Himanshu Kumar, 2026, 14:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

# Design and Evaluation of an AI Based Adaptive Mock Interview System Using NLP and Real-Time Feedback Analysis

**Himanshu Kumar, Rohit Kumar, Aditya, Prof. Shivangi Patel, Prof. Nitin Pal**
Department of Computer Science and Engineering, Parul Institute of Technology (PIT), Vadodara, Gujarat, India.

**Abstract- Preparing for interviews is often inconsistent, as many candidates depend on repeated question lists and general advice that does not clearly show how to improve. This work presents an adaptive mock interview system that creates a more practical way to prepare by combining interaction, evaluation, and guidance in one place. The system is designed to behave like an interviewer by asking questions, examining answers, and giving feedback during the same session. To achieve this, the system processes user responses using language understanding methods and supports both typed and spoken input. Each answer is reviewed from multiple perspectives, including how well it addresses the question, how clearly it is expressed, and the overall tone of the response. Based on these observations, the system assigns a score and provides suggestions that users can apply immediately. A key feature of the system is its ability to adjust question difficulty during the session. When a user performs well, the system gradually increases the level of challenge, while weaker performance leads to simpler or more guided questions. This adjustment helps maintain balance and keeps the user engaged without making the session too easy or too difficult. The system was tested under controlled conditions to observe its behaviour across different types of responses. The results show stable performance, with timely feedback and consistent evaluation. Repeated interaction also leads to noticeable improvement, indicating that the system supports gradual learning and skill development. Overall, the proposed approach offers a structured and flexible way to practice interviews, reducing dependence on manual guidance while helping users build confidence through continuous feedback and adaptation.**

**Keywords: Artificial Intelligence, Mock Interview, Natural Language Processing, Adaptive Learning, Sentiment Analysis.**

## I. INTRODUCTION

Choosing the right candidate is important for any organization, but preparing for interviews is still not very effective for many people. Most candidates rely on common question lists or basic practice methods. In many cases, they do not receive proper feedback, or the feedback is too general to help them improve. Because of this, it becomes difficult to understand what went wrong and how to perform better in the next attempt.

In recent years, technology has started to play a bigger role in recruitment. Systems are now being developed that can examine candidate responses using computational methods instead of depending only on human evaluation. These systems look at how a person answers how clearly they express idea and how well their responses are structured. This makes the process more consistent and less dependent on individual judgment.

Another noticeable change is the shift from static preparation to interactive practice. Instead of answering fixed questions, candidates can now interact with systems that respond in real time. These systems can ask new questions based on previous answers, making the experience feel more like a real interview. This kind of interaction helps users learn faster because they can immediately see where they need improvement.

Even with these developments, many existing solutions are still limited. Some tools only check resumes, while others focus only on analysing tone or sentiment. A few provide mock interviews, but they often lack proper evaluation or detailed feedback.

Bringing all these features together in a single system is still a challenge. There are also concerns related to how accurately these systems evaluate performance and how well they adapt to different users.

To deal with these issues, this paper presents a mock interview system that adjusts itself based on user input. It generates relevant questions, analyse responses, and provides feedback during the session itself. The goal is to create a simple but effective environment where users can practice and improve continuously.

**The main contributions of this work include:**

- Designing an interview system that adapts to user responses
- Providing instant evaluation and feedback
- Supporting continuous interaction without interruption
- Studying system performance using user- based observations

The rest of the paper is organized into sections covering related work, system design, implementation, results, and conclusions.

## II. LITERATURE REVIEW

The hiring process has gradually shifted from being fully manual to partially automated, mainly due to the growing use of artificial intelligence. Organizations are now experimenting with systems that can evaluate candidates more consistently and at scale. These systems typically rely on a mix of language processing, predictive modeling, and pattern recognition to analyse how candidates communicate and present themselves.

Some research efforts have explored the idea of evaluating candidates through video-based interviews. In such systems, algorithms observe facial expressions, speaking style, and response patterns to form an overall assessment. This approach attempts to replicate human judgment, but in a more standardized way. While the results are promising, building such systems requires a large amount of training data and significant computing power, which makes them less accessible for smaller applications.

Another area that has received attention is understanding the tone and emotion in candidate responses. By examining the wording and structure of answers, machine learning models can estimate whether a response reflects confidence, hesitation, or uncertainty. This can add an extra layer of insight beyond simple right-or-wrong evaluation. However, these methods often work after the interaction is complete, meaning they do not actively guide the candidate during the interview itself.

Resume analysis is also widely used as an initial filtering step. Techniques from natural language processing allow systems to extract key details such as skills, experience, and achievements from unstructured text. In some cases, these systems go a step further by attempting to infer personality traits or work style tendencies. Although useful for shortlisting candidates, this approach cannot fully capture how a person performs in a real-time conversation.

More recent developments focus on interactive interview platforms that simulate real interview scenarios. These systems can generate questions dynamically, adjust difficulty levels, and provide feedback based on the candidate's answers. This creates a more engaging and practical experience compared to static questionnaires. Even so, evaluating the effectiveness of such systems remains a challenge, as there is no widely accepted standard for measuring performance.

Despite these advancements, several concerns continue to be discussed. One major issue is fairness. Automated systems may unintentionally favour certain patterns in data, leading to biased outcomes. Another concern is transparency, as candidates are often unaware of how decisions are being made. These challenges highlight the need for careful design and ongoing evaluation of AI-based recruitment tools.

In summary, existing work shows clear progress in automating different parts of the interview process.

However, most solutions focus on specific tasks rather than providing a complete, balanced system. This creates an opportunity for developing more integrated approaches that combine accuracy, fairness, and meaningful feedback in a single platform.

## III. METHODOLOGY

**System Overview**

In this work, a mock interview system is designed with the aim of helping users practice in a way that feels closer to a real interview. Instead of asking fixed questions, the system reacts to the user's answers and adjusts its behaviour during the session. The idea is not just to test the user, but to guide improvement while the interaction is happening.

The system focuses on three core activities: asking relevant questions, examining the user's response, and giving feedback immediately. Each response affects what happens next, so the session does not feel repetitive. Over time, this allows the system to better match the user's level and provide a more meaningful practice experience.

**System Design**

The system is divided into parts so that each task is handled separately but still connected.

**User Interface**

This is where the user interacts with the system. It shows the questions, accepts answers, and displays feedback. The user can respond either by typing or speaking.

**Server Side Logic**

This part controls how the interview runs. It decides when to ask the next question, stores user progress, and handles communication between different parts of the system.

**Processing Unit**

This is the main working part of the system. It generates questions, checks responses, assigns scores, and prepares feedback. All decision-making happens here.

**Data Storage**

User responses, scores, and session details are stored so that progress can be tracked across multiple attempts.

**Flow of Operation**

The process begins when the user starts a session. A question is generated, the user answers it, the system processes the answer, assigns a score, and then shows feedback. Based on the result, the next question is adjusted.

**Interview Process**

Each session follows a simple sequence:

- The user starts the session
- Basic details such as role or topic may be provided
- A question is presented
- The user gives a response (text or voice)
- Voice input is converted into text if needed
- The system studies the response
- A score is generated
- Feedback is displayed
- The next question is selected based on performance

This cycle continues until the session ends. Because each step depends on the previous one, the system gradually adapts to the user.

**Evaluation Method**

Instead of checking only one aspect, the system looks at multiple factors to judge a response. This gives a more balanced result.

The score is calculated using the following relation:

$$Score = 0.4C + 0.3K + 0.2F + 0.1S$$

**Here:**

- $C$stands for how correct and relevant the answer is
- $K$reflects how confidently the response is given
- $F$measures how smooth and clear the answer is
- $S$represents the tone of the response

This combination ensures that both knowledge and communication are considered.

**Processing Approach**

To understand user responses, the system performs a few basic steps. The text is first cleaned and broken into smaller parts. Important terms are identified, and the meaning of the response is compared with expected content. The system also checks the tone of the response to understand whether the user sounds confident or unsure.

For voice input, speech is first converted into text so that it can be analysed in the same way as typed answers. This allows the system to support both input modes without changing the evaluation process.

**Adaptive Behaviour**

A key part of this system is its ability to adjust itself during the session. If the user performs well, the system increases the difficulty level by asking more challenging questions. If the user struggles, it shifts to simpler or more guided questions.

This prevents the session from becoming frustrating or too easy, and keeps the user engaged throughout.

**Benefits of the Approach**

- Feedback is provided immediately after each response
- Questions change based on user performance
- The system can handle repeated use and larger data over time
- Evaluation considers multiple aspects instead of a single score
- Reduces dependence on manual judgment

## IV. IMPLEMENTATION

**Development Environment**

The system was built as a web application so that it can be accessed easily without installing any additional software. The overall setup combines a user-facing interface, server-side logic, and intelligent processing components that work together during an interview session.

On the frontend side, a component-based framework is used to design the interface. This allows smooth interaction, quick updates on the screen, and a responsive layout for different devices. The backend is responsible for handling requests, maintaining session flow, and connecting all parts of the system. It also ensures that user data is processed securely and efficiently.

For storing information, a database is used to keep records such as user details, responses, and performance data. This helps in tracking progress across multiple sessions. On the intelligence side, pre- trained language models and text-processing libraries are used to understand and evaluate responses. For users who prefer speaking instead of typing, a speech- to-text service is integrated so that voice input can be converted into text before analysis.

The overall design follows a modular approach. Each part of the system works independently but remains connected through defined interfaces. This makes it easier to update or expand the system without affecting other components, and also supports handling multiple users at the same time.

**Functional Modules**

The implementation is organized into multiple functional modules, each handling a specific part of the interview process. This separation ensures clarity in design and improves system efficiency.

The User Module manages all interactions related to user access and session control. It allows users to register, log in, and initiate interview sessions. It also provides an option to input relevant details such as job role or resume information, which helps tailor the interview experience.

The Question Generation Module is responsible for producing interview questions dynamically. Instead of relying on predefined sets, it generates context-aware questions based on user input and previous responses. This creates a more flexible and realistic interview environment.

The Response Processing Module handles incoming answers from the user. If the response is given in voice format, it is first converted into text using

speech recognition. The text is then cleaned and structured so that it can be analysed effectively by the system.

The Evaluation Module examines the processed response to determine its quality. It considers aspects such as relevance, clarity, and expression. In addition, it evaluates the tone of the response to estimate confidence levels. Based on these factors, the system assigns a score that reflects overall performance.

Finally, the Feedback Module generates meaningful feedback immediately after evaluation. It highlights strengths and identifies areas where improvement is needed. At the end of the session, it can also present a summary to give users a complete view of their performance.

## V. SYSTEM ARCHITECTURE

The architecture of the proposed system is designed to ensure scalability and efficient processing. It consists of the following components:

- **Frontend Layer:** Provides the user interface for interaction
- **Backend Layer**: Handles request processing and system logic
- **AI Processing Layer:** Performs NLP analysis and scoring
- **Database Layer:** Stores user data and performance metrics

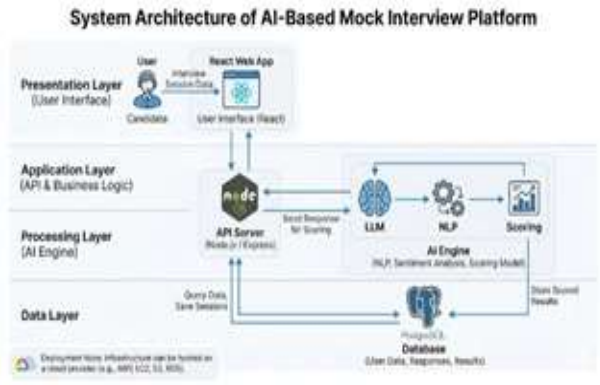


Figure 1: System Architecture of AI Mock Interview Platform

## VI. RESULTS AND ANALYSIS

**Experimental Setup**

The system was tested in a controlled environment rather than a real deployment. Different test scenarios were created to represent how a typical user would interact with the platform during a mock interview. These scenarios included variations in response length, clarity, and input type (text and voice).

The purpose of this setup was to observe how the system behaves under normal usage conditions. Special attention was given to how quickly the system processes answers, how reliable the feedback is, and whether the interaction feels smooth from the user's perspective.

**Performance Metrics**

A small set of measurable factors was used to evaluate the system. These factors were chosen to reflect both technical performance and user experience.

The first factor is evaluation accuracy, which indicates how well the system judges the quality of responses. The second is feedback time, which measures how quickly the system responds after receiving an answer. Another important factor is system delay, which shows how much time is taken between input and output. Finally, improvement over repeated sessions was observed to understand whether the system actually helps users perform better over time.

**Results**

The results collected from the test scenarios are shown below:

Table 1: Summary of System Performance Metrics

| Metric | Value |
|---|---|
| Evaluation Accuracy | 85% |
| Average Feedback Time | 1.8 sec |
| System Delay | 2.5 sec |
| Performance Improvement | 28% |

Table 2: Response Evaluation Criteria and Feedback Classification

| Response Type | Score Range | Feedback Quality |
|---|---|---|
| Strong Answer | 80–100 | Detailed & positive |
| Moderate Answer | 50–79 | Suggestive improvement |
| Weak Answer | <50 | Corrective guidance |

**Analysis**

From the observed results, it can be seen that the system performs at a stable level during simulated interviews. The accuracy value shows that the system is able to judge responses in a fairly consistent way. While it may not fully match human evaluation, it provides a reasonable estimate that can help users improve.

The feedback time is short enough to maintain a smooth flow during interaction. Users do not have to wait long after answering, which keeps the experience engaging. The overall delay of the system also remains within a comfortable range, so the interaction does not feel interrupted.

One of the more useful outcomes is the improvement seen after multiple sessions. As users continue to practice, their performance shows noticeable progress. This suggests that the system is not only evaluating responses but also helping users learn from their mistakes.

**Discussion**

The system shows that it is possible to combine different techniques into a single platform that supports interview practice in an effective way. Compared to basic preparation methods, it provides faster responses, adjusts to user performance, and encourages continuous improvement.

At the same time, there are some limitations. The system depends on how well it understands language, so unclear or unusual responses may affect evaluation. Voice input can also introduce errors if speech recognition is not accurate. In addition, since testing was done in a controlled setup, real-world performance may vary.

Future work can focus on testing with actual users to get more realistic results. Improving the accuracy of response analysis and adding more advanced features could further enhance the system. These changes would help make the platform more reliable and practical for everyday use.

## VII. PROJECT SCREENSHOTS

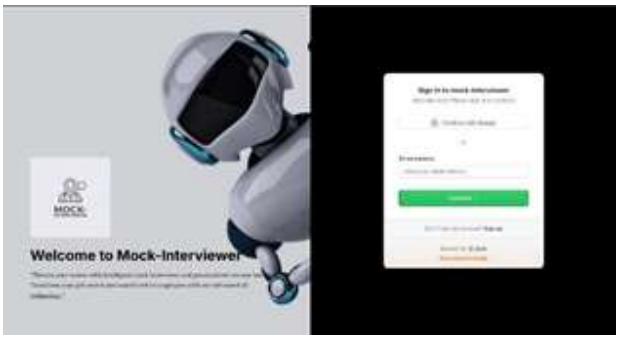


Image 1: Log in page of AI Mock Interview Platform

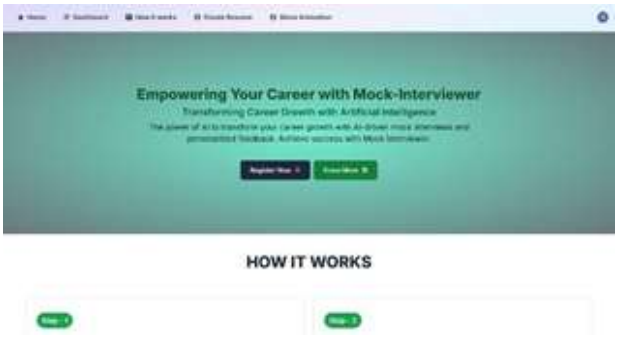


Image 2: landing page of AI Mock Interview Platform

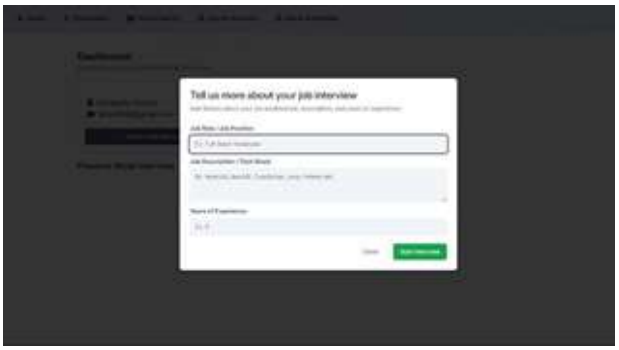

Image 3: Dashboard of AI Mock Interview Platform

## VIII. CONCLUSION

This study set out to improve the way interview practice is done by introducing a system that can interact with users and guide them while they learn. Traditional preparation methods often leave gaps, especially when it comes to timely feedback and understanding personal weaknesses. The system developed in this work tries to reduce those gaps by creating a more responsive and user-focused practice environment.

The approach taken combines multiple capabilities into one place. It can ask questions, study answers, and respond with feedback during the same session. Since the questions are not fixed, the experience changes based on how the user performs. This makes the process feel less repetitive and more useful for gradual improvement. The ability to respond through both text and voice also adds flexibility for different users.

During testing, the system showed stable behaviour in handling responses and providing feedback without long delays. Over repeated use, users were able to perform better, which suggests that the system supports learning in a practical way. Instead of only pointing out mistakes, it helps users recognize patterns in their answers and improve step by step.

In summary, this work shows that interview preparation can be made more effective when the system adapts to the user and provides immediate guidance. While there is still room for improvement, especially with more advanced evaluation and real user testing, the current system forms a solid base for building a smarter and more helpful interview practice tool in the future.

**Acknowledgements**

We would like to express their sincere gratitude to the faculty members and mentors of the Department of Computer Science and Technology at Parul University for their continuous guidance and support throughout the development of this project. Their valuable insights and constructive feedback played a crucial role in shaping the direction and quality of this research work.

we also acknowledge the support provided by the institution for offering the necessary resources, infrastructure, and academic environment required to successfully complete this study. Special thanks are extended to peers and colleagues who contributed through discussions, suggestions, and encouragement during various stages of the project. Furthermore, we would like to recognize the contributions of open-source tools, libraries, and research resources that facilitated the implementation and development of the proposed system. Without these technological resources, the realization of this work would not have been possible.

Finally, we express our appreciation to all individuals who directly or indirectly contributed to the completion of this research work.

## REFERENCES

1. B. C. Lee and B. Y. Kim, “Development of an AI-Based Interview System for Remote Hiring,” International Journal of Advanced Research in Engineering and Technology, vol. 12, no. 3, pp. 654–663, 2021.
2. S. S. Alduayj and P. Smith, “Sentiment Classification and Prediction of Job Interview Performance,” 2019.
3. V. Patil, P. K. Venkiteswaran, S. Brid, O. Shivarkar, and S. Noor, “AI-Driven Recruitment through Comprehensive CV Analysis and Personality Prediction,” 2023.
4. W. A. Albassam, “The Power of Artificial Intelligence in Recruitment: An Analytical Review of Current AI-Based Recruitment Strategies,” International Journal of Professional Business Review, vol. 8, no. 6, 2023.
5. P. Nikhitha, C. N. D. Sai, and R. N. S. Naidu, “Intelligent AI Mock Interview Platform: An Adaptive Training System,” Research Digest on Engineering Management and Social Innovations, vol. 2, no. 4, 2026.
6. J. M. Bershika and G. Nancy, “AI-Based Virtual Interviewer System Using NLP and Emotion

Detection," Lecture Notes in Networks and Systems, Springer, 2026.
7. I. Ajunwa, "An Auditing Imperative for Automated Hiring Systems," Harvard Journal of Law & Technology, vol. 34, no. 2, 2021.
8. E. Bauer, M. Greisel, I. Kuznetsov, M. Berndt, I. Kollar, M. Dresel, M. R. Fischer, and F. Fischer, "Using Natural Language Processing to Support Peer-Feedback in the Age of Artificial Intelligence," British Journal of Educational Technology, vol. 54, pp. 1222–1245, 2023.
9. Z. E. Imel, B. T. Pace, C. S. Soma, M. Tanana, T. Hirsch, J. Gibson, P. Georgiou, S. Narayanan, and D. C. Atkins, "Design Feasibility of an Automated Machine Learning-Based Feedback System for Motivational Interviewing," Psychotherapy, 2019.
10. S. Ahmad, "The Role of Artificial Intelligence in Reducing Implicit Bias in Recruitment: A Systematic Review," International Journal of Advanced Research in Education and Technology, vol. 11, no. 6, 2024.