WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

AI Mock Interview Cheat Detection System

Mrs. N.G. Dharanya[1], Sakthivel K[2], Sagar M[3], Poovel S[4], Navinath M[5]

[1]Associate Professor, Department of Information Technology, Sri Shakthi Institute of Engineering and Technology, TamilNadu, Coimbatore-641062, India

[2]Student, Department of Information Technology, Sri Shakthi Institute of Engineering and Technology, TamilNadu, Coimbatore-641062, India

[3]Student, Department of Information Technology, Sri Shakthi Institute of Engineering and Technology, TamilNadu, Coimbatore-641062, India

[4]Student, Department of Information Technology, Sri Shakthi Institute of Engineering and Technology, TamilNadu, Coimbatore-641062, India

[5]Student, Department of Information Technology, Sri Shakthi Institute of Engineering and Technology, TamilNadu, Coimbatore-641062, India

A B S T R A C T

Online mock interviews are widely used for placement preparation, but most platforms rely on fixed question banks, give generic feedback and cannot detect malpractice such as tab switching or copy-paste. This paper presents the AI Mock Interview Cheat Detection System, a full-stack web platform that combines a Large Language Model (LLM) with a browser-based proctoring engine. The system parses the candidate's resume and the job description, calculates a skill-match percentage and generates a new set of questions for every session. The interview has two rounds: a timed, proctored 20-question assessment with a three-warning rule, followed by a voice interview using the Web Speech API. Each answer is scored on five dimensions by the Claude model, and the readiness report is calculated only from answered questions. The system is built with React, Express.js and Prisma ORM, and functional testing shows answer-specific scoring, reliable violation logging and zero readiness for unattempted sessions.

Keywords: Mock Interview, Cheating Detection, Online Proctoring, Large Language Model, Speech Recognition.

1. Introduction

The growth of remote hiring has moved many assessments to the browser, where candidates can switch tabs, search for answers or paste ready-made text, which makes results unreliable [1]. LLMs can now generate candidate-specific questions and grade open-ended answers against a rubric [2], yet most practice tools still use static questions, fixed scores and no cheating control. This work provides (i) personalised, non-repeating question generation, (ii) a proctored two-round workflow, (iii) a three-warning anti-cheating protocol with an audit trail, and (iv) a five-dimension evaluation whose score depends only on submitted answers.

2. Methodology

A. Resume / JD Analysis and Skill Matching

Resume text (PDF, DOCX, TXT) is extracted on the server and parsed by the LLM into languages, frameworks, tools, databases, projects and level; the job description is parsed into role and required skills. Skills are matched bidirectionally by substring. With R required skills, M matched and P preferred matches, match % = round(85 × M/R + 10 [if P > 0]), bounded to 55–98; missing skills are reported as gaps.

B. Dynamic Question Generation

Each interview gets a unique session ID. The parsed resume, job description, level and skills are sent to the LLM to produce technical, coding, system design, database, project-based, behavioural and scenario questions. New questions are compared with earlier ones for the same candidate and job, and duplicates are regenerated.

C. Two-Round Workflow

Round 1 has 20 questions (10 technical, 10 aptitude) in 25 minutes; a score of 60% or more (12/20) unlocks Round 2, an 8–10 question voice interview. Questions are spoken with Speech Synthesis and answers are captured with Speech Recognition as an editable transcript.

D. Anti-Cheating Protocol

A React hook detects tab switches and minimisation (Page Visibility API [3]), window blur, full screen exit and copy, cut or paste actions (blocked with prevent Default). Each confirmed violation shows Warning 1–3 and is stored in the Security Event table; the fourth violation flags the interview and auto-submits it (Table 1).

E. Evaluation and Scoring

Every answer triggers a new LLM call using only that question and answer, returning strict JSON with technical, relevance, clarity, completeness and communication scores (0–100), strengths, improvements and an ideal answer. Overall score and readiness are computed only from answered questions; with no answers both are 0.

3. Modeling and Analysis

The system uses a three-tier architecture (Fig. 1). The Express API is secured with JWT in http Only cookies [4], Zod validation, Helmet and rate limiting, and calls the Claude API [5] for parsing, generation and evaluation. Prisma ORM stores users, resumes, job descriptions, interviews, questions, answers, evaluations and security events.

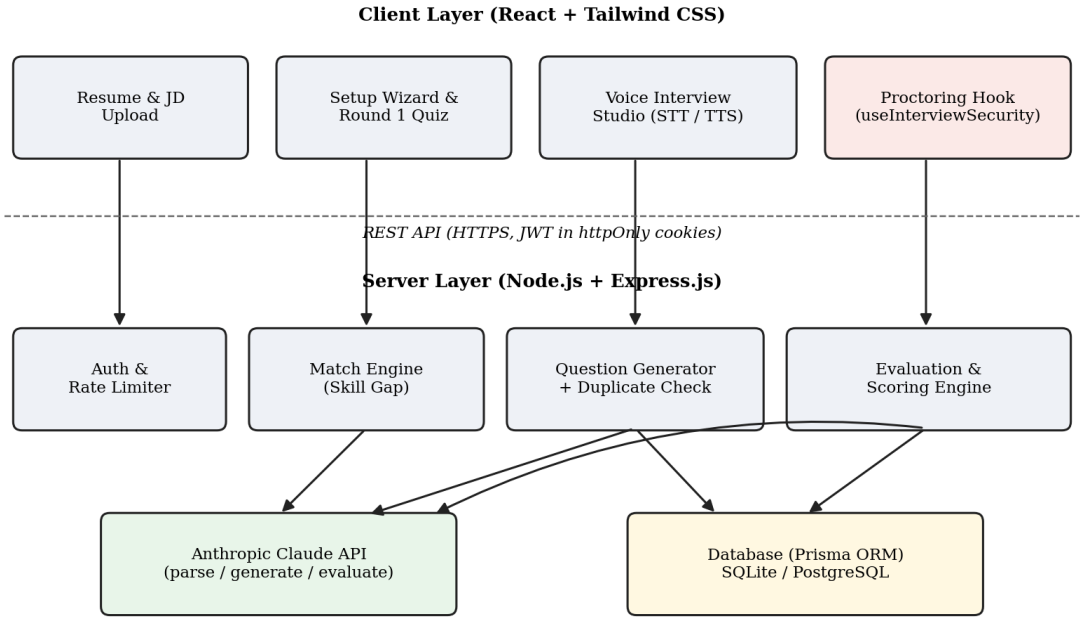


Fig. 1 – System architecture.

Table 1. Monitored events and system response

Event	Detection method	System response
Tab switch / minimise	document.hidden	Log violation, warning modal
Window focus loss	window blur (debounced 1.5 s)	Log violation, warning modal
Fullscreen exit	fullscreen change event	Log violation, warning modal
Copy / cut / paste	clipboard events	Block action, log violation
4th violation	counter greater than 3	Flag, terminate, auto-submit

4. Results and Discussion

Functional testing covered the complete user flow. Fig. 2 shows the Round 1 page with the 20-question structure, 25-minute limit, 60% qualifying score and the anti-cheating notice. Fig. 3 shows the dashboard: seven sessions were recorded, and sessions with no answers are plotted as zero, confirming that no positive score is given to an unattempted interview.

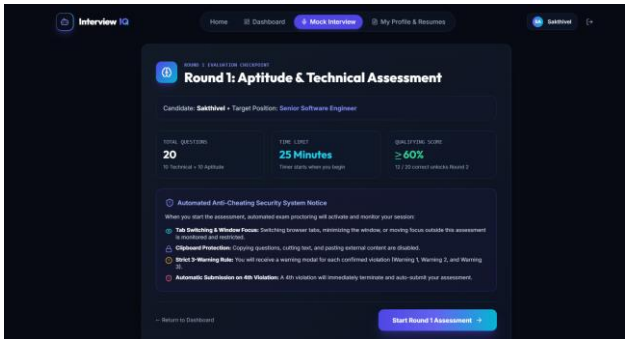


Fig. 2 – Round 1 page with anti-cheating notice.



Fig. 3 – Analytics dashboard.

Table 2. Comparison with existing platforms

Feature	Existing platforms	Proposed system
Questions	Fixed bank, repeated	New set per session from resume + JD
Scoring	Fixed or keyword based	Per-answer, five-dimension LLM rubric
Voice practice	Rarely available	TTS questions, STT answers
Cheating control	None	3 warnings, 4th violation auto-submits
Empty session	May show default score	Always 0

Browser-level proctoring cannot see a second device, so it acts as a deterrent and audit record rather than complete prevention.

5. Conclusion

The proposed platform gives candidates realistic, trustworthy interview practice by combining personalised question generation, a proctored two-round workflow, voice interaction and answer-based AI scoring. Future work includes webcam-based face and gaze monitoring, audio analysis, a coding editor and a recruiter dashboard.

Acknowledgements

The authors thank Dr. S. Prakash (HOD), Dr. M. Deepa (Project Co-Ordinator) and Sri Shakthi Institute of Engineering and Technology for their support.

REFERENCES

1. Y. Atoum, L. Chen, A. X. Liu, S. D. H. Hsu and X. Liu, “Automated Online Exam Proctoring,” IEEE Transactions on Multimedia, vol. 19, no. 7, pp. 1609–1624, 2017.
2. L. Zheng et al., “Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena,” NeurIPS Datasets and Benchmarks Track, 2023.
3. W3C, “Page Visibility (Second Edition),” W3C Recommendation, 2013.
4. M. Jones, J. Bradley and N. Sakimura, “JSON Web Token (JWT),” RFC 7519, IETF, 2015.
5. Anthropic, “Claude API Documentation,” 2026. [Online]. Available: https://docs.anthropic.com (accessed Oct. 24, 2026).