International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI-Powered Virtual Interviewer for Career Preparation: A Resume-Aware Web Platform for Technical and Non-Technical Roles

Siddhi Shriram Lokhande[1], Darshana Prasad Gontla[2], Jiya Salim Patel[3], Prof. V. M. Joshi[4]

[1]Student, Department of Computer Engineering, Vishwabharati Academy's College of Engineering, Ahmednagar
[2]Student, Department of Computer Engineering, Vishwabharati Academy's College of Engineering, Ahmednagar
[3]Student, Department of Computer Engineering, Vishwabharati Academy's College of Engineering, Ahmednagar
[4]Guide and Head, Department of Computer Engineering, Vishwabharati Academy's College of Engineering, Ahmednagar

---

Abstract - Interview performance strongly affects the employability of students and fresh graduates, yet realistic and personalised practice is rarely available to them. Existing artificial intelligence interview tools mostly target information-technology roles, depend on a single opaque language-model score, or need heavy hardware such as game engines and 3D avatars. This paper presents the design and implementation of an AI-powered virtual interviewer: an accessible web platform for technical and non-technical interview preparation. Building on a recent MetaHuman-based interviewer study, the proposed system adds resume-aware personalised questions, a role-specific weighted rubric in which generative AI is never the sole scoring authority, interview etiquette training, progress tracking, and fairness and privacy controls, while deliberately excluding facial expressions from scoring. The platform follows a layered MongoDB, Express.js, React.js and Node.js architecture with Firebase Google authentication verified on the server. The platform offers role selection, a question bank covering technical, banking, business and other non-technical domains, resume analysis, LLM-assisted follow-up questions and explanatory feedback, an etiquette learning coach, and a progress dashboard, with optional speech interaction.

Key Words: virtual interviewer, large language model, resume analysis, rubric-based evaluation, interview etiquette, MERN stack.

1. INTRODUCTION

A job interview is usually the final and often decisive step of a recruitment process. Candidates are judged not only on what they know but also on how clearly they structure answers, how professionally they communicate, and how well they understand the role. For students and fresh graduates this is a new and stressful experience, and realistic practice is hard to find: college placement cells have limited time, human mock interviewers are scarce, and most online material is a static list of questions.

Recent progress in Large Language Models (LLMs) and speech technology makes it possible to build virtual interviewers that ask relevant questions, follow up on answers and give written feedback at any time and at low cost. A recent study in IEEE Access [1] built the Interactive Interview Assistant Application (IIAA), a serious game in which a MetaHuman digital interviewer, an LLM, and speech-to-text (STT), text-to-speech (TTS) and speech-to-animation (STA) components simulate interviews for three information-technology careers. The authors report high user satisfaction but also note response latency, weak recognition of mixed-language speech, scoring that was not validated by human experts, and a small single-institution sample.

Two practical gaps follow from this. First, candidates for non-technical roles such as banking, Human Resources (HR), sales, teaching and customer support need role-specific behavioural practice, communication guidance and feedback, which technology-focused tools do not provide. Second, many automated tools return opaque scores and some rely on appearance-based cues, which raises fairness and privacy concerns. This paper therefore proposes an accessible, browser-based Artificial Intelligence (AI) interviewer that is broader in the roles it supports and more responsible in how it scores.

The contributions of this work are: (i) a critical analysis of the base study [1] and of related work, leading to an explicit research gap; (ii) a software requirement specification and a layered architecture for a text-first web platform that needs no game engine or GPU; (iii) a role-specific, weighted rubric model in which generative AI is never the sole scoring authority; (iv) an implementation on the MongoDB, Express.js, React.js and Node.js (MERN) stack with server-verified Google authentication, a role and question bank, resume analysis, an interview session engine, rubric-based evaluation, an etiquette coach and a progress dashboard; and (v) an evaluation approach covering scoring consistency, fairness, usability and security.

Section 2 reviews related work and the base study. Section 3 describes the proposed system. Section 4 presents the evaluation model and interview methodology. Section 5 explains the implementation. Section 6 presents the evaluation approach, discussion and limitations, and Section 7 concludes with future work.

2. LITERATURE REVIEW

2.1 AI-based interview preparation

Research on technology-supported interview training has grown quickly. Khallelullah et al. [2] analyse interviews that combine video analysis with audio-to-text conversion to assess verbal and non-verbal communication. Jadhav et al. [3] describe a mock-interview simulator with an AI interviewer and posture detection. Qin et al. [4] propose DuerQues, which uses a neural generative model and skill-entity recognition to recommend skill-oriented interview questions. Lee et al. [5] combine virtual reality with a generative-AI chatbot for multilingual interview practice by Hong Kong graduates, and Mishra et al. [6] describe a GPT-4 based virtual mock-interview platform with context-aware questions and

personalised feedback. EZInterviewer [7] generates questions and dialogue from job descriptions and candidate resumes, which directly supports resume-aware personalisation. Table 1 summarises these works and their limitations.

Table -1: Summary of related work

Work	Technique	Limitation
[1]	MetaHuman avatar, ChatGPT 3.5, wav2vec2 STT	Three IT roles; no resume use; unvalidated LLM score; heavy hardware
[7]	Dialogue generation from job description and resume	Text only; no feedback, etiquette or tracking
[4]	Neural generation, skill-entity recognition	Question generation only
[6]	GPT-4 mock interview	Limited detail on rubric and fairness
[3]	AI interviewer with pose detection	Appearance cues; fairness not discussed
[5]	VR with generative-AI chatbot	Needs VR set-up; regional focus
[2]	Video and audio-to-text analysis	Reliability of non-verbal metrics

2.2 Analysis of the base study

The base study [1] was implemented in Unreal Engine 5.2 for Thai-language users. Three Thai-capable STT models were compared on five phrases processed ten times each: Whisper [8] was fastest but inaccurate, a fine-tuned Thonburian Whisper was accurate but slow, and a wav2vec2 model gave the best balance (95.58% accuracy) and was selected. LLaMA 2 was rejected because it destabilised the engine and needed about 60 GB of storage, so ChatGPT 3.5 Turbo 16k was used with a prompt that defines an HR persona, behavioural rules and a scoring task (a 1–10 score plus a short critique for each answer). Questions are drawn at random from a career-specific bank in two modes: a basic session of about 10 minutes and a comprehensive session of about 30 minutes in which each answer has a 50% chance of triggering an in-depth follow-up. Speech uses Google and Azure Thai neural voices, and facial animation is produced by the MetaHuman SDK and layered with idle animations. A minimum of 75 characters per TTS request gave the best balance between start-up delay and pauses.

Ten final-year students rated the system on a 10-point scale adapted from the System Usability Scale (SUS) [9]. The overall experience scored 8.2 and usefulness of feedback 8.9, while accuracy of speech detection scored lowest at 7.0. The combined TTS and STA stage was the bottleneck at 21.09 characters per second, compared with 50.38 for STT and 45.64 for ChatGPT.

From the viewpoint of this project, the study has several limitations that guide our design: it covers only three IT career paths; questions come from a fixed bank and are not personalised to the candidate's resume; scoring is a single LLM-generated value without a published rubric, weights or consistency tests; there is no etiquette teaching or cross-session progress view; fairness is not discussed beyond data privacy; and an Unreal Engine avatar needs high-end hardware, which limits accessibility.

2.3 Supporting technologies

Serious games are designed for purposes beyond entertainment [10], and systematic reviews stress rigorous evaluation such as pre-test and post-test designs [11]. Virtual humans with emotions and social cues can make training more engaging but are costly to build [12]. LLMs are widely used for personalised feedback and content generation, although ethics, privacy and over-reliance remain concerns [13]. Speech recognition offers immediate feedback, but accuracy drops with accents, noise and overlapping speech [8, 14]; for this reason we keep voice input optional.

2.4 Research gap

Individual technologies are well studied, but the survey shows that (G1) most systems, including the base study, target a few IT roles while non-technical roles receive little attention; (G2) resume-aware question generation [7] is rarely combined with feedback, etiquette teaching and progress tracking in one platform; (G3) automatic scores are usually produced by an LLM without published rubrics, consistency testing or validation; (G4) non-verbal analysis is sometimes used as a performance signal although facial expressions are unreliable proxies of competence and are affected by culture, disability and camera conditions; and (G5) privacy, accessibility and explainability are seldom treated as design requirements. Table 2 contrasts the base study with the proposed system. We do not claim that no non-technical interview application exists; the claim is limited to the integration of these capabilities in one accessible platform, whose evaluation is described in Sec. 6.

Table -2: Base study versus proposed system

Feature	Base study [1]	Proposed system
Domains	UI/UX, software engineering, game development	Technical and non-technical roles
Interviewer	3D MetaHuman avatar	Text-first AI interviewer; optional voice
Platform	Unreal Engine desktop application	Web application (MERN)
Resume use	Not used	Extraction, user review, personalised questions
Scoring	LLM score, 1–10	Weighted rubric, rule checks and LLM explanation
Etiquette	Not included	Learning coach with STAR method, quizzes
Progress	Download of one session	Dashboard across sessions
Appearance cues	Not used for scoring	Optional, never part of the score
Fairness, privacy	Non-persistent data model	Fair mode, consent, deletion, accessibility

3. PROPOSED SYSTEM

3.1 Objectives and scope

The platform lets a candidate practise interviews for a chosen domain, role and experience level, receive structured feedback, learn interview etiquette and track progress. Its objectives are to support technical and non-technical roles, analyse an uploaded resume, generate personalised questions, conduct text and optional voice interviews, evaluate answers against role-specific rubrics, teach etiquette, and protect privacy. The system is a preparation aid only. It does not make hiring decisions, does not use facial expressions, eye contact or appearance to score competence, and does not provide a 3D avatar.

3.2 Requirements

The software requirement specification defines eighteen functional requirements (FR-1 to FR-18) and non-functional requirements. Functional requirements cover Google sign-in and server-side token verification, selection of domain, role, level, difficulty and mode, storage and retrieval of roles and questions, interview sessions with stored answers, session summaries, administrator management of content, resume

upload and correction, personalised questions, rubric-based evaluation, etiquette lessons, a progress dashboard, optional speech and presentation observations, a fair-interview mode, feedback flagging and data deletion. Non-functional requirements include response of about one second for non-AI calls, at most three steps from sign-in to starting an interview, keyboard navigation and a text-only option, secrets kept in environment variables, restricted cross-origin access, input and upload validation, explicit consent for microphone or camera, and visible rubrics and feedback reasons so that scores rest only on job-relevant criteria.

3.3 System architecture

The system uses a layered client–server architecture (Fig. 1). Each layer has one responsibility, so the AI provider, resume parser or speech library can be replaced without changing the rest of the system. The presentation layer is a React single-page application; an authentication layer uses Firebase Google sign-in on the client and ID-token verification on the server; the Express application layer exposes a REST (Representational State Transfer) API with authentication and validation middleware, a session manager and the scoring workflow; an AI services layer provides the language model, resume parser and optional speech and vision; MongoDB holds all data; and deployment targets Render and MongoDB Atlas.

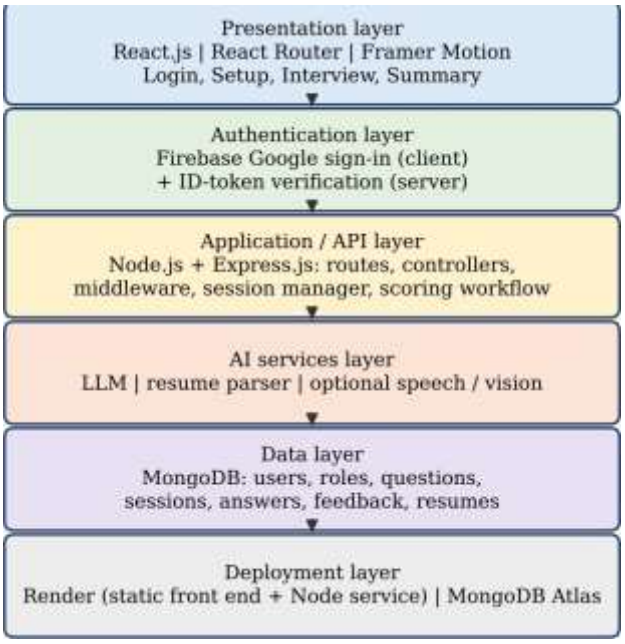


Fig -1: Layered system architecture.

3.4 Workflow and modules

After signing in, the candidate selects domain, role, level, difficulty and mode and may upload a resume, which is extracted and then reviewed and corrected by the user. An interview plan is created, the AI interviewer asks questions, each answer is evaluated against the role rubric, follow-up questions are asked when needed, and a feedback report with etiquette tips and suggested practice is produced and saved with permission (Fig. 2). The functionality is divided into nine modules: authentication and profile (M1), role and question bank (M2), interview session engine (M3), resume analysis (M4), evaluation and feedback (M5), etiquette coach (M6), progress dashboard (M7), fair mode and privacy (M8), and optional speech and presentation (M9).

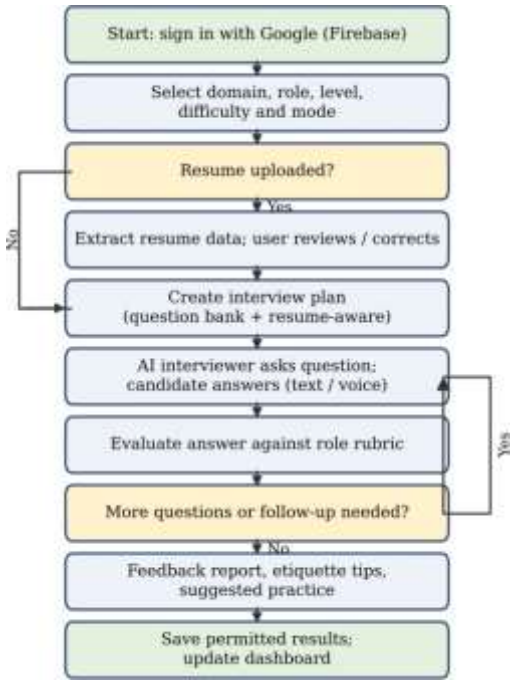


Fig -2: Proposed system workflow.

4. EVALUATION MODEL AND INTERVIEW METHODOLOGY

The methods in this section are the core of the platform. Their purpose is to remove the main weakness of the base study, in which one LLM output was the whole score.

4.1 Role-specific rubric and scoring

Each role family has a rubric of weighted criteria. Table 3 shows an illustrative rubric for a banking or customer-service interview. The weights are prototype values to be adjusted and validated for each role; they are not industry standards. For technical roles the criteria are technical correctness and conceptual understanding, problem-solving approach, explanation of the solution and trade-offs, and use of examples. Facial appearance and expressions are not part of any rubric.

Table -3: Illustrative rubric for non-technical roles

Criterion	Weight	What it evaluates
Answer relevance	30%	Whether the response addresses the question
Role knowledge	25%	Understanding of relevant banking or customer-service concepts
Communication clarity	20%	Whether the answer is understandable and organised
Reasoning and examples	15%	Quality of explanation and relevant examples
Answer structure	10%	Clear beginning, main point and conclusion

For an answer a with criterion scores s_i in [0, 10] and weights w_i whose sum is 1, the answer score and the session score over m answers are

$$S(a) = \Sigma\, w_i s_i, \quad \Sigma\, w_i = 1 \qquad (1)$$

$$S_{session} = (1/m)\, \Sigma\, S(a_j), \; j = 1,\ldots,m \qquad (2)$$

Each s_i combines rule-based checks (for example expected keywords or concepts from the question bank, answer length within a sensible range, and the presence of Situation, Task, Action and Result (STAR) components for behavioural

questions) with AI-supported semantic analysis constrained to the rubric. The AI component must return a structured object with a score and a short justification per criterion; invalid output is rejected and retried (Fig. 3). Feedback follows the order of what was good, what is missing, how to improve, and a sample improved outline.

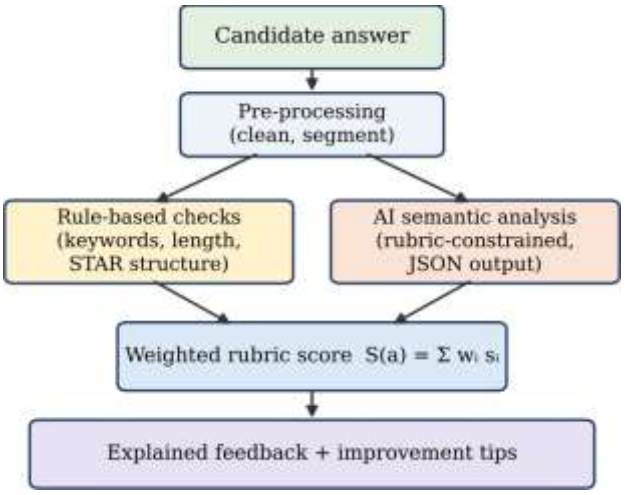


Fig -3: Proposed answer-evaluation pipeline.

4.2 Content-based follow-up logic

The base study decides on a follow-up with a fixed 50% probability [1]. We keep its basic and comprehensive session modes but make the decision depend on the answer: a follow-up is asked when relevance is low, the answer is too short, or it contains a vague claim, up to a maximum depth (Algorithm 1).

Algorithm 1: Follow-up decision logic

```
for each planned question q:
  ask(q); a = receive_answer()
  s = evaluate(a, rubric(q)); depth = 0
  while depth < MAX_DEPTH and
        (s.relevance < T_LOW or
         a.is_too_short or a.has_vague_claim):
    f = generate_followup(q, a)
    ask(f); a = receive_answer()
    s = evaluate(a, rubric(q)); depth += 1
  store(q, a, s)
generate_report(session)
```

4.3 Resume analysis and personalised questions

The user uploads a PDF or DOCX file whose type and size are validated. Text is extracted, and rules together with LLM-assisted structuring return education, skills, projects, internships and experience as structured data. The user reviews and corrects this data before use and may delete the resume. Skills are compared with the role requirements to suggest skill gaps. The module does not infer sensitive characteristics and does not penalise information that the role does not require. The interview plan is a hybrid: a fixed share of questions comes from the curated role bank to keep difficulty consistent, and the rest is generated from resume items and the role description, then checked for length, duplication and relevance.

4.4 Etiquette, presentation feedback and fairness checks

Etiquette lessons are reviewed static content with scenario questions and short quizzes covering preparation, the STAR method, closing and follow-up messages. Presentation feedback, if enabled with explicit permission, is limited to observable practice conditions such as the face being visible and lighting; it is shown separately, never changes the job-skill score, and a camera-off option is always available. Scoring consistency will be tested by submitting equivalent answers several times and by checking whether irrelevant wording, spelling style or accent-related transcription changes alter scores. Agreement with a small human review panel will be measured with Cohen's kappa or a similar measure, as the base study itself proposes as future work [1].

5. IMPLEMENTATION

5.1 Technology stack

The MERN stack with Firebase authentication was chosen because one language (JavaScript) is used on both sides, the ecosystem is mature, free tiers exist for development, and document storage suits role, question, resume and feedback data that may evolve. The front end uses React [15] with React Router, Framer Motion and Axios. The back end uses Node.js with Express [16], Mongoose and firebase-admin, and the database is MongoDB [17]. Google sign-in is provided by Firebase Authentication [18], which avoids building password management. Razorpay supports optional premium features, kept separate from scoring, and the application is deployed on Render.

5.2 Front end

The React application contains pages for Login, Setup, Interview, Summary, Resume, Lessons and Dashboard. Login offers "Continue with Google". Setup loads domains and roles from the API and lets the user choose level, difficulty and mode. Interview shows the current question in an interviewer bubble with a text area, optional microphone input, a progress indicator and an end button. Summary presents scores and feedback for the session, Resume lets the user upload and correct extracted data, Lessons delivers etiquette content, and Dashboard shows previous sessions and trends. An Axios interceptor attaches the current user's Firebase ID token to every request, and the interface follows the accessibility requirements (labels, keyboard focus and contrast).

5.3 Back end and security

The back end follows a routes–controllers–models structure. A middleware function reads the bearer token and calls the Firebase Admin verification method on every protected request; client claims are never trusted, and a user profile is created on first sign-in. Sessions are always looked up by both identifier and the authenticated user, so one user cannot read or write another's session. Other measures already applied are environment variables for all secrets, a restricted cross-origin origin, a JSON body size limit, input checks, a 5000-character limit on answers, and a role check on administrator routes.

5.4 Database and API

MongoDB stores the platform data, and Mongoose schemas enforce its structure (Fig. 4 and Table 4). A seed script inserts roles and questions for technical roles (Java, Python, web development, data analyst), banking and finance (probationary officer, clerk, relationship manager), business and management (HR, marketing, sales) and other non-technical roles (customer support, administration, teaching). Question types are technical, behavioural, knowledge and situational.

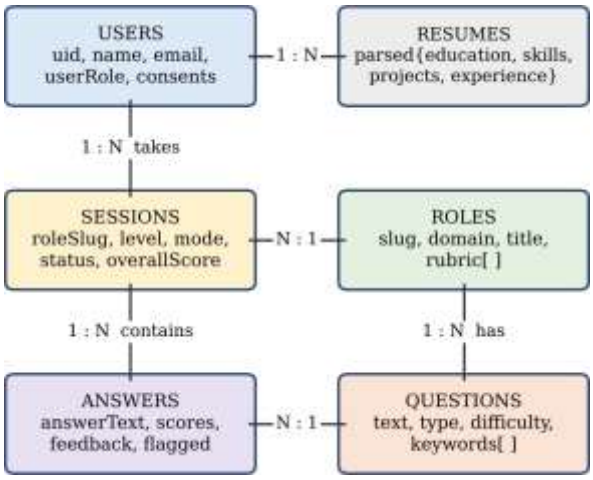


Fig -4: Entity–relationship diagram.

Table -4: MongoDB collections

Collection	Main fields
users	uid, name, email, userRole, consents
roles	slug, title, domain, rubric[{criterion, weight}]
questions	roleSlug, text, type, difficulty, keywords
sessions	userId, roleSlug, level, mode, status, overallScore
answers	sessionId, questionId, answerText, criterionScores, feedback
resumes	userId, parsed data, reviewedByUser
lessons, progress	lesson content, quiz results, per-topic statistics

The REST API is exposed under /api and every endpoint except the health check requires a valid ID token. Endpoints verify the token and return the profile, list roles, return questions for a role and level, and manage sessions: creating a session draws random active questions for the chosen role and difficulty, answers are saved one by one, and ending a session returns the summary. Further endpoints handle resume upload and parsing, rubric evaluation, progress data, lessons, data deletion and optional payments. Administrators can create, edit and deactivate roles and questions.

6. EVALUATION, DISCUSSION AND LIMITATIONS

6.1 Evaluation approach

The platform is evaluated along the dimensions in Table 5. Functional testing covers sign-in, role selection, resume upload, interviews, feedback and the dashboard. Question relevance is judged by reviewers against the selected role and resume. Scoring consistency is tested by submitting equivalent answers several times, and agreement with a human review panel is measured with Cohen's kappa. Fairness review confirms that no facial or protected characteristic is used and examines the effect of irrelevant wording or accent. Usability is measured with a SUS-style questionnaire [9] completed by students preparing for technical and non-technical interviews.

Table -5: Evaluation dimensions

Dimension	Method
Functional	Test cases for each module and user flow
Resume extraction	Supported file types, missing fields, malformed files, user correction
Question relevance	Reviewer judgement against role and resume
Scoring consistency	Repeated equivalent answers; agreement with human panel
Fairness	Check for non-job-relevant signals; wording and accent effects
Usability	SUS-style questionnaire
Performance, security	Response time, upload limits, access control, API security

6.2 Discussion

By design the platform removes the avatar-rendering stage that the base study identified as its main latency bottleneck [1], and it runs in an ordinary browser, which lowers the hardware barrier for students. The rubric-based model addresses the gaps in scoring transparency and consistency, resume-aware questions make practice relevant to each candidate, and the explicit exclusion of facial expressions from scoring addresses a fairness concern that earlier systems leave open. Etiquette lessons and the progress dashboard extend the platform from a question-and-answer tool to a learning aid.

6.3 Limitations

The question bank and rubrics were written by the project team from public role descriptions and reviewed by the guide, so they are prototype material and the rubric weights are not industry standards. Generated questions and feedback may be inaccurate and are shown with clear limitations. Resume extraction can fail for scanned or unusual documents, and speech recognition quality varies with accent, noise and microphone [14]. The platform depends on external services such as Firebase, database hosting and AI providers, with their cost, rate limits and outages. Finally, practice scores do not guarantee the outcome of a real interview, and our novelty claim concerns the integration of capabilities and remains subject to further literature review.

7. CONCLUSION AND FUTURE WORK

This paper presented an AI-powered virtual interviewer for technical and non-technical career preparation. Starting from a critical analysis of a MetaHuman-based interviewer study, it described a text-first web platform with resume-aware personalisation, a role-specific rubric model in which generative AI is not the sole scoring authority, etiquette training, a progress dashboard, and built-in fairness and privacy safeguards, implemented on the MERN stack with Firebase authentication.

Future work includes validating the automated scoring against a human expert panel with inter-rater agreement metrics, and a pre-test and post-test study with a larger multi-institution sample and a control group to measure real learning gain, as the base study also suggests [1]. Further directions are multilingual interviews with better handling of mixed-language speech, a lightweight avatar where hardware permits, hybrid processing with local speech components to reduce dependence on online APIs, and cohort dashboards, group-discussion practice and company-specific question sets for placement cells.

ACKNOWLEDGEMENT

The authors thank Prof. V. M. Joshi for expert guidance and continuous encouragement throughout this work, the Principal of Vishwabharati Academy's College of Engineering, Ahmednagar, and the staff of the Department of Computer Engineering for their support.

REFERENCES

1. Mongkoljaturong, K., Manitsakulwong, M., Moodleah, S.: AI-Powered MetaHuman Interviewer: Serious Game for Student Job Interview Skills. IEEE Access 13 (2025) 205505–205520
2. Khallelullah, S., Kumar, V.S., Mahati, M., Soni, G., Sai, C.B.: Analysis of Advanced Technology Integrated Interviews. In: Proc. 8th Int. Conf. Inventive Syst. Control (ICISC) (2024) 613–617
3. Jadhav, B., Sawant, A., Shah, A., Vemula, P., Waikar, A., Yadav, S.: A Comprehensive Study and Implementation of the Mock Interview Simulator with AI and Pose-Based Interaction. In: Proc. 1st Int. Conf. Cognit., Green Ubiquitous Comput. (IC-CGU) (2024) 1–5
4. Qin, C., Zhu, H., Shen, D., Sun, Y., Yao, K., Wang, P., Xiong, H.: Automatic Skill-Oriented Question Generation and Recommendation for Intelligent Job Interviews. ACM Trans. Inf. Syst. 42(1) (2024) 1–32
5. Lee, L.-K., Chan, E.H., Tong, K.K.-L., Wong, N.K.-H., Wu, B.S.-Y., Fung, Y.-C., Fong, E.K.S., Leong Hou, U., Wu, N.-I.: Utilizing Virtual Reality and Generative AI Chatbot for Job Interview Simulations. In: Proc. Int. Symp. Educ. Technol. (ISET) (2024) 209–212
6. Mishra, P.K., Arulappan, A.K., Ra, I.H., Rose, G., Lee, Y.S.: AI-Driven Virtual Mock Interview Development. In: Proc. Joint 13th Int. Conf. Soft Comput. Intell. Syst. and 25th Int. Symp. Adv. Intell. Syst. (2024) 1–4
7. Li, M., Chen, X., Liao, W., Song, Y., Zhang, T., Zhao, D., Yan, R.: EZInterviewer: To Improve Job Interview Performance with Mock Interview Generator. In: Proc. 16th ACM Int. Conf. Web Search Data Mining (2023) 1102–1110
8. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust Speech Recognition via Large-Scale Weak Supervision. In: Proc. Int. Conf. Mach. Learn. (2022) 28492–28518
9. Brooke, J.: SUS: A "Quick and Dirty" Usability Scale. In: Usability Evaluation in Industry, vol. 189. Taylor & Francis, London (1996) 4–7
10. Michael, D.R., Chen, S.L.: Serious Games: Games That Educate, Train, and Inform. Muska & Lipman/Premier-Trade, Chicago (2005)
11. Boyle, E.A., Hainey, T., Connolly, T.M., Gray, G., Earp, J., Ott, M., Lim, T., Ninaus, M., Ribeiro, C., Pereira, J.: An Update to the Systematic Literature Review of Empirical Evidence of the Impacts and Outcomes of Computer Games and Serious Games. Comput. Educ. 94 (2016) 178–192
12. Swartout, W., Gratch, J., Hill, R.W., Hovy, E., Marsella, S., Rickel, J., Traum, D.: Toward Virtual Humans. AI Mag. 27(2) (2006) 96–108
13. Dong, B., Bai, J., Xu, T., Zhou, Y.: Large Language Models in Education: A Systematic Review. In: Proc. 6th Int. Conf. Comput. Sci. Technol. Educ. (CSTE) (2024) 131–134
14. Liesenfeld, A., Lopez, A., Dingemanse, M.: The Timing Bottleneck: Why Timing and Overlap Are Mission-Critical for Conversational User Interfaces, Speech Recognition and Dialogue Systems. In: Proc. 24th Meeting SIGDIAL (2023) 482–495
15. Meta Open Source: React documentation. https://react.dev
16. OpenJS Foundation: Express – Node.js web application framework. https://expressjs.com
17. MongoDB Inc.: MongoDB documentation. https://www.mongodb.com/docs
18. Google: Firebase Authentication documentation. https://firebase.google.com/docs/auth