

Integrated Interview And Coding Practice Platform

M. Rudra Kumar
Professor, IT Department, Mahatma
Gandhi Institute of Technology,
Hydera-
bad, Telangana

Abdul Rasheed,
²Student, IT Department, Mahatma
Gandhi Institute of Technology,
Hyderabad,
Telangana

Kanaka Sharon
Student, IT Department, Mahatma
Gandhi Institute of Technology,
Hyderabad,
Telangana

Abstract: — Interview preparation websites are gaining popularity as candidates are seeking tech as a means to have their technical and soft skills adjust. This is even though the majority of the currently available platforms have blatant flaws in the form of lack of structured feedback, no multi-modal interaction, no interview practice as part of aptitude and coding tests, and vice versa. We are suggesting a Multi-Mode Interview and Coding Practice Platform that enables interview simulating, aptitude testing and general codifying practice in one working platform. The user will practice syntax or verbal based interview questions, we will take and store more contextual information in the form of transcripts and visual scores to evaluate them later in a structured manner. The system was developed with React as the frontend user interface, Next.js as the frontend service, FastAPI as the backend service, Firebase as the authentication and cloud based data storage and Judge0 API as the code execution in a controlled cloud based secure setup. The system, which is known to have test-case-based evaluation, also includes aptitude testing modules and analysis and code assessment modules to this basic functionality. The analytics of progress tracking in the form of cloud based dashboards assist the user to realise some trends on a more extended period of time. We hypothesize in this paper that multi-modal interaction, structured evaluation mechanisms and secure execution environments significantly enhance performance, reliability and user-experience of interview preparation systems and scale and accessibility maintainability.

Keywords: Interview Preparation, Coding Platform, Aptitude Testing, FastAPI, Firebase, Judge0, Multi-Modal System.

1. INTRODUCTION

Interview preparation is defined as the practice through which applicants engage in preparation in order to excel in a job interview. The technical rounds are usually concerned with the programming and other related issues, whereas the non-technical ones are concerned with the psychometric and HR interviews. The candidates mostly spend their practice time on going through a set of static question banks, practice on mock interviews and do their aptitude and code practice using different systems. Although these approaches have been in wide use, they still have several disadvantages such as lack of structured feedback, lack of real-time assessment, lack of Customization, and lack of coherence among various learning platforms. Those problems allow candidates to underestimate their skills and preparedness to real-life interview, and lack of them with a unified system can provide them with structured assessment, live-time feedback, and combined practice in various areas. Intelligent and interactive e-learning solutions have been made possible by trends in web technologies, cloud computing and artificial intelligence.

They are able to use several modalities like text, speech, visual data, etc, to offer more analysis of user performance. Furthermore, its cloud-based nature allows the safe storage of the user information and tracking of progress over time to develop scalable and efficient system that will allow improving the user experience without excluding the possibility of precise and consistent evaluation systems.

To address the weaknesses of the current systems this study presented a Multi-Modal Interview and Coding Practice Platform. In this system, the user has the chance of practising the interview questions through voice or text based input in storing the contextual information to be evaluated in an organised manner. It is written in React and Next.js to design the frontend interface and FastAPI to execute them, Firebase to authenticate and store them in a cloud, and Judge0 API to run them safely. It also has analytics dashboards and progress tracking capabilities, allowing users to keep track of the fact that they are progressing. This system will be able to prepare the interviews in a more efficient and secure way within a shorter time by incorporating the use of multi-modal interaction, safe execution and a structured evaluation.

2. RELATED WORK

Due to the blistering development of artificial intelligence and web technologies, the systems of interview preparation have been developed to offer automated feedback, on-the-fly interaction, and performance analytics. These systems seek to address the shortcomings of the traditional methods of interview preparation including failure to customize interviews, slow responses in terms of feedback and disjointed learning materials. A number of researches have pointed out that AI-based interview platforms can optimize the performance of candidates by offering organized assessment, immediate feedback, and simulation of actual interview situations [1], [3], [7]. Besides, recent trends in intelligent learning systems

highlight the significance of incorporating numerous assessment modules and adaptive feedback systems to enhance the user engagement and performance in terms of learning.

AI-Based Interview Preparation Systems: A number of studies have been conducted investigating AI-based interview preparation systems. Patil et al. [1] suggested a deep learning-based mock interview system that assesses the candidate based on speech-to-text, facial expression, and behavioral features. Likewise, Jagtap et al. [3] created a real-time interview simulation application that combines voice recognition, facial analysis, and AI-based feedback to develop communication skills. Rao et al. [10] proposed an AI-based mock interviewing system, which can create dynamic questions and give a personalized feedback based on natural language processing methods. Such systems show that AI is effective in enhancing the results of interviews; the majority of these systems put emphasis on the simulation of an interview and fail to offer an opportunity to engage in a comprehensive environment encompassing technical and aptitude-based tests.

Multi-Mode and Behavioral Analysis Systems: The contemporary interview platforms are more directed towards the analysis of both verbal and non-verbal messages. Khapekar et al. [6] have highlighted the need to assess the emotional and behavioral levels of tone, confidence, and facial expressions during interviews. On the same note, Rangnath et al. [3] emphasized that the feedback is more accurate and the user more engaged with the combination of facial analysis, voice processing, and conversational AI. These methods will give more insights into the performance of the candidates and assist in the discovery of communication gaps, though, they tend to be limited to behavioral analysis and are not combined with coding evaluation and structured testing mechanisms.

AI-Driven Hybrid Interview Systems: Newer systems seek to offer a full suite of solutions by integrating various features. The AI-based platform suggested by Kotthari et al. [9] comprises resume analysis, identifying skill gaps, simulating interviews, and generating NLP-based questions. Nithya et al. [8] have investigated intelligent interview preparation systems that use machine learning, sentiment analysis, and predictive analytics to deliver personalized career advice and tracking performance. Such platforms emphasize the advantages of integration and personalization, though, they still cannot provide the real-time code execution and unified architecture of the systems which will help to provide the smooth interaction between various modules.

Coding Interview and Collaboration Platforms: Coding assessment systems are also needed to prepare the technical interview. Madan et al. [2] created an online interview system that combines collaborative coding systems, video conferencing, and real-time communication. Equally, Surendra et al. [4], [5] suggested AI-assisted coding interview systems that help execute codes in real time, automatically assess them, and detect plagiarism through secure sandbox systems. Additionally, research on online judge systems [11] highlights the significance of these systems in automated programming evaluation, scalability and effective evaluation of large number of submissions of codings. Another important issue brought up by Došilović et al. [12] is the design of scalable and efficient code execution systems that guarantee secure and efficient execution of programs posted by users. Although these systems enhance accuracy in technical assessment as well as reliability in execution, they are in most cases restricted to coding procedures and not on communication skills and aptitude testing in a single system.

Generative AI and Adaptive Interview Systems: The recent development of generative AI has made adaptive and personalized interview systems possible. Suguna et al. [7] have suggested an interview platform with Gen-AI which is capable of dynamically creating questions and giving feedback on the performance of users. Furthermore, Nofal et al. [11] showed how generative AI and virtual environments can be used to simulate real-world interview settings and enhance training of the candidates. These systems add to personalization, flexibility, and interactivity, but they are not often connected to formal evaluation systems, coding assessment packages, or full analytics systems necessary in the preparation of interviews on a holistic basis.

Research Gap:

It is clear that, as per the literature, the available systems are concentrated on a particular aspect like interview simulation, coding assessment, or behavioral analysis. Nevertheless, a cohesive system that combines multi-modal interview analysis, coding experience, and aptitude assessment as well as cloud-based analytics and tracking progress is lacking. The proposed system will solve these limitations by offering an integrated, scalable, and holistic solution to the interview preparation.

3. METHODS AND MATERIALS

The proposed system aims at offering an integrated and holistic system of interview preparation by incorporating interview simulation, coding assessment, and aptitude test into a single system. The system is aimed at providing a structured evaluation, real-time feedback, and constant performance monitoring on the basis of modern web technologies and cloud-based services. The proposed system has the architecture of frontend interface, backend processing system, code execution engine and cloud storage services that collaborate to guarantee smooth interaction and effective system operation.

Frontend Interface Development:

React.js and Next.js will be used to develop the user interface of the system to ensure that the interface is responsive and user-friendly. The interface should also be capable of supporting several modules such as interview practice, coding exercises and aptitude test. The system allows interaction with the user in the form of text-based responses and code editors and dynamically shows outputs, feedback, and analytics. The frontend provides a seamless flow of navigation between modules and real-time interaction with the backend services by making API calls.

Backend System Integration:

The implementation of the backend is performed with the help of the FastAPI framework, which performs the work of accepting requests, processing them, and exchanging data with the other system components. It is important to process user inputs, handle session data and interface with other external services like the code execution engine. Validation of inputs, generation of responses and storage of results are also done in the backend making the system performance efficient and scalable.

Code Execution Engine:

The coding module is connected to a secure execution engine on the basis of the Judge0 API that allows executing the user-submitted code in real time on a variety of programming languages. The system sends the code and input test cases to the execution engine which executes the request under a sandboxed environment to maintain security and isolation. This is then sent back to the backend and presented to the user. This strategy enables the safe implementation of programs without impacting the host system and automated assessment of the coding tasks.

Multi-Modal Evaluation Mechanism:

The system uses a multi-modal evaluation mechanism to evaluate the performance of users in various dimensions. The user feedback in the interview module is judged in terms of relevance, clarity and form whilst response timing and the nature of response is also taken into account. The system gives a structured feedback to enable users know their strengths and improvements. During aptitude module, the performance is rated in terms of accuracy and time taken, and the users are allowed to work on the problem-solving ability.

Cloud Storage and Authentication:

Firebase is applied to user authentication and cloud-based data storage. It has secure log in features and it stores user specific information like performance history, coding outcomes and interview responses. Cloud-based approach helps users to have access to their progress even when they are not present and also offers scalability to support more than one user at a time.

User Interaction and Module Interaction:

The system incorporates various modules in one interface enabling users to alternate between interview practice, coding and aptitude testing without any form of interference. All modules communicate to the backend in isolation but share common user information and analytics. Such integration guarantees the uniformity of user experience and the absence of multiple platforms.

System Workflow and Process:

1. The system is authenticated by the user through secure authentication systems.
2. The system authenticates users and provides access to the platform.
3. The user picks an option that he likes (Interview / Coding / Aptitude).
4. The user makes input either in the form of interview answers, a code submission or aptitude answers.

5. The input data are allowed into the backend via API requests by the frontend.
6. The input is processed by the backend which determines the operation needed.
7. In the case of coding, the back end sends the request to the code execution engine (Judge0 API).
8. In the case of interview and aptitude modules, the backend is connected to the evaluation system.
9. The system reads the input and produces the right results and feedbacks.
10. The output is processed and sent back to the frontend where it is shown to the user in real time.
11. The system logs performance, results and feedback within the cloud database.
12. The stored data can be accessed to enable a user to refer to it in future and to analyze its performance.

4. EXPERIMENTAL STUDY

This part will outline the process of developing and conducting an experimental test of the suggested multi-modal interview and coding practice platform. The system deployed was also tested to assess how it managed user interactions, processed interview responses, and ran code safely and was also tested on how it gave structured feedback on various modules. The communication of frontend interface with back-end server, cloud services and code running engine was also reviewed in order to assure the overall functionality of the system in a way that is efficient and reliable.

The suggested system was built with the help of modern web technologies, and the frontend interface was built with React.js and Next.js to offer the user an interactive and responsive interface. It has created the layer of backend with the assistance of the FastAPI layer that managed API calls, received user input, and maintained communication between various parts of the system. Authentication and cloud storage were implemented via Firebase, which allows users to log in safely and store performance data permanently. The Judge0 API was also added to provide the possibility to run user-submitted code in a safe and real-time environment.

To test the system, various test scenarios were developed to model the actual user interactions in all the modules. The interview module allowed users to give answers to various kinds of questions, and the system was able to analyze these answers with respect to relevance, clarity, and structure. Multiple-choice questions that were of different levels of difficulty were used to test aptitude module to determine accuracy and response time. During the coding module, users provided programs in other programming languages, which were then run via the Judge0 API, and results compared to the expected results.

The system was also put to test on the number of users and requests that it could handle simultaneously. Various user accounts were set up and different conditions of real world usage were simulated. The authentication system was tested to make sure that the platform and their data could be accessed by authorized users only. Firebase authentication was implemented to test secure login and session management, and cloud storage was implemented to test the consistency and availability of user data across sessions.

The frontend and backend components were thoroughly tested through the process of API requests and checking their accuracy and latency of responses. The interface with the code execution engine was also put to test to ensure that submissions of code were done right and results would be returned without delays. The confined execution environment provided an environment in which user code could be safely executed without impacting the system.

The system exhibited consistent performance when being tested on all modules. The interview evaluation module was consistent and structured in feedback, the coding module was precise in executing programs with minimal response time, and the aptitude module was suitable in measuring performance of the user. The platform was able to sustain the user session information and performance history to be accessed later.

Moreover, the system also made sure that user interactions are handled effectively without losing data or having inconsistency. The cloud architecture has made it possible to scale and to provide good performance even when many requests are made at a time. The experimental analysis indicates that the proposed system successfully incorporates various modules into one platform, which offers a smooth and effective environment to prepare interviews.

5. RESULTS

The findings of the experimental assessment prove that the suggested system has been effective to offer a dependable and efficient interview preparation system by combining the interview simulation, coding test and aptitude test in one environment.

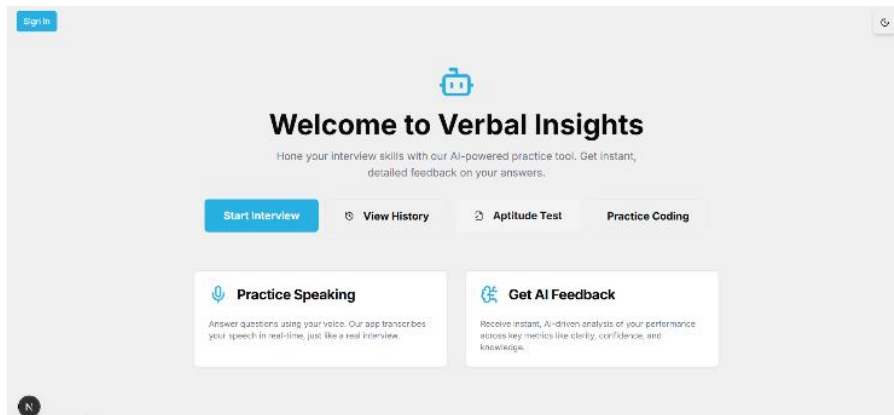


Fig 1. The interview preparation site home page.

This number represents the primary interface of the system, which has an opportunity to use a variety of modules, including interview practice, aptitude tests, coding, and performance history. It acts as the navigation point of users.

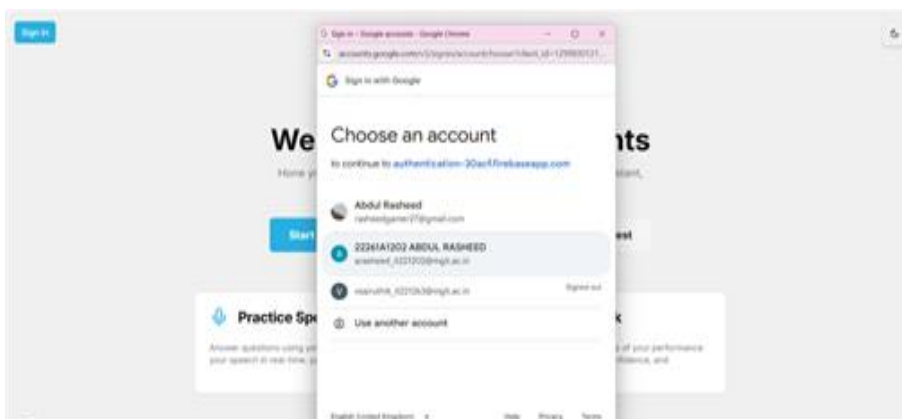


Fig 2. Google sign-in (Firebase authentication) user authentication.

This character depicts the safe log-in service with Google authentication. It makes sure that the platform and information of authorized users are only accessed by them. The process of authentication is fast, dependable, and easy

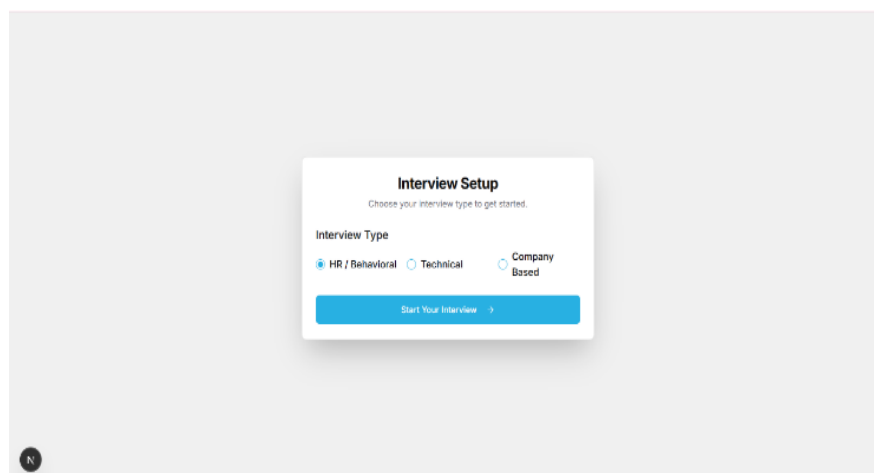


Fig 3. The interview setup interface to choose the type of interview

6

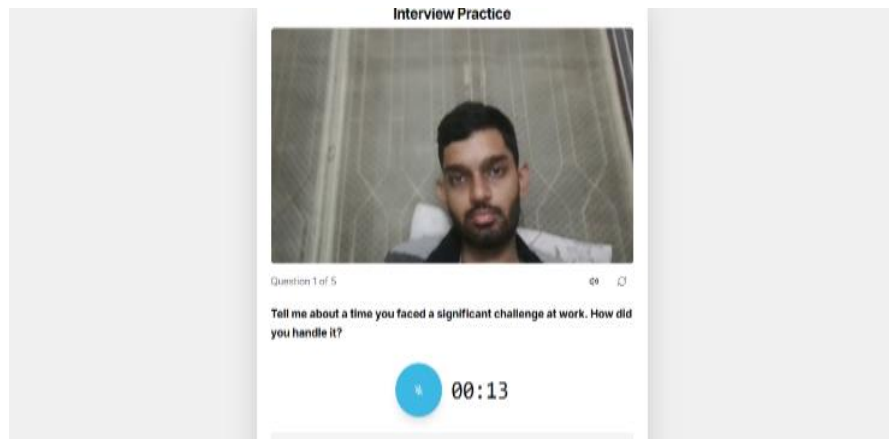


Fig 4. Real-time question and response capture interface

This characterization demonstrates the live interview simulation model where users respond to questions by voice recognition. The system records real-time responses and offers a timer to make structured responses. It is very similar to a real interview situation.

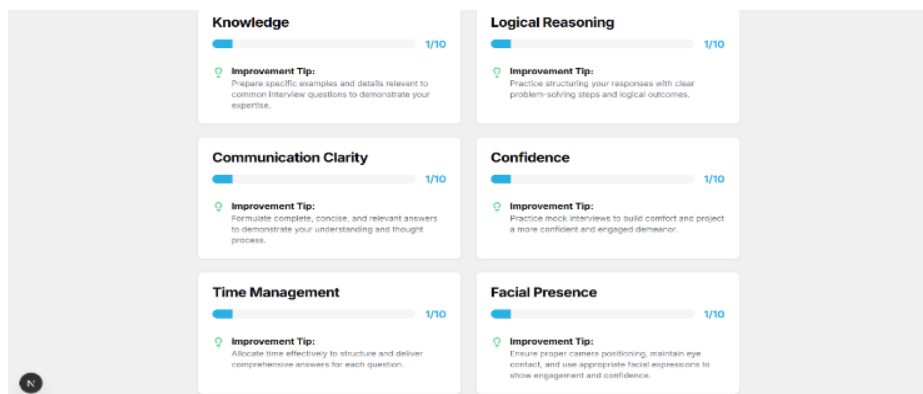


Fig 5. Interview performance analysis and dashboard.

This number shows the feedback information that was given after a session of an interview. It contains general scores and parameter analysis like knowledge, confidence, and communication. The system also offers better performance improvement recommendations.

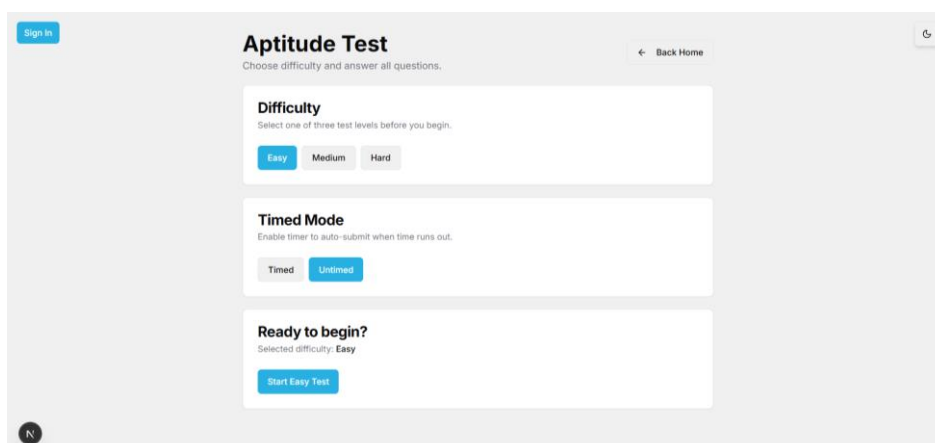


Fig 6. Aptitude test interface with difficulty level choice and setting up of the test.

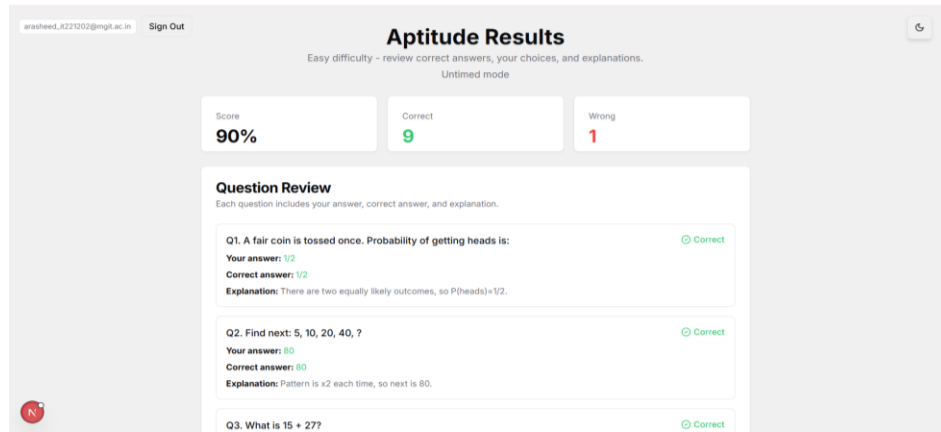


Fig 7. Aptitude test Results

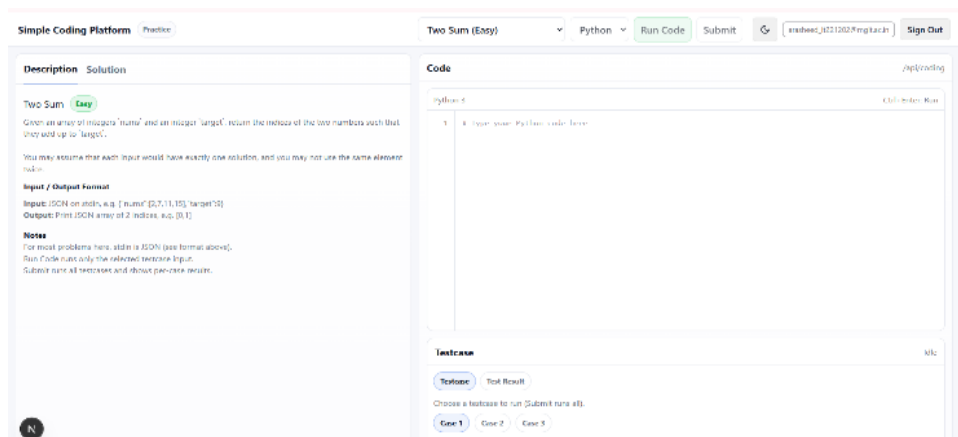


Fig 8. Coding platform user interface with problem description, code editor and testcase execution panel

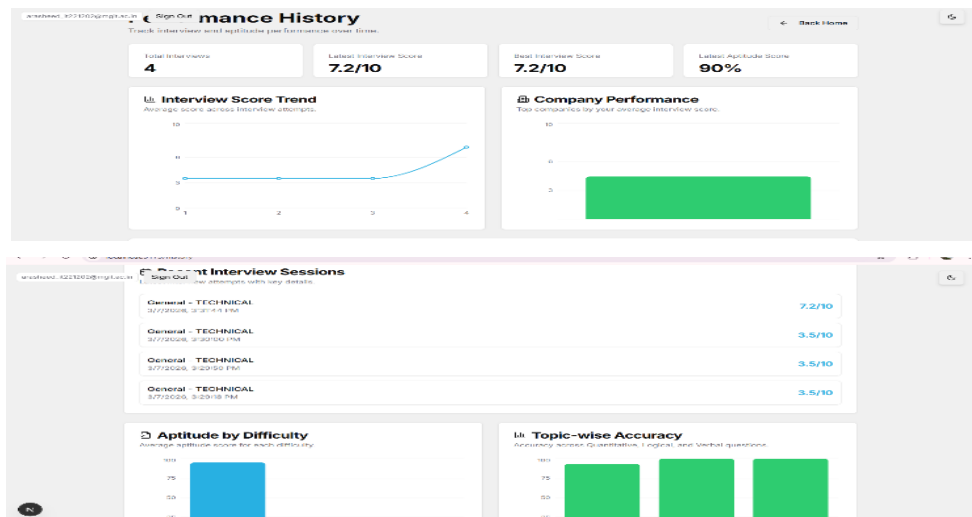


Fig. 10. Dashboard of performance history on interview trends and analytics.

This figure has the analytics dashboard showing the user performance across various sessions. It contains the trends in score, topic-wise accuracy, and history of the session. This assists the users keep track on their progress and where they are performing poorly.

6. CONCLUSION

The Multi-Mode Interview and Coding Practice Platform that is proposed in this paper is expected to enhance the efficiency, availability, and consistency of interview preparation systems. The system combines interview simulation, coding assessment, and aptitude testing in a single environment, which overcomes the usual drawbacks of traditional preparation techniques, including the absence of a structured feedback, the disjointed learning tools, and the inability to evaluate in real-time. Using the latest web technologies and cloud services, the suggested system will offer an efficient and scalable solution to improve the performance of candidates in a variety of skills areas.

React.js and Next.js were used to make the frontend interface, FastAPI as a backend processing, Firebase as authentication and cloud storage, and Judge0 API as a secure code executor. All these functionalities collaborate to provide a smooth user-platform interaction and secure data processing and generation of responses in real-time. Multiple modules make different modules such as practicing interview questions, solving coding problems, and attempting aptitude tests available on the same platform and without having to use many tools.

A systematic review system was used to improve the performance of the platform by determining how well users performed with different parameters including the quality of responses, accuracy, and interaction patterns. The system also stores performance history and analytics on cloud storage where users can monitor their progress as time goes by. It was a constant feedback mechanism that enables users to determine their strengths and weaknesses and in that way enhances their readiness to the interview.

The outcome of the experiment proved the system to be effective in all modules, allowing proper execution of coding, good evaluation and easy interaction with the user. The platform is able to process various user requests and sustain good performance and data integrity. Secure execution environment combined with cloud-based analytics, makes sure that the system can be reliable and scalable to real-world applications.

In general, the offered system proves that the multi-modal interaction, real-time evaluation and the cloud-based analytics may significantly enhance the process of interview preparation. The system can be improved in the future by adding more sophisticated AI-based assessment tools, individualized learning suggestions, voice recognition, and provision of scalability as it can be more responsive to the needs of various users and adapt to changing trends in technology.

REFERENCES

- [1] R. Patil, A. Butte, S. Temgire, V. Nanekar, and S. Gavhane, "Real Time Mock Interview using Deep Learning," International Journal of Engineering Research & Technology (IJERT), vol. 10, no. 05, pp. 392–394, 2021.
- [2] S. Madan et al., "Online Platform for Coding Exams and Interviews," International Journal of Scientific Research in Computer Science, 2024.
- [3] S. R. Jagtap, V. Kulkarni, Y. Pachorkar, O. Taur, S. Gupta, and U. Pujeri, "AI-Driven Real-Time Interview Simulation App with Voice Recognition and Facial Analysis," Indian Journal of Science and Technology, vol. 18, no. 25, pp. 2058–2066, 2025.
- [4] S. Surendra et al., "AI-Based Coding Interview Platform with Real-Time Evaluation," International Journal of Engineering Research, 2024.
- [5] S. Surendra et al., "Secure Online Coding Assessment System using Sandbox Execution," IEEE Conference on Computing Systems, 2023.
- [6] S. Khapekar, S. Bothara, T. Babar, and R. Kine, "AI Powered Mock Interview System with Real-Time Voice and Emotion Analysis," International Journal of Novel Research and Development (IJNRD), vol. 10, no. 2, pp. d178–d184, 2025.
- [7] S. Suguna, K. Sarankumar, S. Nithiskumar, R. Rethesh, and B. Kumar, "Virtual Self Practice Mock Interview Using Generative AI," International Journal of Current Science, vol. 14, no. 4, 2024.
- [8] R. Nithya and S. V., "Intelligent Job Interview Preparation and Career Advancement," International Journal of Computer Science (IJCS), vol. 13, no. 1, pp. 11–21, 2025.
- [9] P. Kothari, P. Mehta, S. Patil, and V. Hole, "InterviewEase: AI-Powered Interview Assistance," ResearchSquare, 2024.
- [10] G. R. Rao, B. C. Reddy, A. S. Kalyan, G. K. Priya, and K. Rajesh, "AI-Powered Mock Interview Preparation," International Journal for Modern Trends in Science and Technology, vol. 11, no. 04, pp. 65–70, 2025.
- [11] Wasik, S. Antezak, J. Badura, A. Laskowski, and T. Sternal, "A Survey on Online Judge Systems and Their Applications," ACM Computing Surveys, vol. 51, no. 1, pp. 1–34, 2019.
- [12] H. Z. Došilović and I. Mekterović, "Robust and Scalable Online Code Execution System," Journal of Systems and Software (Elsevier), vol. 164, 2020.