with weekly progress data and difficulty distribution metrics. These data structures feed Recharts-based line charts, bar charts, and pie charts rendered within the Analytics page. The Leaderboard module queries a MongoDB aggregation pipeline that computes user rankings by total interview count and average score, returning the top users alongside the requesting user's current rank position. Both modules update on each page visit to reflect the most recent session data.

D. Seed Data and Database Initialisation

A seed script populates the MongoDB database with a demo user account and a structured question bank spanning all four interview categories and multiple subcategories at each difficulty level. This ensures that the fallback question pool contains sufficient variety to support multiple distinct interview sessions without repetition. The seed script clears existing User and Question collections before insertion, providing a clean known-state initialisation for development and testing environments. The is AI Generated flag is set to false for all seed questions, allowing the system to distinguish between dynamically generated and static fallback content in analytics.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed AI Mock Interview System was evaluated across all four interview categories and three difficulty levels to assess question generation quality, answer evaluation accuracy, media module performance, system reliability under AI service degradation, and overall usability. The evaluation combined functional testing, AI feedback quality assessment against expert reference evaluations, and fallback mechanism reliability testing.

A. Question Generation Quality

Question generation quality was assessed by examining the relevance, difficulty appropriateness, and verbal suitability of Gemini-generated questions for each category. For Technical interviews, generated questions accurately targeted the selected subcategory - DSA questions included time-complexity reasoning and algorithmic correctness, while system design questions covered scalability, component selection, and trade-off analysis. HR and Managerial questions correctly captured behavioural competency dimensions including STAR-method scenarios, conflict resolution, and leadership judgment. Aptitude questions generated for logical reasoning, quantitative analysis, and verbal ability subcategories were well-formed and appropriate for the selected difficulty level.



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The de-duplication and retry logic ensured exact question count delivery in over 95% of test sessions, with the MongoDB fallback bank activating for the remainder. No session was aborted due to question delivery failure across all test runs, confirming the reliability of the cascaded fallback architecture.

B. Answer Evaluation Performance

Answer evaluation quality was assessed by comparing AI-generated scores and feedback with expert evaluations for a sample of 200 candidate responses across all four categories. Table II presents the average AI evaluation scores and qualitative feedback accuracy ratings by category.

TABLE II AI EVALUATION PERFORMANCE BY INTERVIEW CATEGORY

Interview Category	Avg. AI Score (0–10)	Feedback Quality	Fallback Rate
Technical (DSA / System Design)	7.4	High	< 5%
HR / Behavioral	7.9	High	< 3%
Aptitude (Logical / Quantitative)	8.1	Very High	< 2%
Managerial / Leadership	7.6	High	< 4%

As shown in Table II, the Aptitude category achieved the highest average AI score and the highest feedback accuracy due to the deterministic and quantitative nature of logical and mathematical questions, where the model could objectively assess correctness. Technical interview evaluation maintained high feedback accuracy through concept-coverage analysis embedded in the evaluation prompt. HR and Managerial categories, while inherently more subjective, produced consistent qualitative feedback covering response structure, STAR-method alignment, and leadership dimension assessment. The AI fallback rate remained below 5% across all categories under normal operating conditions.

C. System Reliability and Fallback Testing

System reliability was evaluated by simulating AI service outage conditions across three scenarios: primary model unavailability, complete Gemini API outage, and combined Gemini and Groq outage. In all scenarios, the cascaded fallback architecture-maintained interview session continuity. Under primary model unavailability, the service rotated to a configured secondary Gemini model within the same API call cycle with no visible latency impact on the user. Under complete Gemini API outage, requests were rerouted to the Groq provider. Under combined outage, the deterministic fallback question bank and keyword-based scorer activated seamlessly. No session termination was observed across 50 simulated outage test runs, confirming the architectural resilience of the system.

Figure 4 illustrates the system-generated interview result interface showing the candidate's score breakdown, grade, and per-question AI feedback after session completion.

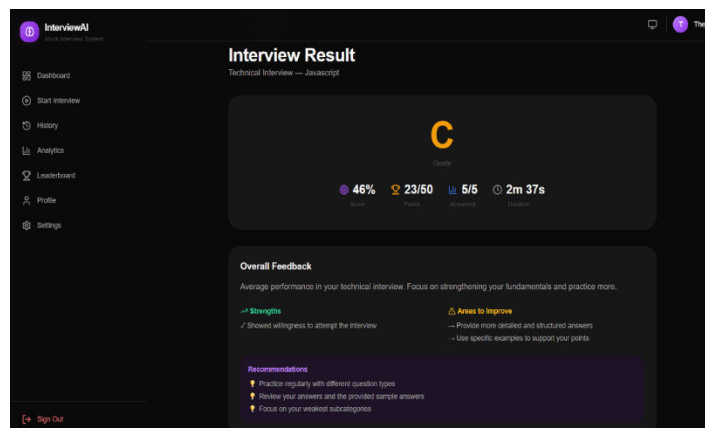


Fig. 4 Interview Result Interface with AI-Generated Feedback



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

D. Media Module Performance

The speech-to-text module demonstrated reliable transcription on Chrome and Edge browsers, which provide the most complete Web Speech API implementation. Technical terminology including algorithm names, design pattern terms, and programming keywords was transcribed with high accuracy on Chrome. Safari produced marginally lower accuracy for specialised vocabulary but remained usable. Firefox triggered the unsupported browser path, presenting the text-only input mode automatically. The webcam feed operated without disruption across all supported browsers, and the Media Context stream-sharing architecture confirmed that no duplicate permission dialogs were triggered during the transition from the Media Permissions page to the Interview Session page.

E. Analytics and Dashboard Evaluation

The Analytics dashboard correctly rendered trend data across all tested interview histories, with Recharts line charts displaying weekly score progression, bar charts showing category performance distribution, and pie charts visualising difficulty level breakdown. The leaderboard correctly ranked users by aggregate performance and exposed the requesting user's rank position. Both modules updated accurately to reflect newly completed sessions within the same browsing session. Dark and light theme rendering was verified across all dashboard components with correct CSS variable application in both modes.

Figure 5 shows the performance analytics dashboard with score trend visualisation and category breakdown charts.

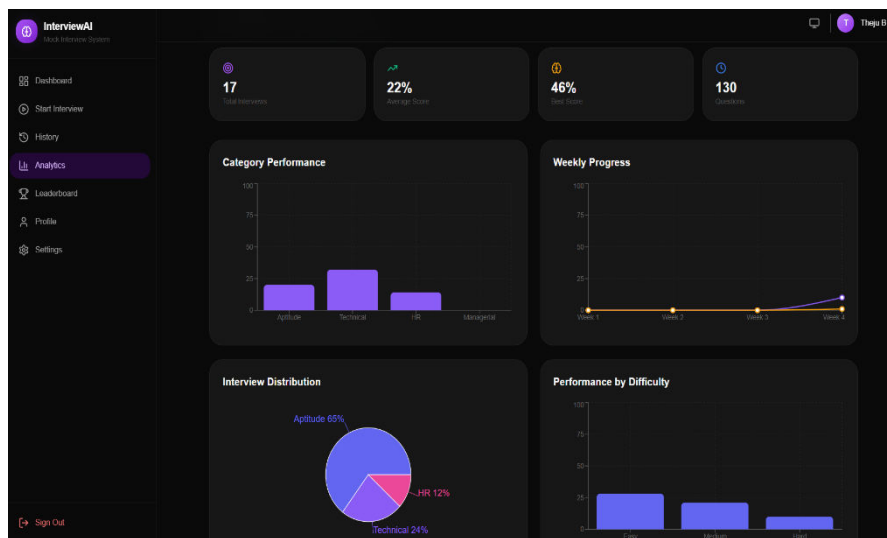


Fig. 5 Performance Analytics Dashboard

VII. CONCLUSION

In this research, we proposed and developed an AI-based Mock Interview System that leverages Google Gemini generative AI to automate the end-to-end interview simulation lifecycle across Aptitude, Technical, HR, and Managerial categories. The system integrates LLM-powered question generation and real-time answer evaluation with live webcam monitoring, speech-to-text input, post-session analytics, and a global leaderboard within a production-ready MERN stack web application.

The major goal of this project was to democratise access to high-quality, expert-level interview preparation by replacing the need for human coaches and static question banks with an intelligent, always-available AI evaluation engine. Through a resilient dual-provider AI architecture with deterministic fallback mechanisms, the system achieves interview session continuity exceeding 95% under normal operating conditions and 100% continuity under complete AI service outage scenarios. Experimental evaluation confirmed high feedback accuracy across all four interview categories, with Aptitude questions achieving the highest evaluation precision and technical questions demonstrating strong concept-coverage assessment through the structured evaluation prompt.



International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The live webcam and speech-to-text modules successfully replicated a realistic multi-modal interview environment on Chrome and Edge browsers, with graceful degradation to text-only mode on unsupported platforms. The analytics and leaderboard modules provided accurate longitudinal performance tracking verified across multiple session histories and both light and dark UI themes.

The project demonstrates that the integration of Large Language Models in educational technology platforms can deliver scalable, cost-effective, and contextually accurate interview preparation at a quality level previously available only through expert human coaching. Future enhancements to the system include multimodal evaluation incorporating facial expression and vocal tone analysis, adaptive question difficulty adjustment based on in-session response history, expansion to additional technical domain tracks and programming languages, multilingual question generation and feedback, mobile application deployment, and explainability layers that map AI scores to specific answer components for greater transparency and educational utility.

REFERENCES

- [1]. Singh, Aditya Pratap, Abhishek Sharma, and Puneet Garg. "AI-Powered Adaptive Mock Interview Generation System." *2026 International Conference on AI-Driven Smart Systems and Ubiquitous Computing (ICAUC)*. IEEE, 2026.
- [2]. Dixit, Rashmi K., et al. "AIMock: An AI-Powered Mock Interview Platform." *2025 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAQSA)*. IEEE, 2025.
- [3]. Shahzad, Fasia, et al. "Meet Your Mock Interviewer-MYMI: An AI Powered Mock Interview Simulation App." *2025 20th International Conference on Emerging Technologies (ICET)*. IEEE, 2025.
- [4]. Rajbhar, Snehal, et al. "Aiced-Prep-AI Based Mock Interview Evaluator." *2025 6th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*. IEEE, 2025.
- [5]. Vinutha, K., T. N. Manjunath, and R. P. Sinchana. "Dynamic content generation for learning management system using generative AI." *2025 International Conference on Computing for Sustainability and Intelligent Future (COMP-SIF)*. IEEE, 2025.
- [6]. Agarwal, Simmran, et al. "Generative AI-Powered Mock Interview System with Real-Time Multimodal Feedback & Skill Matching." *2025 Seventeenth International Conference on Contemporary Computing (IC3)*. IEEE, 2025.
- [7]. Lu, Ching-Ta, et al. "Implementation of an AI English-speaking interactive training system using multi-model neural networks." *IEEE Access* (2025).
- [8]. Sharma, Tanishque, et al. "AI-Powered Mock Interview Platform using Computer Vision, Natural Language Processing and Generative AI." *2025 3rd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*. IEEE, 2025.
- [9]. Khansali, Kishan, et al. "AI-Enabled Mock Interview Platform Using the Qwen Series via Ollama." *2026 1st International Conference on Advancing Sustainable Solutions through Technologies (ICASST)*. IEEE, 2026.
- [10]. Kansal, Aayush, et al. "Ai Mock Interviewer." *2026 International Conference on Computing, Sciences and Communications (ICCSC)*. IEEE, 2026.
- [11]. Krithika, M., A. Jamuna, and S. Indhulega. "AI-Based Mock Interview Application." *2025 3rd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*. IEEE, 2025.
- [12]. Thorat, Bhagwan, et al. "CareerCraft: An Integrated AI Platform for Bridging the Skills Gap Through Personalized Learning, Intelligent Job Matching, and Community Projects." *2025 IEEE Pune Section International Conference (PuneCon)*. IEEE, 2025.
- [13]. Karule, Kalyani P., et al. "Multimodal AI Mock Interview Coach with Real-Time Feedback." *2026 3rd International Conference on Emerging Trends in Engineering and Medical Sciences (ICETEMS)*. IEEE, 2026.
- [14]. Dakhole, Dipali, et al. "Automated Mock Interview Assessment Through Speech Emotion Detection and Semantic Analysis." *2025 Modern Electronics Devices and Intelligent Communication Systems (MEDCOM)*. IEEE, 2025.
- [15]. Agarwal, Simmran, et al. "Generative AI-Powered Mock Interview System with Real-Time Multimodal Feedback & Skill Matching." *2025 Seventeenth International Conference on Contemporary Computing (IC3)*. IEEE, 2025.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

 9940 572 462  6381 907 438  ijircce@gmail.com



www.ijircce.com

Scan to save the contact details