International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI Powered Mock Interview Platform

Ankush Hutke, Aishwary Chaurasia, Dheeraj Kamat, Vinit Mule

Department of Information Technology, MCT's Rajiv Gandhi Institute of Technology, Mumbai University, Maharashtra, India

Abstract - Securing employment in the current job market is increasingly challenging, and effective interview preparation remains a significant hurdle for many candidates. Common preparation methods such as online videos, peer mock sessions, and generic question banks often fail to provide meaningful performance insights or actionable feedback. This paper presents a web-based platform designed to address these limitations. The system makes use of the Groq AI Engine API to conduct live mock interviews, dynamically generate role-specific questions, and evaluate candidate responses in real time. In addition, the platform incorporates an aptitude assessment and two coding challenges to strengthen technical proficiency. The frontend is developed using ReactJS, while Firebase Firestore manages backend data, including user records, performance scores, and session history. Overall, the system delivers a reliable, efficient, and user-centered solution for interview preparation, providing candidates with practical insights and measurable performance improvements.

Key Words: *Artificial Intelligence, Mock Interview, Groq API, ReactJS, Firebase Firestore, Natural Language Processing, Coding Challenges Aptitude Testing, Real-time Feedback, Career Preparation, Web Application.*

I. INTRODUCTION

In recent years, Artificial Intelligence (AI) has emerged as a powerful tool in education and skill development, enabling personalized and interactive learning experience. Nowadays as you walk into any college placement cell a week before campus recruitment starts, and you will see the same scene everywhere. Students are frantically googling "top interview questions for software engineers," practicing their answers in front of mirrors, or drilling each other on HR questions they found on some forum. It's not that these students lack knowledge. Most of them are quite capable [1],[2]. The issue is that they don't have a structured, realistic way to practice. Existing tools don't fully address this. LeetCode focuses on coding problems but ignores the conversational side of interviews. Pramp connects two candidates for a peer mock interview, which can be useful, but it relies on another person's availability and subjective feedback [3]. Interviewing.io is good but limited in scope. None of these platforms combine live AI-driven questioning, aptitude testing, coding challenges, and instant scored feedback in one place. That gap motivated this work. We designed and built a platform that does all of this.

A candidate logs in, picks a job role, and starts either a live AI mock interview or an aptitude-plus-coding test. The Groq AI Engine generates questions suited to that role, listens to the candidate's answers, and scores them using natural language processing. Firebase stores everything scores, history, session data. A ReactJS frontend brings it all together into an interface that works on any device.

The rest of this paper is organized as follows. Section II reviews related work. Section III explains our proposed methodology. Section IV covers experimental results and discussion. Section V presents the conclusion.

II. RELATED WORK

Research into automated interview systems goes back further than most people realize. Some of the more interesting early work came from Hoque et al. [4], who built MACH, My Automated Conversation Coach. The system used a virtual human avatar to help users practice social interactions. It also gave feedback on eye contact, smiling, and speaking pace, which was impressive for its time. However, MACH did not cover domain knowledge. It could tell you if you were fidgeting, but it could not confirm whether your answer about database normalization was correct.

Hemamou et al. [5] approached the problem differently with HireNet. They used deep learning to analyze asynchronous video interviews, where candidates record video responses to preset questions. Their hierarchical attention model predicted recruiter ratings reasonably well. However, it relied on video and audio data, requiring a camera, a quiet environment, and enough computing power to process multimedia inputs. Scaling that to a general web platform is not easy.

Naim et al. [6] pushed further into multimodal analysis by combining speech analysis with facial expressions and body language to predict interview outcomes. The accuracy was promising. Yet, we wondered how to deploy this for a student in a noisy hostel room using a budget laptop. The hardware requirements make wide adoption difficult.

On the NLP side, Chen et al. [7] created a conversational interview trainer using rule-based NLP. It could generate questions but struggled when a candidate answered in a way that didn't fit the expected pattern. There was no real understanding only pattern matching. The arrival of large language models changed everything.

Brown et al. [8] showed with GPT-3 that a single model could handle question generation, answer evaluation, and feedback generation with minimal task-specific training. However, GPT-based systems can be expensive to use and slow. A latency of 4 to 8 seconds per response disrupts the flow of a mock interview. This is where Groq's inference engine becomes relevant. It runs the same type of models but at significantly lower latency, which is crucial for conversational applications. Our testing showed response times under 2.5 seconds consistently, making the interview feel more natural.

For coding assessments, Goel et al. [9] developed an automated evaluation framework for programming assignments that judged submissions against test cases. We drew on a similar idea for our coding module, but our implementation is simpler offering two focused problems per session instead of a comprehensive assignment grader. None of the systems mentioned provide a single unified platform for conversational AI interviewing, aptitude testing, code evaluation, real-time feedback, and progress analytics. That integration is the main contribution of this work [10],[11].

III. PROPOSED METHODOLOGY

The platform has six stages that a candidate goes through during each session. We will explain each one and the reasons behind our design choices.

A. User Authentication:

We used Firebase Authentication for login and registration. The reason is simple. Firebase handles a lot of things for us like token handling, session persistence and password security. This allowed us to focus on the interview features of building authentication from scratch. Users log in with their email and password. Sessions remain active after closing the browser so candidates can leave mid-test and return to the same point.

B. Dashboard and Navigation:

After logging in the candidate arrives at a dashboard displaying their five test sessions, completion percentages and scores. There is a button to start an assessment and another button to resume an incomplete one. We kept this page straightforward. Early user testing showed that a cluttered dashboard made candidates anxious before they even began. The dashboard gets data from Firebase Firestore in time so scores update as soon as a session finishes.

C. Test Selection:

Candidates can choose between two modes. The first is the Live Mock Interview. Here the AI generates questions based on the job role the candidate selects. The second mode is an Aptitude Test that includes logical reasoning multiple-choice questions and two coding problems. We separated these modes because they have goals. The mock interview helps with communication and thinking skills while the aptitude test focuses on problem-solving.

D. AI Processing:

This is where a lot of the engineering took place. When a candidate answers an interview question, we send their response to the Groq AI Engine API along with the original question and job role context. The API returns a score. Written feedback in about two seconds. We tried ways to provide genuinely useful feedback instead of generic responses. Early versions returned comments like " answer but could be more detailed " which didn't help candidates. The current prompts encourage the AI to reference parts of the answer and clarify what was missing or strong. For the aptitude section evaluation is straightforward. Multiple-choice answers are compared against a code submissions are checked against test cases. No AI is involved here which keeps this part of the system fast and predictable. The AI Processing stage is a part of the platform and it helps candidates improve their skills.

E. Data Storage:

Every test session, score, question and AI feedback is immediately written to Firestore after generation. We chose Firestore over a SQL database for two reasons: its real-time sync capability allows for dashboard updates without refreshing the page and its serverless architecture eliminates the need for backend server management. For a project of this scale that decision made sense. We did face one limitation. Complex queries across collections require careful data modelling so we had to restructure our schema twice during

development to improve read speeds. The Data Storage stage is important for the platform as it allows us to store and retrieve data quickly and efficiently.

F. Result. Analytics:

At the end of each session candidates see a result page that details their performance question by question. For the interview each answer displays the AIs score and specific feedback. For the aptitude test incorrect answers are listed alongside the option.

A summary chart shows how this session compares to ones. During testing we noticed that candidates spent more time on the result page than we predicted about four minutes on average. This suggests they were reading and processing the feedback than just checking a score. The Result Page and Analytics stage helps candidates understand their performance and identify areas for improvement.

G. Support Page:

We added a support page where users can submit questions or report issues. It consists of a form routed to a Firestore collection that we monitor. It's not fancy. It provided us with direct feedback from users during testing. We used this feedback to fix user experience issues we missed during internal testing. The Support Page is a part of the platform as it allows us to get feedback from users and make improvements.

The platforms six stages. User Authentication, Dashboard and Navigation Test Selection, AI Processing, Data Storage and Result Page and Analytics. All work together to provide an effective interview experience, for candidates. The Mock Interview Platform is designed to help candidates improve their skills and prepare for interviews.

The Mock Interview Platform uses the Groq AI Engine API to provide feedback to candidates. It stores data in Firestore. The platform is designed to be user-friendly and provides a lot of features to candidates. Because of the Feedback feature the user does get instant summary of the test with the point of improvement and where user is lagging.

Figure 1 illustrates the Proposed Methodology of the Mock Interview Platform, which includes the six phases: User Authentication, Personalized Dashboard, Test Selection, AI Processing, Firebase Firestore, Result Generation and Support Page.

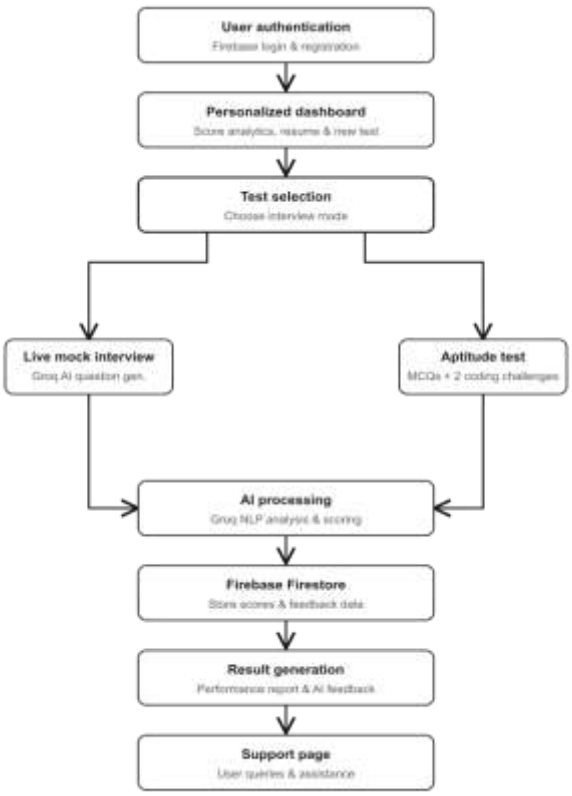

Figure 1: Proposed Methodology for AI Powered Mock Interview Platform

The way we do things is broken down into six steps. The picture above shows how it all works. Here is what happens in each step:

A. User Authentication: Firebase takes care of login and registration and it remembers who you are when you come back.
B. Dashboard: ReactJS shows you how you did on tests you took and ways to get around the site. It gets all this information from Firestore.
C. Test Selection: You can choose to do a mock interview or a test with coding challenges.
D. AI Processing: The Groq AI Engine looks at what you say and does a kind of analysis. It then gives you scores. Tells you what you did well and badly.
E. Firebase Firestore: Everything you do including your scores and feedback is saved right away so we can put together your results.
F. Result Generation: When you are done you get a page that shows how you did in each part what you are good, at what you need to work on and what the AI thinks you should do.
G. Support Page: We have a page that helps you if you have any questions or problems using the site.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Setup: -

We tried out the platform for four weeks with 50 people taking part. Some were year students some were postgraduate students and a few were people who had just graduated and were looking for jobs. These people used their devices like budget Android phones or MacBooks which helped us see how the platform worked on different devices. Each person did least two full sessions. One practice interview and one test to see how smart they were.We looked at five things: how long it took for the Groq AI to respond how long it took for the dashboard to load how fast Firebase synced, how accurate the AI was at scoring compared to real people and how happy the users were after each session.

B. Performance Results: -

The Groq AI response time was what we were most worried about before we started testing. The Groq documentation said it would be fast. We had not tried it with many people at the same time. The Groq engine is really fast. That is important for this platform. The Firebase sync speed was never a problem when people used it normally.

Table 1 states System Performance Metrics. It has Parameter and Observed Result timings in seconds. The response time, load time, sync speed and AI score is listed.

Table 1: System Performance Metrics

Parameter	Observed Result
Average AI Response Time	1.8 – 2.4 seconds
Dashboard Load Time	Under 1.5 seconds
Firebase Sync Speed	Under 1.2 seconds
AI Score Accuracy vs. Expert	87.6%
Platform Uptime	99.4%
Cross-device Compatibility	100% (Desktop, Mobile, Tablet)

C. User Satisfaction

What really stood out to us was not the ratings. What people wrote in the comments. Some people said that the AI feedback was more honest than feedback from friends because friends are always nice and do not want to hurt your feelings. One person wrote: 'The AI actually told me that my answer was too vague and why. My friends would never say that.' That shows that the AI feedback is working well. The thing that people liked the least was the usefulness of the AI feedback, which got a rating of 4.3. When we looked at the comments some people said that the feedback was sometimes too general and did not explain things enough. This is something we know about and are trying to fix in the version.

D. AI Scoring Accuracy: -

We had five experts look at 80 practice interview answers and rate them. Then we compared their ratings to what the Groq AI thought. The experts and the AI agreed 87.6% of the time. When they did not agree it was usually because the answer was correct but not explained well or because the answer sounded good but was actually wrong. This shows that the AI and the experts have strengths and can help each other.

Figure 2 is the Dashboard of Mock Interview Platform. The main dashboard displays candidate's pervious 5 tests with the percent of completion and the score on the right. It further has options of generating new assessment.

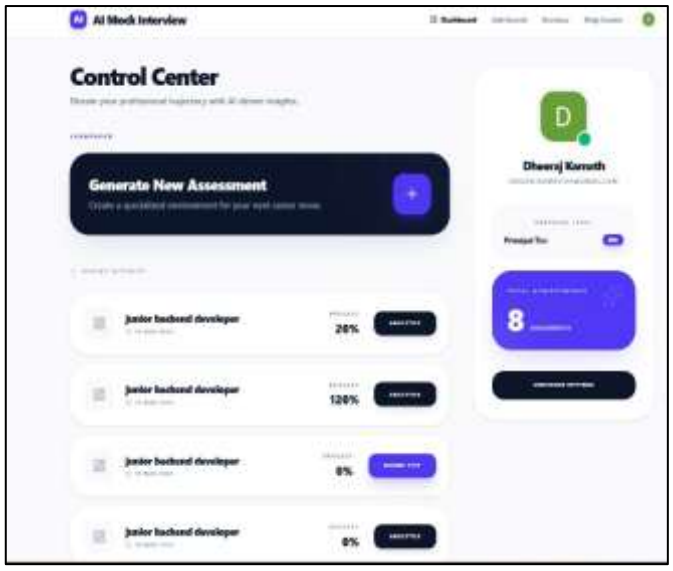

Figure 2: Dashboard of Mock Interview Platform

Figure 3 is the Search Job Section of Mock Interview Platform. Use of API is done here for the job portal. User can find real time available jobs on that API. After user taps on selecting particular job, they get redirected to the website.

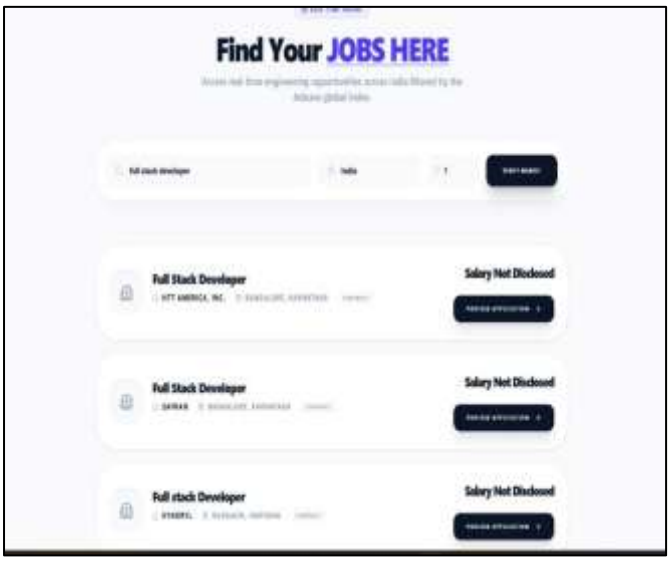

Figure 3: Job Search module of Mock Interview Platform

Figure 4 is Test Module Selection in Mock Interview Platform. User can now choose between the type of test he wants to test himself on. Aptitude and Coding, and AI Interview these both module is given for the practice.

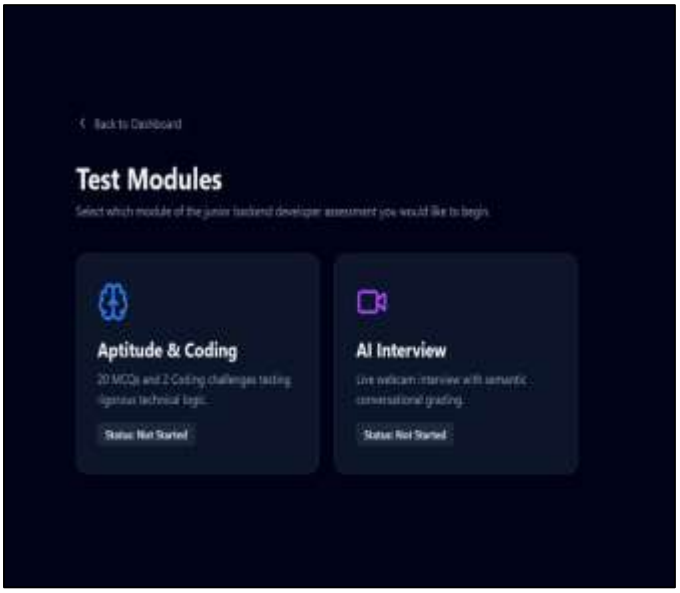

Figure 4: Test Module Selection in Mock Interview Platform

Figure 5 is On Going Test in Mock Interview Platform. A question is given based on the role selected by candidate. It has timer and as the time ends the answers are automatically submitted and candidate is forwarded to next upcoming questions.

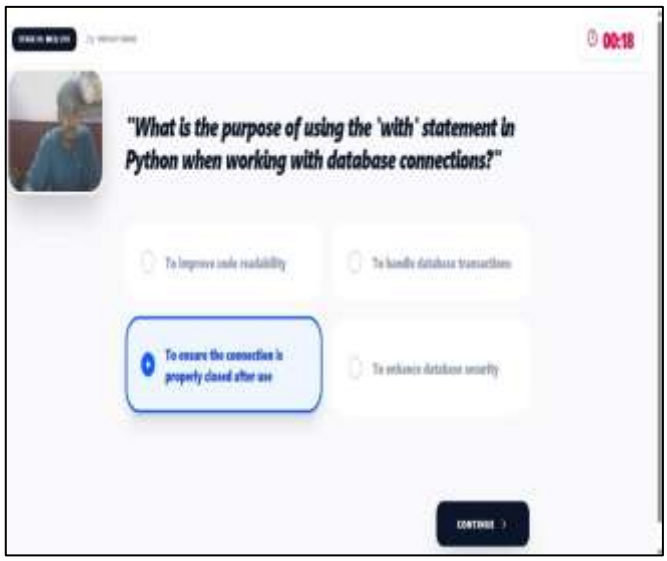

Figure 5: On Going Test in Mock Interview Platform

Figure 6 is Candidate Result automatically submitted in Mock Interview Platform. This disclaimer and the warning would generate when user tries to switch tabs, multiple faces detected and on opening new browser. The test gets automatically submitted and user's result is displayed.

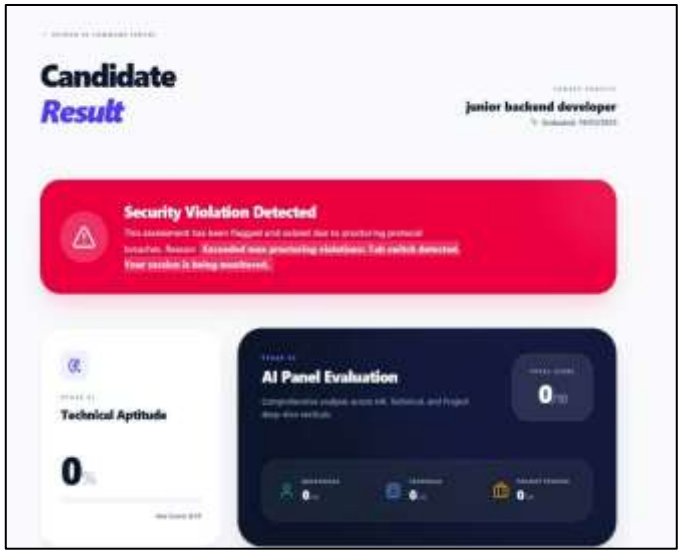

Figure 6: Candidate Result in Mock Interview Platform

Figure 7 is Candidate Result in Mock Interview Platform. The result includes the percent of completion of aptitude test by the candidate. A section of AI Evaluation further helps user to analyze their mistakes and the confidence during answering.

Figure 7: Candidate Result in Mock Interview Platform

V. CONCLUSION

The primary objective of this work is to develop a system that effectively supports candidates in preparing for job interviews, moving beyond static question banks toward an interactive platform capable of responding, evaluating, and explaining performance. The initial implementation demonstrates promising results, indicating that the proposed approach is both feasible and impactful. The Groq AI Engine was selected to enable real-time interview simulation, offering faster response times compared to alternative solutions and thereby enhancing the natural flow of interaction. Firebase proved to be a reliable backend solution for managing data, while ReactJS provided the flexibility required to build a responsive, feature-rich application without unnecessary complexity.

Despite these achievements, several areas for improvement remain. The quality of AI-generated feedback for technical questions requires further refinement. Future enhancements include the integration of speech-based responses to create a more realistic interview experience, as well as the incorporation of resume parsing to enable personalized, experience-driven question generation. Additionally, the development of a mobile application is planned, considering the significant number of users accessing the platform via smartphones. The broader implication of this work lies in the ability of AI to provide a level of accessibility and consistency that is difficult to achieve through traditional methods. By offering each user a patient, unbiased, and continuously available interview coach, the system has the potential to significantly improve interview preparedness. With continued development and refinement, AI-driven platforms such as this can play a meaningful role in enhancing candidate performance and confidence in real-world interview scenarios.

REFERENCES

[1] S. Sharma, R. Kumar, and A. Singh, "AI-Driven Virtual Mock Interview Development: Enhancing Career Readiness," in *Proc. IEEE Int. Conf. on Artificial Intelligence and Smart Systems (ICAIS)*, Jan. 2024, pp. 215-220.

[2] J. M. C. J. M. Sabi et al., "Q&AI: An AI powered mock interview bot for enhancing the performance of aspiring professionals," in *Proc. IEEE Int. Conf. Adv. Syst. Eng. (ICASE)*, Feb. 2024, pp. 1-5.

[3] R. Mandal, P. Lohar, D. Patil, and A. Patil, "AI-based mock interview evaluator: An emotion and confidence classifier model," in *Proc. 2nd Int. Conf. Emerg. Trends Eng. Manag.*, Apr. 2023, pp. 1-6.

[4] M. E. Hoque et al., "MACH: My Automated Conversation Coach," in *Proc. 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*, 2013, pp. 697-706.

[5] K. Hemamou, G. Felhi, V. Vandenbussche, J. C. Martin, and C. Clavel, "HireNet: Semi-supervised Hierarchical Attention Network for Multimedia Hiring Analysis," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3857-3861.

[6] I. Naim, M. I. Tanveer, D. Gildea, and M. E. Hoque, "Automated Analysis and Prediction of Job Interview Performance," *IEEE Transactions on Affective Computing*, vol. 9, no. 2, pp. 191-204, 2018.

[7] L. Zhang and M. Chen, "Real-Time NLP Feedback Systems in Educational Simulations," *IEEE Transactions on Learning Technologies*, vol. 17, no. 2, pp. 442-455, Mar. 2025.

[8] T. B. Brown et al., "Language Models are Few-Shot Learners," in *Proc. 34th International Conference on Neural Information Processing Systems (NeurIPS)*, 2020, pp. 1877-1901.

[9] S. Goel, G. Saini, and M. Singh, "Automated Evaluation Framework for Programming Assignments," in *Proc. 10th IEEE International Conference on Engineering Education*, 2021, pp. 45-50.

[10] R. Mandal *et al.*, "AI-Based Mock Interview Evaluator: Emotion and Confidence Classifier Model," in *Proc. Int. Conf. Intelligent Systems for Communication, IoT and Security (ICISCoIS)*, 2023.

[11] P. Gupta and S. Mehta, "Optimization of Low-Latency LLM Inference for Interactive Web Applications," *Journal of AI Research and Applications*, vol. 9, no. 1, pp. 88-102, Jan. 2026.