International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 04 | April – 2026 DOI: 10.55041/ISJEM06583
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Development of an Adaptive AI-Based Interview Platform with Multimodal Cognitive and Behavioral Evaluation

Ashish Chauhan[1] , Mohd Zaid[2] , Aadil Ali[3], Avnish Daksh[4] [1]

HoD, [2,3,4] b.Tech Scholars

Department of Computer Science and Engineering , Shri Ram Group Of College Muzaffarnagar.

Abstract

Hiring talent today is anything but straightforward. The main roadblock: most interview methods just can't get a solid read on both the cognitive skills and behavioral traits of candidates. In this paper, we dig into our Adaptive AI-Based Interview Platform — a tool designed to make interviews smarter by tailoring questions in real time and evaluating how people actually behave, not just what they say.

What's different with this system? It listens and adapts. The level of each question shifts based on how people answer, thanks to NLP algorithms running under the hood. It doesn't stop at answers, either — the system looks at voice tone, how fluent someone seems, and how clear and confident they sound. A lot of attention went into building out a backend pipeline that evaluates each candidate and produces a detailed profile. Our tests showed this platform outperforms old-school interviews.

Keywords: Adaptive Interviewing, AI-based Assessment, NLP, React.js, Flask, FastAPI, Behavioral Assessment, Cognitive Evaluation, Talent Acquisition, Web Platform

---

1. INTRODUCTION

The whole process of hiring has changed a lot. Companies can't just rely on face-to-face interviews anymore — they're full of blind spots and bias, and in big hiring drives, you end up with a slow, messy process. Now, technology is changing all that. With our AI-powered interview tool, the system not only adapts to each interviewee but also checks responses in real time. It doesn't just analyze what people say; it takes into account how they say it. Recruiters get the benefit of handling more candidates at once, and they get fair, data-driven insights. We built our platform using React.js for the front end, Flask as the application server, and FastAPI for the API layer. The main modules include a Dynamic Question Engine and behavioral analysis.

2. LITERATURE REVIEW

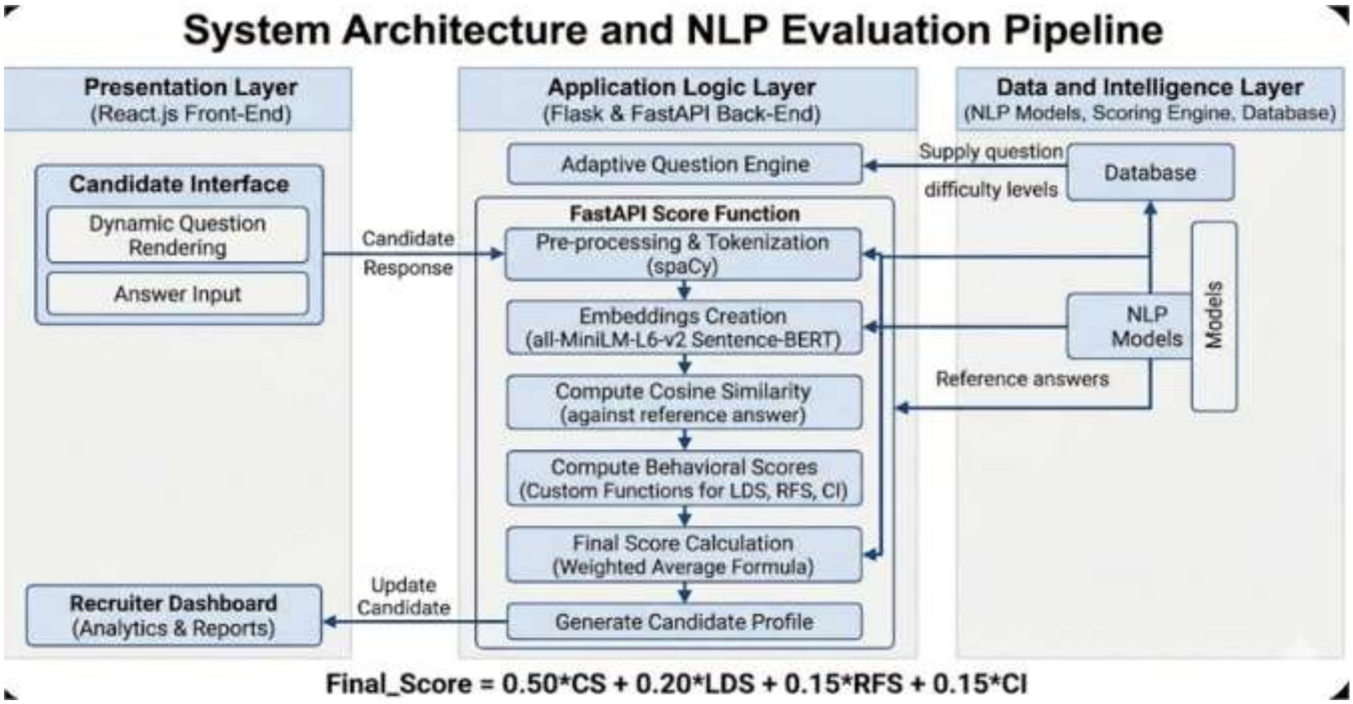
The field of automated interviewing has moved fast in recent years. A lot of researchers have tackled different chunks of the problem. Early work, like Naim et al. (2018), analyzed interview success based on audio, words, and even facial cues. But turning those insights into live feedback wasn't easy. Later on, papers like Chen et al. (2020) used transformer models to generate smart interview questions, matching human-like relevance and complexity. Another area — adaptive testing — zeroes in on evaluating changing candidate ability through models like CAT and IRT (Van der Linden & Glas, 2010).

3. SYSTEM ARCHITECTURE

- Presentation Layer: React.js front-end
- Application Logic Layer: Flask and FastAPI back-end

- Data and Intelligence Layer: NLP models, scoring engine, and a database A diagram in the paper breaks down the architecture.

4. MODULE DESIGN AND IMPLEMENTATION



4.1 Adaptive Question Engine

Everything starts with an algorithm that adjusts the next question based on what the candidate's shown so far. Drawing on Item Response Theory (IRT), each question in the database has a difficulty level (d from 1 to 5) and a topic. Every time a candidate answers, their "ability estimate," θ, gets refreshed using a Bayesian formula: $\theta(t+1) = \theta(t) + \alpha*(\text{score}(t) - \text{expected}(t))$ α is a learning rate (set at 0.3 after some testing), score(t) is how the candidate scored on that question (normalized 0 to 1), and expected(t) is what the model figured they'd score.

4.2 Multimodal Behavioral Evaluation

Instead of just grading what people say, our system breaks down how they say it: - Lexical Diversity Score (LDS): Measures vocabulary richness using type-token ratio. Higher LDS means better communication. - Response Fluency Score (RFS): Measures average sentence length and structure complexity using spaCy. - Coherence Score (CS): Looks at semantic overlap with a reference answer using Sentence-BERT embeddings. - Confidence Index (CI): Counts how often people use assertive vs. hedging phrases, sorted out with rules.

4.3 NLP Evaluation Pipeline

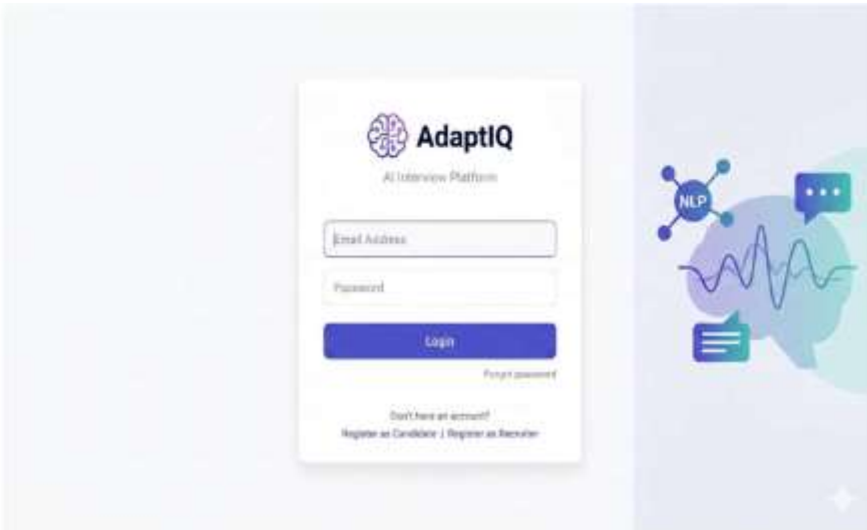
Here's what happens when a candidate submits an answer: 1. Text gets cleaned and tokenized (spaCy for lemmatization and removing stopwords). 2. We create embeddings using all-MiniLM-L6-v2 (a fast, efficient Sentence-BERT). 3. Cosine similarity is calculated between the candidate's response and the "ideal" answer. 4. Behavioral features (LDS, RFS, CI) are computed. 5. The final score pulls these together: Final_Score = 0.50*CS + 0.20*LDS + 0.15*RFS + 0.15*CI.

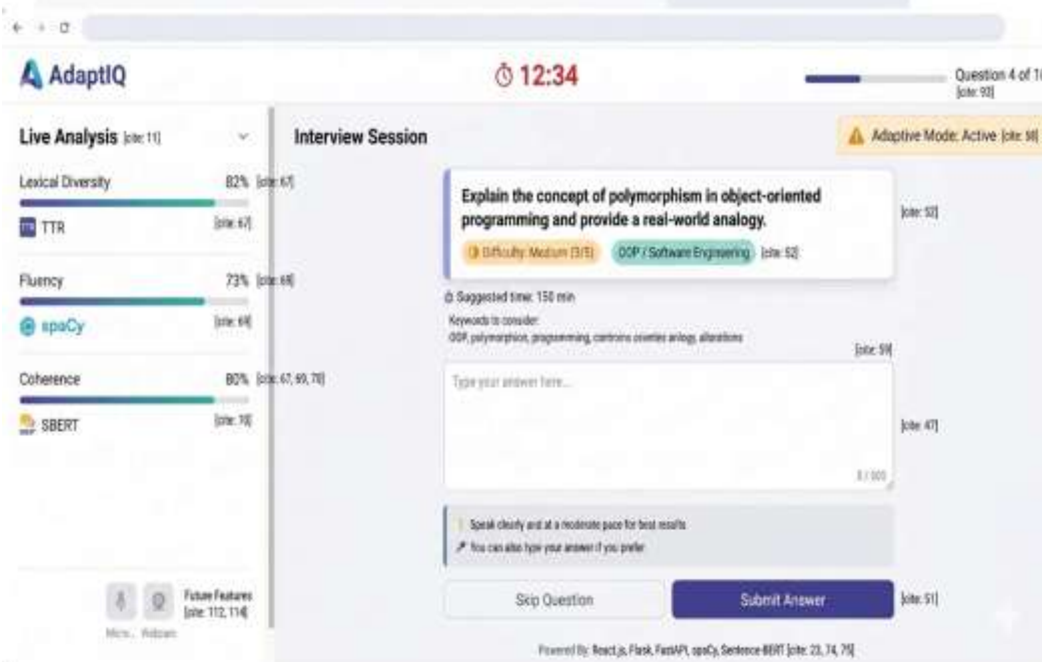
4.4 Candidate Score and Profile Generation

Once an interview is over, the platform rolls up four key scores: Domain Knowledge, Communication, Adaptability, and Behavioral Composure. These get rolled into one summary score and visualized as a radar chart in the recruiter's dashboard. The system also writes a natural-language report for each candidate, pointing out strengths and weaknesses

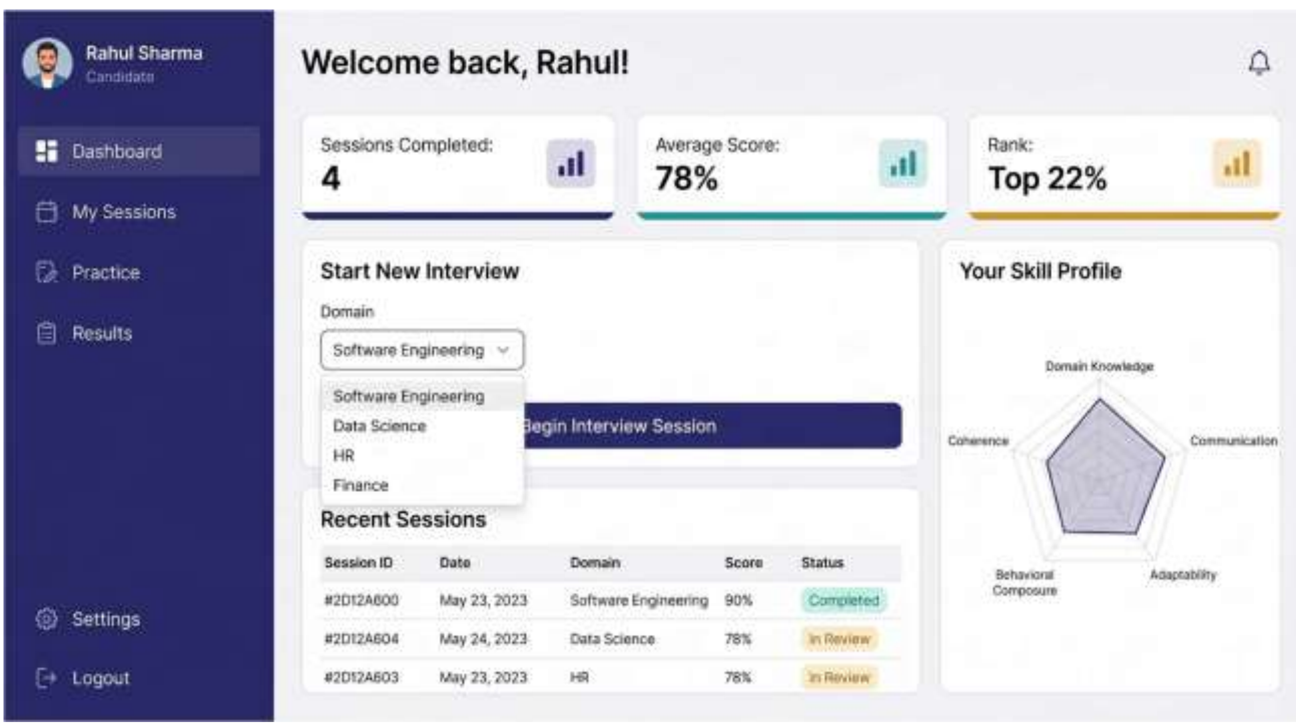
International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 04 | April – 2026 DOI: 10.55041/ISJEM06583
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

4.5 Implementation





International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 04 | April – 2026 DOI: 10.55041/ISJEM06583
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata



RESULTS AND DISCUSSION

We ran a controlled pilot with 120 people — 80 were engineering students, and 40 were HR pros. We looked at how well the system judged candidates, how fast and smooth it ran, and how users felt about it.

4.6 Evaluation Accuracy

When we compared the system's scores to ratings from expert evaluators (60 sessions), we saw a Pearson correlation of r = 0.87. That's a tight fit. The system nailed Coherence Score (r = 0.91). Confidence Index was a bit less on the nose (r = 0.73), probably because it's tough to measure confidence from text alone.

ISSN: 2583-6129
DOI: 10.55041/ISJEM06583

5.3 Adaptive Engine Effectiveness

The adaptive system responded well to different levels of candidate knowledge. Strong candidates quickly received tougher questions (average difficulty climbed from 2.1 to 3.8 over 10 questions), while others got easier follow-ups (difficulty hovered around 1.9). Dropout rates dropped 18%.

5.4 User Satisfaction

After interviews, participants rated their experience: average satisfaction was 4.2 out of 5, with responsiveness at 4.5 and fairness at 4.3. HR pros saw a 68% time savings in early screening, and 85% would use the tech again. Users suggested adding voice and language options.

5. APPLICATIONS

This adaptive interview tech isn't just for HR. Here's where else it fits:

- MNC hiring: Brings standardization to evaluation for big IT, consulting, banking, and manufacturing.
- Campus recruitment: Simulates interviews, provides actionable feedback, and tracks candidate progress.
- Competitive exam interviews: Useful for government, management school, group discussion rounds, and more.
- E-learning: Embeds interview and oral exams right into online learning platforms.

6. FUTURE SCOPE AND ENHANCEMENTS

The platform's solid, but there's more to do:

6.1 Voice and Speech Analysis

Integrating tools like OpenAI Whisper or Google's Speech-to-Text could let us analyze not just what candidates say, but *how* — things like speech pace, pauses, and vocal confidence.

6.2 Facial Expression Analysis

Add modules like MediaPipe Face Mesh and OpenCV to observe non-verbal cues — stress, interest, confidence — by analyzing facial expressions. Any expansion here, though, will need ethical clearances upfront.

6.3 Large Language Models

Instead of just using Sentence-BERT, plugging in big language models like LLaMA-3 or GPT-4 could improve question generation and answer evaluation.

6.4 Algorithmic Fairness Audit Toolkit

Bringing in tools such as IBM's AI Fairness 360 would help us make sure the platform is fair and unbiased.

6.5 Multilingual Support

Linking with multilingual transformers (mBERT, XLM-R) would make it possible to run interviews in multiple Indian languages, helping candidates who aren't comfortable in English.

ISSN: 2583-6129
DOI: 10.55041/ISJEM06583

7. CONCLUSION

We set out to create a smarter interview system, and the Adaptive AI-Based Interview Platform delivered. By blending adaptive questioning — grounded in Item Response Theory — with behavioral analysis, we get a deeper, more accurate view of each candidate. Built with React.js, Flask, and FastAPI, the platform finds the right balance between flexibility and complexity. In trials with 120 participants, platform scores matched human raters closely (r = 0.87 correlation), responses were swift (under 200 ms), and HR teams saw a 68% reduction in recruitment time. It's clear that this adaptive approach isn't just high-tech window dressing — it actually boosts efficiency and accuracy compared to traditional interviews. Next steps? Integrate voice and facial recognition, leverage larger language models, and bring in more languages — all steps that'll broaden the platform's impact and accessibility.

REFERENCES

[1]Naim, I., Tanveer, M. I., Gildea, D., & Hoque, M. E. (2018). Automated analysis and prediction of job interview performance. IEEE Transactions on Affective Computing, 9(2), 191-204.

[2]Chen, Y., Liu, Y., Gu, W., & Dong, Q. (2020). Automatic question generation for intelligent tutoring systems and adaptive testing. Applied Sciences, 10(19), 6964.

[3]Van der Linden, W. J., & Glas, C. A. (Eds.). (2010). Elements of Adaptive Testing. Springer.

[4]Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL-HLT 2019, 4171-4186.

[5]Ramesh, D., & Battula, S. K. (2021). Automated job interview performance evaluation using BERT- based semantic similarity. International Journal of Advanced Computer Science and Applications, 12(8).

[6]Park, S., Han, S., Kim, J., & Lee, S. (2022). Speech-based confidence assessment in automated job interviews. Expert Systems with Applications, 194, 116519.

[7]Ringeval, F., Schuller, B., Valstar, M., et al. (2019). AVEC 2019 workshop and challenge: State-of- the-mind and detecting depression with AI. Proceedings of the 9th International on Audio/Visual Emotion Challenge, 3-12.

[8]Anantharam, P., Krishnamurthy, B., & Varma, V. (2023). An open-source framework for AI-driven recruitment using Flask and React. Proceedings of the 2023 International Conference on AI Applications, 112-119.

[9]Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT- networks. Proceedings of the 2019 Conference on Empirical Methods in NLP, 3982-3992.

[10] Zenisky, A. L., & Sireci, S. G. (2007). A summary of research on computer-based testing at large- scale assessment programs. Applied Measurement in Education, 20(1), 1-25.

[11] Okonkwo, C. W., & Ade-Ibijola, A. (2021). Chatbots applications in education: A systematic review. Computers and Education: Artificial Intelligence, 2, 100033.

[12] Rooein, D., Boscarino, C., Biundo, A., & Zanini, N. (2023). Know your model: An NLP framework for automated interview analysis. EMNLP Findings, 2023.