IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131061

LLM-Based Interview Assistance and Q&A Generator

Dr. Madhuri P. Borawake[1], Yashraj Nitin Bhandari[2], Shreepad Pravin Jadhav[3], Tushar Ramdas Markad[4], Apurva Santosh Kudale[5]

HOD, Associate Professor, PDEA's College of Engineering Manjari (BK), Pune, Maharashtra, India[1]

Department of Computer Engineering PDEA's College of Engineering Manjari (BK), Pune, Maharashtra, India[2-5]

Abstract: Preparing for technical interviews requires repeated practice, relevant questions, and constructive feedback. Learners often rely on static question banks or informal practice, which may not adapt to a selected role, topic, or the learner's response. This paper presents the design of an LLM-based interview assistance and question-and-answer (Q&A) generation system intended to support structured interview practice and, in a planned extension, quiz generation. The current prototype provides a web-based interaction flow using Flask, Google Gemini through LangChain, AssemblyAI speech-to-text, and Murf text-to-speech. A response can be captured, processed, and passed to a language model for contextual feedback. The proposed next phase extends the same foundation with Retrieval-Augmented Generation (RAG), which will retrieve relevant material from a curated knowledge base and use it to ground interview questions, model answers, explanations, and quiz items. The quiz-generation module and RAG integration are planned extensions and are not reported as completed implementation. Since formal user studies and quantitative benchmarking have not yet been conducted, this paper describes the system design, expected workflow, research motivation, and evaluation plan rather than claiming measured performance. The system is intended as a learning aid, not as a replacement for human interviewers or expert review.

Keywords: Large Language Models, interview assistance, Q&A generation, Retrieval-Augmented Generation, quiz generation, personalized learning, natural language processing

I. INTRODUCTION

Technical interviews assess a candidate's conceptual understanding, problem-solving approach, communication, and ability to apply knowledge. Students and early-career applicants may have difficulty finding consistent practice that reflects a target role and provides useful feedback. Static question banks can help with revision, but they may not respond to a learner's answer, explain a misconception, or adapt the next question to the learner's needs.

Large Language Models (LLMs) can generate fluent responses and support conversational interaction. However, unconstrained generation can also produce unsupported, outdated, repetitive, or incorrect material. A useful interview assistant therefore needs a structured interaction flow, explicit prompts, and mechanisms for grounding generated content. RAG offers one possible approach by retrieving relevant passages from a knowledge base and supplying them as context to the language model.

This project focuses on a combined learning workflow: interview practice and feedback in the current prototype, with a Q&A/quiz-generation module and RAG support planned for the next development phase. The intended system allows learners to practise role- or topic-specific questions and later test their understanding using generated assessment items. The research contribution at this stage is a design and implementation-oriented proposal that clearly separates working prototype functions from future modules.

A. Problem Statement

Learners need an accessible tool that combines interactive interview practice with reliable, topic-specific learning material. Existing static resources may not provide personalised follow-up or a single workflow for interview practice and self-assessment. At the same time, LLM outputs may contain inaccuracies unless the system incorporates suitable context and review. The problem is to design a web-based assistant that supports interview response feedback and can be extended with a RAG-grounded Q&A and quiz-generation pipeline.

B. Objectives

- Develop a web interface for interactive interview practice and response feedback.
- Use speech-to-text and text-to-speech services to support spoken interaction where available.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131061

- Design a future RAG pipeline to retrieve relevant knowledge before generating interview questions, answers, and explanations.
- Add a quiz-generation module that can create topic-aligned questions, options, answer keys, and explanations.
- Define an evaluation plan for relevance, factual correctness, usefulness, latency, and usability.

II. RELATED WORK

Research on LLM-based interview systems has explored question generation, technical and behavioural assessment, and automated feedback. Pasaribu et al. describe an LLM-based system for IT talent interviews, including behavioural and technical interview stages and structured assessment [1]. Their work provides a useful precedent for organizing interview content around competencies rather than generating unrelated questions.

Interview training systems have also examined the role of interaction and feedback. Hosseini et al. studied avatar-based feedback in job interview training and reported effects on action identification, anxiety, mood, and self-assessed performance [2]. Mongkoljaturong et al. introduced a digital-human interview training application combining an LLM with speech technologies [3]. These works suggest that practice quality depends not only on question generation but also on how learners interact with the system and receive feedback.

For automatic multiple-choice question generation, Hang et al. proposed MCQGen, which combines an LLM, prompt engineering, and RAG to support personalised question generation [4]. Awalurahman et al. reviewed transformer and LLM approaches for MCQ generation and identified prompt engineering, fine-tuning, RAG, and evaluation as important research directions [5]. So et al. proposed a pipeline with verification for generating educational MCQs, emphasising correctness, explanation quality, and diversity [6].

The literature motivates a system that treats interview practice and quiz generation as related but distinct tasks. Interview assistance requires contextual dialogue and response feedback, whereas quiz generation requires valid question stems, plausible distractors, a single or explicitly defined answer key, and explanations. RAG may support both tasks by supplying trusted, topic-relevant source material, but retrieval alone does not guarantee correctness; generated content still needs verification.

A. Research Gap

Prior work often concentrates on either interview simulation and assessment or educational question generation. For a student-facing prototype, there is an opportunity to connect these functions within one learning workflow: practise answering interview questions, review feedback, and assess the same topic through a quiz. The proposed extension also emphasises retrieval-grounded generation across both modules. The gap addressed here is therefore an integration and design gap, not a claim that interview assistants, quiz generators, or RAG are individually new.

B. Proposed Contribution

The proposed contribution is a modular web application design that brings together interview assistance, spoken interaction support, and a planned RAG-based Q&A/quiz pipeline. It explicitly distinguishes generated practice content from source-grounded material and defines quality checks for future evaluation.

III. SYSTEM OVERVIEW AND ARCHITECTURE

The system is organised into a presentation layer, interaction and orchestration layer, language-model layer, speech-service layer, and planned retrieval/assessment layer. The current prototype is a Flask web application. LangChain is used to orchestrate interaction with Google Gemini. AssemblyAI is used for speech-to-text, while Murf provides text-to-speech functionality. The RAG and quiz-generation components are proposed extensions and must be implemented and tested before they can be described as operational.

TABLE I. LAYERS AND RESPONSIBILITIES OF THE PROPOSED SYSTEM

Layer / module	**Responsibility**
Web interface (current)	Collect user input and present questions, responses, and feedback.
Application orchestration (current)	Flask routes and LangChain-based model interaction.
LLM response generation (current prototype)	Google Gemini generates conversational responses and feedback based on the prompt and supplied input.
Speech services (current prototype)	AssemblyAI converts speech to text; Murf converts text to speech.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131061

Knowledge retrieval (planned RAG)	Chunk approved documents, embed and index them, retrieve relevant passages for each query, and pass context to the LLM.
Q&A and quiz generator (planned)	Generate topic-aligned questions, answer keys, distractors, and explanations from user-selected topics and retrieved context.
Validation and evaluation (planned)	Check answer consistency, source support, duplication, difficulty, and learner usability.

A. Interview Assistance Workflow

1) The learner opens the web application and enters a topic, role, or interview prompt.
2) For spoken interaction, the audio input is transcribed by the speech-to-text service; text input can be processed directly.
3) The application sends the prompt and relevant conversation context to the LLM orchestration layer.
4) The LLM produces a response or feedback intended to help the learner improve clarity, correctness, and completeness.
5) If text-to-speech is enabled for the response, the output is passed to the speech service and presented to the learner.

B. Planned RAG Workflow for Both Modules

The planned RAG extension will begin with a curated collection of trusted learning resources, such as course notes, approved technical documentation, and topic-specific reference material. Documents will be cleaned, divided into manageable chunks, converted into vector embeddings, and stored in a searchable index. At query time, the retriever will find passages relevant to the learner's topic or answer. The retrieved context, the task instructions, and the learner's request will be passed to the LLM. For interview assistance, this can support grounded question creation and answer feedback. For quiz generation, it can provide evidence for question stems, answer options, answer keys, and explanations.

A post-generation validation stage is recommended. It should check whether each generated answer is supported by retrieved material, whether distractors are clearly incorrect, whether the question has an unambiguous answer, and whether near-duplicate items occur. If retrieved evidence is insufficient, the system should indicate uncertainty or request better source material instead of presenting a guess as a verified fact.

IV. METHODOLOGY

A. Current Prototype

The prototype uses Flask to provide the web application flow. LangChain connects the application to Google Gemini for language-model interactions. AssemblyAI and Murf support speech input and output, respectively. The application receives user content, prepares a prompt, invokes the model, and returns a response to the interface. The implementation is a first prototype; its present scope should not be interpreted as a completed production-ready interview assessment platform.

B. Planned Quiz Generation

The quiz module is planned to accept a topic or selected learning content and a requested number or type of questions. A generated item may include a question stem, answer options, the correct answer, and an explanation. The design should support configurable difficulty and question types. For multiple-choice questions, the generator must avoid multiple correct options, implausible distractors, duplicated questions, and explanations that conflict with the answer key. A human review step is recommended for high-stakes educational use.

C. Planned Retrieval-Augmented Generation

RAG combines retrieval from an external knowledge collection with language generation. The planned pipeline is:
(i) ingest and preprocess reference documents; (ii) chunk documents with metadata; (iii) create embeddings and build an index; (iv) retrieve top-ranked passages for a user query; (v) construct a prompt that includes retrieved evidence and task-specific constraints; (vi) generate the requested interview or quiz content; and (vii) validate the result and preserve source references where feasible. The implementation choice of embedding model, vector store, chunk size, and retrieval strategy remains a development decision to be confirmed experimentally.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131061

D. Prompt and Output Design

Prompts should specify the learner's topic, target role, difficulty, response format, and constraints. Interview feedback can be organised into strengths, missing concepts, correctness, clarity, and suggested improvement. Quiz output should be structured so the interface can display each question and its options consistently. Structured output reduces formatting errors but does not by itself establish factual accuracy.

E. Evaluation Plan

Because the project has not yet completed formal evaluation, no numerical performance claims are made. Future evaluation should use a test set of topics and reference documents, and compare outputs with and without retrieval. Human reviewers can rate relevance, correctness, clarity, answerability, distractor quality, and alignment with source material. Operational measures should include response latency, retrieval time, failure rate, and usability feedback. Results should report sample size, evaluation rubric, reviewer agreement where available, and limitations.

V. EXPECTED BENEFITS AND DISCUSSION

The integrated design is expected to make practice more convenient by bringing interview rehearsal and self-assessment into one interface. Spoken interaction may help learners practise communicating answers aloud, while text feedback can support reflection and revision. The planned RAG layer is expected to improve alignment with selected reference content and make the system easier to adapt to different technical subjects. These are expected benefits, not experimentally established outcomes.

Combining the modules creates design challenges. Interview feedback should be constructive and avoid presenting subjective model judgements as definitive hiring assessments. Quiz generation needs strict validation because an incorrect key can mislead learners. Retrieved sources may be incomplete, contradictory, or outdated. The system should therefore identify source limitations, avoid unnecessary collection of personal data, and encourage users to verify important technical information.

A. Limitations

The current work is a prototype and the quiz-generation and RAG components are planned rather than completed. The system has not yet been evaluated through a controlled user study or benchmark. Language-model responses can be inaccurate, speech recognition can mis-transcribe technical vocabulary, and speech synthesis can introduce latency. Retrieval quality will depend on the coverage and quality of the knowledge base. The system is not intended to make hiring decisions or certify a candidate's competence.

VI. FUTURE SCOPE

- Implement document ingestion, chunking, embeddings, vector search, and source-aware retrieval for both interview and quiz workflows.
- Develop quiz generation with configurable question count, topic, difficulty, question format, answer key, and explanation.
- Add automated checks for answer-key consistency, duplicate questions, source support, and distractor validity, followed by human review.
- Store learner progress securely and optionally use previous practice topics to recommend revision areas.
- Evaluate the application with students using predefined rubrics and report relevance, correctness, usability, latency, and failure cases.
- Explore multilingual content and role-specific knowledge bases while assessing the impact of domain coverage and retrieval quality.

VII. CONCLUSION

This paper presents an IEEE-style design description of an LLM-based interview assistance and Q&A generation system. The current prototype combines a Flask interface, Google Gemini through LangChain, and speech services for spoken interaction and model-generated feedback. The proposed next phase adds RAG to ground both interview assistance and quiz generation in curated reference material. This paper deliberately distinguishes implemented prototype functions from planned modules and does not claim unmeasured accuracy or usability results.

The main direction is to connect practice, feedback, and assessment in one modular learning tool while improving factual grounding and output quality. Completing the RAG and quiz components, followed by systematic human

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131061

evaluation, will be necessary to determine whether the proposed design delivers reliable educational value.

REFERENCES

[1] D. K. H. Pasaribu, A. Dewandaru, and G. A. P. Saptawati, "Development of LLM-Based System for IT Talent Interview," in Proc. 2024 IEEE International Conference on Data and Software Engineering (ICoDSE), 2024. (Bibliographic details should be confirmed from the source PDF before submission.)

[2] S. Hosseini, J. Quan, X. Deng, Y. Miyake, and T. Nozawa, "Avatar-Based Feedback in Job Interview Training Impacts Action Identities and Anxiety," IEEE Transactions on Affective Computing, vol. 15, no. 3, pp. 1608–1620, 2024, doi: 10.1109/TAFFC.2024.3363835.

[3] K. Mongkoljaturong, M. Manitsakulwong, and S. Moodleah, "AI-Powered MetaHuman Interviewer: Serious Game for Student Job Interview Skills," IEEE Access, vol. 13, 2025, doi: 10.1109/ACCESS.2025.3640017.

[4] C. N. Hang, C. W. Tan, and P.-D. Yu, "MCQGen: A Large Language Model-Driven MCQ Generator for Personalized Learning," IEEE Access, vol. 12, 2024, doi: 10.1109/ACCESS.2024.3420709.

[5] H. W. Awalurahman, R. F. Aji, and I. Budi, "Transformer and Large Language Models for Automatic Multiple-Choice Question Generation: A Systematic Literature Review," IEEE Access, vol. 13, 2025, doi: 10.1109/ACCESS.2025.3590423.

[6] C. C. So, C. H. Lee, D. Yang, A. W. K. Loh, and C. K. Sze, "Automatic Generation of High-Quality MCQs With LLMs for Artificial Intelligence Education," IEEE Access, vol. 13, 2025, doi: 10.1109/ACCESS.2025.3624820.

[7] C. Kim et al., "Fairness-Aware Multimodal Learning in Automatic Video Interview Assessment," IEEE Access, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3325891.

[8] F. Nagasawa, S. Okada, T. Ishihara, and K. Nitta, "Adaptive Interview Strategy Based on Interviewees' Speaking Willingness Recognition for Interview Robots," IEEE Transactions on Affective Computing, vol. 15, no. 3, 2024, doi: 10.1109/TAFFC.2023.3309640.

[9] A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems, vol. 30, 2017.

[10] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Advances in Neural Information Processing Systems, vol. 33, 2020.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License