Real-Time AI Voice Interviewing Agent: Mock Interview System for Next-Gen Job Seekers

Kumaran S
Department of CSE
Rajalakshmi Engineering College
Chennai, India
230701159@rajalakshmi.edu.in

Kabilesh P
Department of CSE
Rajalakshmi Engineering College
Chennai, India
230701133@rajalakshmi.edu.in

Mrs. Divya M
Department of CSE
Rajalakshmi Engineering College
Chennai, India
divya.m@rajalakshmi.edu.in

Abstract—**Job interviews are critical, high-stakes events impacting career trajectories. However, effective preparation is challenging due to limitations of existing tools and high coaching costs. Traditional platforms often lack the real-time, dynamic nature of interviews and neglect verbal communication skills. This paper presents a "Real-Time AI Voice Interviewing Agent," an innovative system leveraging Artificial Intelligence (AI) to provide a realistic, adaptive, and accessible mock interview environment. Users engage in voice-based interviews with an AI agent utilizing Natural Language Processing (NLP), Speech-to-Text (STT), and Text-to-Speech (TTS) for natural conversations. The system features an adaptive workflow where the AI adjusts questioning based on candidate responses, role requirements, and skill level. Built with Next.js, Tailwind CSS, Firebase, and the Vapi SDK integrated with Large Language Models (LLMs) like Google Gemini, it offers immediate, comprehensive feedback on performance metrics including pace, fluency, content relevance, and confidence. By providing a voice-first, adaptive, and feedback-rich experience, this AI-powered system aims to democratize interview coaching, empowering candidates to build confidence and succeed in career-defining moments.**

Index Terms—**Conversational AI, Mock Interview, Voice Agent, Natural Language Processing (NLP), Speech-to-Text (STT), Text-to-Speech (TTS), Feedback Generation, Next.js, Firebase, Vapi, Gemini.**

I. Introduction

Securing employment in today's competitive job market hinges significantly on a candidate's ability to perform well during interviews [1]. Interviews serve as the primary gateway for employers to assess not only technical skills but also crucial soft skills such as communication, critical thinking, and cultural fit. Despite the undeniable importance of interview proficiency, many job seekers, particularly recent graduates and career changers, struggle to adequately prepare [2]. This lack of preparation can stem from various factors, impacting confidence and performance.

The existing paradigm for interview preparation presents several challenges. Traditional methods, like reviewing common questions, lack interactivity and objective feedback. While online platforms have emerged, many are text-based, completely missing the vital element of verbal communication [3]. These platforms often present static, one-size-fits-all question banks that do not cater to specific job roles or experience levels. Furthermore, solutions that do offer realistic simulations or human coaching are frequently expensive (often \$50-\$150 per session) or positioned towards recruiters, creating a significant barrier to access. This affordability crisis limits opportunities for crucial practice, disadvantaging many candidates.

The limitations in current methods highlight a distinct market gap: the need for an accessible, realistic, and personalized interview practice tool. The rise of Artificial Intelligence (AI), particularly in Natural Language Processing (NLP) and conversational AI, presents a unique opportunity to address these shortcomings [4]. AI technologies can power systems capable of understanding spoken language, engaging in dynamic conversations, adapting questioning strategies in real-time, and providing nuanced feedback on both the content and delivery of a candidate's responses, closely simulating human interviewers. Such systems promise scalability and consistency unmatched by human coaches alone.

This project proposes the development of a "Real-Time AI Voice Interviewing Agent," a sophisticated mock interview system for job seekers. Leveraging cutting-edge AI, including advanced voice recognition (STT), natural language generation (NLG), and realistic speech synthesis (TTS), this platform aims to revolutionize interview preparation. The AI agent employs adaptive questioning techniques, tailoring the interview flow based on the user's input and target job role. Crucially, immediately following the session, the AI delivers detailed, actionable feedback on fluency, tone, response structure, and content relevance. By providing an affordable, on-demand, and highly realistic practice environment with personalized insights, this system seeks to empower candidates, build their confidence, refine their communication skills, and ultimately enhance their chances of interview success.

II. Literature Review

The application of AI to interview preparation and assessment aims to overcome the limitations of traditional methods by offering scalability and data-driven insights. Early approaches often focused on analyzing textual responses or non-verbal cues from video recordings [1]. Systems utilizing NLP techniques like TF-IDF and early versions of BERT were explored for scoring written answers based on keyword matching and semantic similarity, but missed verbal nuances.

Research involving multimodal analysis, combining video (facial expressions) and audio (speech patterns, tone), sought a more holistic assessment. Systems using Hidden Markov

Models (HMMs) or Support Vector Machines (SVMs) demonstrated moderate success [2], but often struggled with real-time interaction complexity, large labeled dataset requirements, and potential inherent biases in training data.

The advent of more powerful language models, such as RoBERTa, marked a significant advancement in automated content evaluation. Studies found that RoBERTa achieved superior performance in understanding semantic context compared to TF-IDF/BERT, leading to more accurate assessments [5]. Memory networks have also been employed to evaluate the content of short answers [3], though generalizability remains a challenge for longer, complex responses.

Parallelly, the field of conversational AI has matured significantly. Research into dialogue state tracking, using techniques like Long Short-Term Memory (LSTM) networks, aimed to enable AI agents to maintain context for coaching applications [6]. Systems simulating patient interviews explored complex architectures for managing real-time control loops to create more natural interactions, although sometimes lacking robustness or sufficient conversational depth for diverse scenarios [7], [8].

More recently, the emergence of powerful Large Language Models (LLMs) like GPT-4 and Google's Gemini [**?**] has catalyzed the development of highly interactive AI agents. Systems like "Smart Prep" leverage ChatGPT to power interactive interview preparation, combining speech/text analysis with feedback [9]. Studies also explored using GPT-4 with Zero-Shot Learning (ZSL) to generate personalized interview preparation guides [10], indicating improved relevance over simpler methods. However, reliance on external APIs raises concerns about cost, latency, internet dependency, and the potential for generic feedback if prompts are not carefully engineered.

The proposed Real-Time AI Voice Interviewing Agent builds upon these advancements while specifically addressing the need for realistic, voice-based, real-time interaction combined with immediate, structured feedback. It differentiates itself by focusing on the conversational dynamics of a voice interview, leveraging the generative capabilities of modern LLMs (facilitated by platforms like Vapi [**?**]) to create adaptive and natural-sounding conversations. It integrates NLP content analysis with real-time speech characteristic analysis and provides schema-driven feedback via LLMs, addressing the gap for affordable, realistic, voice-centric practice platforms found in the literature.

III. Proposed System

The proposed "Real-Time AI Voice Interviewing Agent" is a web-based platform designed to provide job seekers with a realistic, interactive, and personalized mock interview experience.

A. Overview

The system operates through a voice-first interface where users interact naturally with an AI agent. This agent utilizes sophisticated Speech-to-Text (STT), Large Language Models (LLMs like Google Gemini [**?**] via Vapi integration [**?**]), and Text-to-Speech (TTS) technologies for dynamic conversation. Unlike static question banks, the AI employs adaptive workflows, modifying questions based on user responses, target role, experience level, and interview focus (Technical/Behavioral), ensuring relevance and a challenging practice session appropriate for the user's goals.

A critical component is instant, comprehensive feedback. Post-interview, the AI analyzes the entire interaction transcript, evaluating multiple dimensions:

- **Content Analysis:** Evaluating the relevance, depth, structure, and clarity of answers using LLM semantic understanding.
- **Verbal Delivery Analysis:** Assessing aspects like speaking pace, perceived fluency (identifying potential overuse of filler words implicitly), and clarity, based on data captured by the voice platform.
- **Structured Feedback:** Presenting the evaluation in an organized format (scores, textual summaries) highlighting strengths, identifying specific areas for improvement, and offering concrete suggestions, guided by predefined JSON schemas.

B. System Architecture

The system employs a modern tech stack: Next.js [**?**] (React) frontend with Tailwind CSS, Firebase [**?**] backend (Auth, Firestore), and the Vapi SDK [**?**] for real-time voice orchestration. Fig. 1 illustrates the component interactions and data flow:

1) **Frontend (Next.js):** User interface for configuration, interaction, and viewing results. Built with React and styled with Tailwind CSS.
2) **Voice Interaction Core (Vapi SDK):** Manages bidirectional audio stream, STT, LLM calls (via Vapi's configured assistant), and TTS synthesis, providing real-time conversational capabilities.
3) **Backend & Data (Firebase):** Firebase Authentication handles secure user management. Firestore Database stores persistent data: user profiles, interview configurations (including generated questions), and structured feedback results.
4) **Server-Side Logic (Next.js API Routes/Server Actions):** Securely executes logic like generating interview questions and generating final feedback using Google Gemini [**?**] via the Vercel AI SDK, interacting with Firebase Admin SDK.

IV. Implementation Details

The system comprises interconnected modules, leveraging cloud services and modern web technologies.

A. Module 1: User Authentication

Firebase Authentication (Email/Password) manages user identity. A secure session management strategy employs HTTP-only cookies: client-side ID tokens obtained upon login are sent to a backend endpoint (Next.js Server Action or API

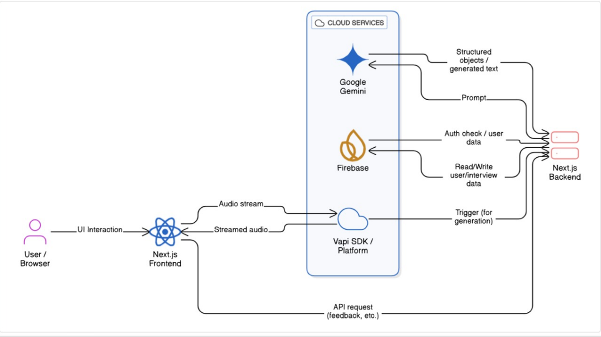


Fig. 1. System Architecture Diagram illustrating the data flow between the Next.js Frontend, Vapi SDK, Firebase, and Google Gemini.

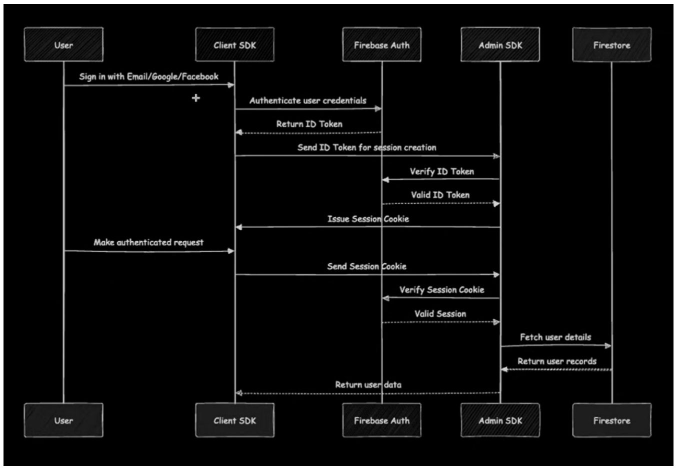


Fig. 2. Authentication Flow using Firebase Auth, Client SDK, Admin SDK, and Session Cookies.

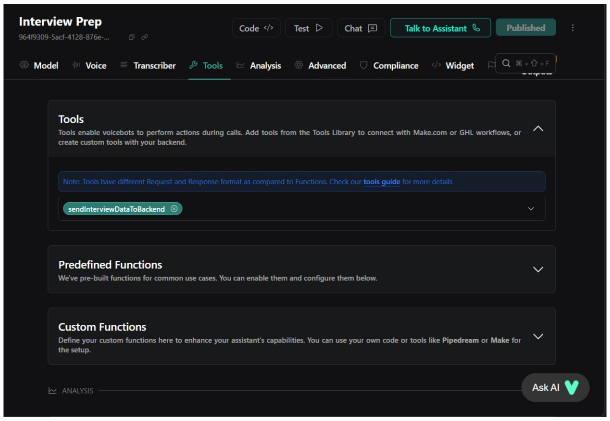
Route). This endpoint verifies the token using the Firebase Admin SDK and issues a session cookie (Fig. 2). Subsequent server-side requests are authenticated using this cookie, enhancing security. User profile details are persisted in Firestore.

B. Module 2: Interview Configuration and Generation

Users define parameters (role, level, type, question count) via a Vapi voice interaction configured through Vapi's dashboard (System Prompt + API Request Tool, Fig. 3). The Vapi tool (Fig. 4) collects these parameters from the conversation and makes a POST request to the '/api/vapi/generate' backend endpoint. This endpoint constructs a prompt for Google Gemini (Vercel AI SDK), requesting tailored questions in a specific JSON array format. The parsed question list and configuration parameters are saved to Firestore's 'interviews' collection.

C. Module 3: Real-Time Voice Interview Interaction

This core module utilizes the Vapi SDK ('@vapi-ai/web') within a Next.js Client Component ('Agent.tsx'). The component fetches generated questions from Firestore upon loading. The 'vapi.start()' function initiates the call, passing a predefined Assistant configuration object (specifying AI persona, TTS voice model, conversational LLM model) and injecting



Tools

Tools enable voicebots to perform actions during calls. Add tools from the Tools Library to connect with Make.com or GHL workflows, or create custom tools with your backend.

Note: Tools have different Request and Response format as compared to Functions. Check our tools guide for more details

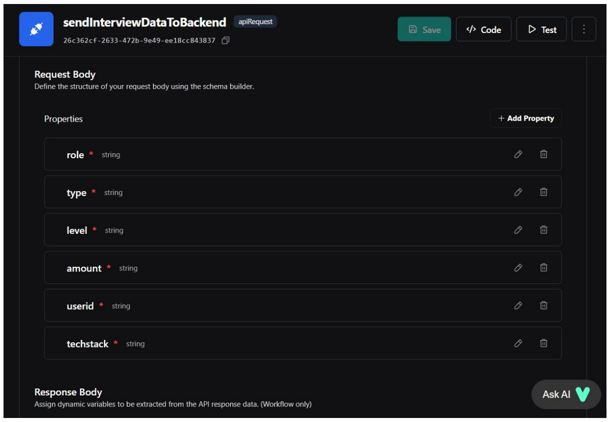
Predefined Functions

We've pre-built functions for common use cases. You can enable them and configure them below.

Custom Functions

Define your custom functions here to enhance your assistant's capabilities. You can use your own code or tools like **Pipedream** or **Make** for the setup.

Fig. 3. Vapi Assistant Configuration showing the attached API Request Tool ('sendInterviewDataToBackend').


sendInterviewDataToBackend
apiRequest
26c362cf-2633-472b-9e49-ee18cc843837
Save
Code
Test
+ Add Property
Ask AI

Request Body

Define the structure of your request body using the schema builder.

Properties

role * string

type * string

level * string

amount * string

userid * string

techstack * string

Response Body

Assign dynamic variables to be extracted from the API response data. (Workflow only)

Fig. 4. Vapi API Request Tool parameters configured to match backend expectations (e.g., role, level, textStack).

the fetched 'questions' as a context variable. Vapi manages the real-time STT-LLM-TTS communication loop. React's 'useEffect' hook is crucial for managing Vapi event listeners ('call-start', 'call-end', 'message', 'speech-start/end', 'error'), updating component state ('useState' for call status, transcript, speaking indicators). A cleanup function within 'useEffect' removes listeners ('vapi.off') upon component unmount to prevent memory leaks. Final transcript segments are collected into the 'messages' state array. The call is terminated via user action ('vapi.stop()') or natural conclusion.

D. Module 4: Feedback Generation and Presentation

Upon the 'call-end' event for an interview session, the frontend triggers a server action ('createFeedback'), passing the complete 'messages' transcript array. The backend formats this transcript into a coherent string. It then invokes Google Gemini's 'generateObject' function (via Vercel AI SDK), providing a detailed evaluation prompt and a predefined Zod schema ('feedbackSchema') which enforces a structured JSON output. This schema mandates fields like 'totalScore', 'strengths' (array), 'areasForImprovement' (array), 'finalAssessment' (string), and 'categoryScores' (object mapping categories to scores and comments). The validated, structured feedback object is saved to Firestore's 'feedback' collection, linked to the interview. The frontend then redirects the user to the feedback page, which fetches and renders this data, ensuring consistent and actionable insights are presented.

V. Results and Discussion

The implemented system, following the reference tutorial's structure, successfully demonstrated the intended functionalities and core objectives.

1) User Authentication and Session Management: Firebase Authentication enabled successful user registration and login. The secure session management using HTTP-only cookies functioned correctly, ensuring appropriate route protection and persistent user sessions across requests (See Fig. 5 for UI).

2) Interview Configuration and Generation: The Vapi voice workflow effectively captured user-defined interview parameters (as configured in Fig. 3 and 4). Backend logic utilizing Google Gemini generated contextually relevant questions based on these specifications, which were successfully persisted in Firestore (example structure in Fig. 8).

3) Real-Time Voice Interview Interaction: The core voice module performed effectively. Real-time interaction, orchestrated by Vapi's STT-LLM-TTS pipeline, was achieved with generally acceptable conversational latency. The AI agent adeptly followed the interview structure, posing the generated questions and engaging in conversational responses (See Fig. 7 for interface).

4) Feedback Generation and Presentation: Post-call processing successfully captured the transcript. Google Gemini's 'generateObject' function reliably produced structured feedback conforming to the predefined schema. This feedback was successfully stored in Firestore and presented clearly on the dedicated feedback page. The main dashboard (Fig. 6) correctly displayed available interviews for the user.

5) Overall Performance and Limitations: The Next.js application exhibited good responsiveness and usability. The voice interaction provided a natural and significantly more realistic practice environment compared to text-based alternatives. The immediacy and structured nature of the feedback were identified as highly valuable features. Observed limitations included occasional noticeable latency in the voice interaction loop, variability in STT accuracy depending on audio conditions and user accent, a strong dependency on the underlying LLM's quality and the effectiveness of prompt engineering for both conversation and feedback generation, and the absence of granular speech analytics (e.g., filler word counts, pace metrics) beyond the LLM's holistic assessment.

VI. Conclusion and Future Scope

A. Conclusion

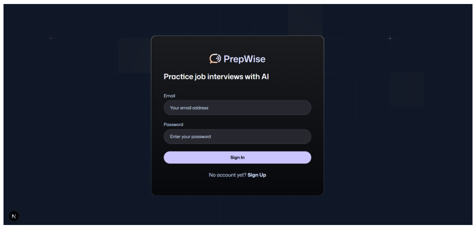
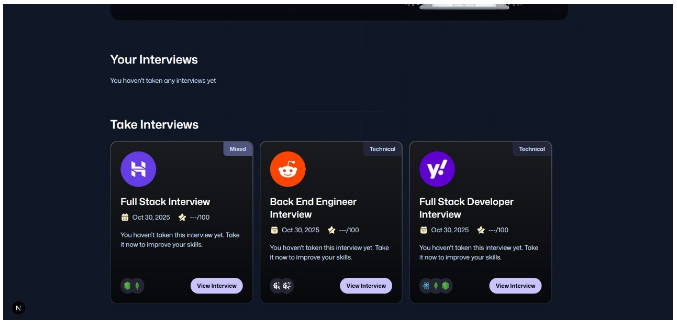


Fig. 5. User Authentication Screen.


Your Interviews

You haven't taken any interviews yet

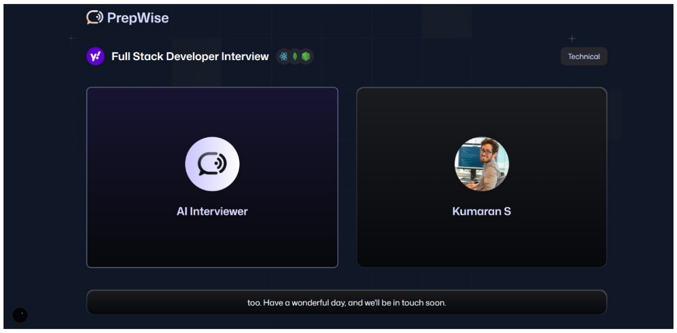
Take Interviews

Mixed — Full Stack Interview — Oct 30, 2025 — ---/100 — You haven't taken this interview yet. Take it now to improve your skills. — View Interview

Technical — Back End Engineer Interview — Oct 30, 2025 — ---/100 — You haven't taken this interview yet. Take it now to improve your skills. — View Interview

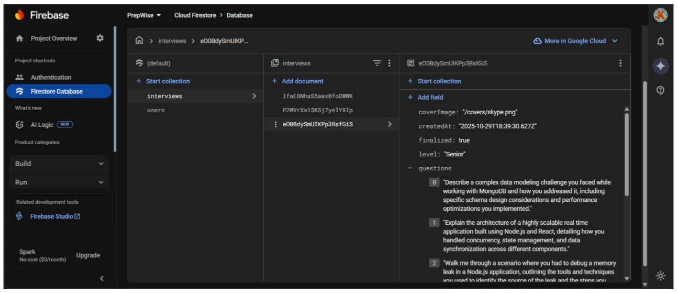
Technical — Full Stack Developer Interview — Oct 30, 2025 — ---/100 — You haven't taken this interview yet. Take it now to improve your skills. — View Interview

Fig. 6. Main User Dashboard after Login, showing available interview cards.



too. Have a wonderful day, and we'll be in touch soon.

Fig. 7. Application Interface during an AI Interview Session.



```
interviews > eO0BdySmUIKP...

(default)                interviews                   eO0BdySmUIKPp3BsfGiS
+ Start collection       + Add document               + Start collection
interviews               IfaEBNhaS5aex0foDWWK         + Add field
users                    P2MVrXa15KSj7yelYXIp         coverImage: "/covers/skype.png"
                         eO0BdySmUIKPp3BsfGiS         createdAt: "2025-10-29T18:39:30.627Z"
                                                      finalized: true
                                                      level: "Senior"
                                                      questions
                                                        0 "Describe a complex data modeling challenge you faced while
                                                          working with MongoDB and how you addressed it, including
                                                          specific schema design considerations and performance
                                                          optimizations you implemented."
                                                        1 "Explain the architecture of a highly scalable real time
                                                          application built using Node.js and React, detailing how you
                                                          handled concurrency, state management, and data
                                                          synchronization across different components."
                                                        2 "Walk me through a scenario where you had to debug a memory
                                                          leak in a Node.js application, outlining the tools and techniques
                                                          you used to identify the source of the leak and the steps you
```

Fig. 8. Example Data Structure for a Generated Interview in Firestore Database.

This project successfully demonstrated the design and implementation of a ”Real-Time AI Voice Interviewing Agent.” By integrating advanced AI technologies (STT, LLMs, TTS) within a modern web framework (Next.js, Firebase, Vapi), the system effectively addresses the limitations of conventional interview preparation methods. It moves beyond static, text-based interactions to provide a dynamic, voice-driven environment simulating real-world interviews, offering a significant improvement in practice realism.

The core achievement lies in the seamless orchestration of these technologies, enabling natural, real-time conversations with an AI agent capable of adaptive questioning based on user specifications. Furthermore, the integration of an AI-powered feedback mechanism, utilizing the structured output capabilities of LLMs like Gemini, represents a significant value proposition, delivering immediate and actionable insights. This system offers a promising solution to democratize interview coaching, making realistic practice and personalized feedback accessible and affordable, thereby holding significant potential to positively impact the career prospects of job seekers by enhancing their confidence and communication skills.

B. Future Scope

While the current system provides a robust foundation, numerous avenues exist for future enhancement to increase its value and effectiveness.

- **Enhanced Speech Analytics:** Integrate specialized libraries or APIs (e.g., AssemblyAI, Deepgram) to provide granular feedback on vocal delivery metrics, such as quantifiable filler word counts (‘um‘, ‘uh‘), precise pace variation analysis against benchmarks, and objective tone/sentiment analysis throughout the interview.
- **Multilingual Support:** Extend functionality to multiple languages. This involves configuring Vapi’s STT and TTS for different languages, ensuring the chosen LLM supports them, and potentially translating persona prompts and feedback schemas.
- **Video Integration:** Incorporate optional video stream processing to simulate video interviews. Computer vision techniques could analyze non-verbal cues like estimated eye contact duration, facial expression appropriateness, and general posture, adding another layer of feedback realism, albeit with increased complexity.
- **Deeper Specialization & RAG:** Develop more extensive prompt libraries tailored to niche job roles or industries. Implement Retrieval-Augmented Generation (RAG) by allowing users to upload job descriptions or company information, enabling the LLM to generate highly specific questions and evaluate answers against that context.
- **Adaptive Learning Paths:** Implement logic to dynamically adjust question difficulty or AI interviewing style based on detected user proficiency across multiple sessions. Introduce guided modules focusing on specific interview techniques (e.g., behavioral questions using the STAR method).
- **Integration with Job Platforms:** Explore API integrations with platforms like LinkedIn or job boards to allow users to directly import job descriptions, automatically configuring highly relevant mock interviews and potentially aligning feedback with desired competencies.

By pursuing these directions, the system can evolve into an even more comprehensive, personalized, and indispensable tool for career development and success in the modern job market.

References

[1] L. Chen, R. Zhao, C. W. Leong, B. Lehman, G. Feng, and M. E. Hoque, “Automated video interview judgment on a large-sized corpus collected online,” in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 2017, pp. 504–509.

[2] K. Priya, S. M. Mansoor Roomi, P. Shanmugavadivu, M. G. Sethuraman, and P. Kalaivani, “An Automated System for the Assesment of Interview Performance through Audio & Emotion Cues,” in *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*. IEEE, 2019, pp. 1049–1054.

[3] S. Yang, “Deep Automated Text Scoring Model Based on Memory Network,” in *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*. IEEE, 2020, pp. 480–484.

[4] L. Weaver-Lambert, *The AI Value Playbook: How to make AI work in the real world*. Packt Publishing, 2024.

[5] G. S. Harsh, Y. S. S. Vivek, M. P, S. K. Rout, S. R. Reddy, and B. K. Sethi, “Automated Interview Evaluation System Using RoBERTa Technology,” in *2024 1st International Conference on Cognitive, Green and Ubiquitous Computing (IC-CGU)*. IEEE, 2024, pp. 1–8.

[6] M. H. Su, C. H. Wu, K. Y. Huang, T. H. Yang, and T. C. Huang, “Dialog state tracking for interview coaching using two-level LSTM,” in *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2016, pp. 1–5.

[7] T. Hashimoto *et al.*, “Voice Dialog System for Simulated Patient Robot and Detection of Interviewer Nodding,” in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2021, pp. 01–06.

[8] Y. Shen *et al.*, “A Humanoid Robot Dialogue System Architecture Targeting Patient Interview Tasks,” in *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*. IEEE, 2024, pp. 1394–1401.

[9] R. M. Marvaniya, A. S. Acharya, D. M. Detroja, V. K. Dabhi, and H. B. Prajapati, “Smart Prep: AI Based Interactive Interview Preparation System,” in *2025 4th OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 5.0*. IEEE, 2025, pp. 1–6.

[10] T. Ramu and N. S. Naik, “Interview Preparation Guide Generation Leveraging GPT-4, ZSL and Hybrid Techniques,” in *2024 IEEE International Conference on Intelligent Signal Processing and Effective Communication Technologies (INSPECT)*. IEEE, 2024, pp. 1–6.