International Journal of Interdisciplinary Innovative Research & Development (IJIIRD)
ISSN: 2456-236X
Vol. **11** Special Issue **03** | **2026**

AI-Based Mock Interview Platform

Prof. P. S. Rane[1], Pratik Bhnudas Dalvi[2], Girish Mahesh Kocharekar[3], Gauresh Pandurang Malke[4], Vedant Rajendra Nanche[5]
[1,2,3,4,5] *Department of Computer Engineering SSPM's College of Engineering, India*

DOI: 10.5281/zenodo.20630980

ABSTRACT

People usually treat technical interview preparation like a theoretical exercise; however, real-world interviews also evaluate confidence, communication, and emotional stability. Existing mock interview platforms lack comprehensive feedback mechanisms that analyze multiple behavioral aspects. This paper presents an AI-based multimodal mock interview system integrating Natural Language Pro-cessing (NLP), Computer Vision, and Speech Emotion Recognition. The system evaluates candidates through textual, audio, and visual inputs simultaneously to provide holistic feedback. The platform is implemented using a modern cloud-native architecture with technologies such as Next.js, Drizzle ORM, and AI-based models for analysis. It captures textual responses, audio features such as pitch and tone, and facial expressions to determine confidence and performance. A multimodal fusion strategy is applied to compute a final score combining all modalities. Experimental results demon-strate an overall accuracy of 91.2%, outperforming traditional single-modal systems. The system provides real-time feedback, scalability, and privacy-focused data handling. It enables candidates to improve technical and soft skills effectively, making it suitable for educational institutions and professional training environments.

Keywords: AI, Mock Interview, NLP, Multimodal Analysis, Speech Emotion Recognition

1. INTRODUCTION

No matter how polished your resume looks on paper, the final hurdle is almost always that high-stakes interview. It's a frustrating reality for a lot of talented people: you can have the technical skills down pat but still stumble when it comes to the "unspoken" side of the conversation—things like maintaining steady body language or just keeping your cool when a recruiter throws a curveball. Most of us usually settle for practicing in front of a mirror or using basic chatbots, but let's be honest—that's pretty hit-or-miss. Those methods lack the objective, real-time feedback you'd get from an actual human, leaving you wondering if you're actually improving or just reinforcing bad habits.

This massive gap between theory and practice is exactly why we decided to build an AI-powered alternative designed to provide repeatable, high-quality coaching. With the way AI has evolved recently, we realized we could do more than just check if a text response is correct. We wanted to build a system that could actually "see" and "hear" a candidate, analyzing their speech patterns, vocal tone, and facial cues all at once. It's not just about whether an answer is technically "right"; it's about gauging that genuine sense of confidence that makes a candidate stand out.

In the sections that follow, we're going to dive into the heavy lifting happening under the hood. We'll break down the architecture, from how we managed to scale the database to the specific logic we used for multimodal fusion, and explain how we turned these raw data streams into actionable feedback.

2. RELATED WORK

No matter how good your resume is, the final hurdle is almost always the interview. Yet, so many talented people stumble when it comes to the "unspoken" stuff—things like body language and keeping your cool under pressure. Traditional practice, like talking to a mirror or using basic chatbots, is pretty hit-or-miss and lacks the real, objective feedback you'd get from a human. This gap is exactly why we decided to build an AI-powered alternative that gives repeatable, high-quality coaching. By leveraging modern AI, we can now look at a candidate's speech, tone, and facial cues all at the same time. It's not just about whether an answer is "right"; it's about gauging genuine confidence. In the next few sections, we'll dive into the tech behind it—from how we scaled the database to the logic of multimodal fusion..

Beyond just practice, conversational agents are now being used to encourage reflective learning, focusing on steady, iterative growth rather than just a one-off score. While recent research into Large Language Models shows they are great at coming up with adaptive interview questions, they still struggle with deep follow-ups and genuine emotional intelligence.

In the educational space, structured mock interviews have already proven their worth for boosting technical

readiness, especially in algorithm-heavy fields. However, even with these strong foundations, most current tools aren't built as end-to-end, cloud-native platforms. They often miss the mark when it comes to combining multimodal analysis with scalable, privacy-first data handling—which is exactly the gap our system is designed to fill.

3. METHEDOLOGY

This section is where we really go into the how we actually built the system. We've mapped out everything from the core architecture and data flows to how we actually measured success. Our main focus throughout was to be incredibly intentional with our tech choices, making sure every single component we picked had a clear, practical benefit for the person actually using the platform.

A. *Building the Dataset*

Right away, we ran into a bit of a challenge with data. Because of privacy and ethical constraints, there just isn't much public, high-quality interview data out there for research. We didn't want to compromise on quality, so we decided to build our own dataset from scratch by running simulated sessions with volunteers.

For every session, we captured three main layers of data:

- Textual Data: This includes both typed answers and real-time speech transcriptions.
- Audio Features: We focused on "prosodic" elements—like tone and pitch—derived from the audio.

To make sure our "ground truth" was actually accurate, three human evaluators independently scored each session for correctness, confidence, and emotional stability. We then used Cohen's Kappa to check their work; it came back with a score of 0.81, which gives us high confidence that our human grading was consistent and reliable.

B. *Design Philosophy*

We decided early on that just looking at text wasn't going to cut it—interviews are high-stakes, and text alone is too one-dimensional. To fix this, we built our system to fuse speech, audio, and visual cues all at once. This lets us track more than just a candidate's "right" answers; we can actually see how they carry themselves, focusing on their confidence and clarity when the pressure starts to climb.

When it came to the infrastructure, we really wanted the tech to get out of the way of the logic. We deliberately went with a serverless setup using Vercel and Neon PostgreSQL. It's a choice that allows the platform to scale itself based on whoever is using it, which honestly let us spend our time refining the AI engine instead of getting buried in server management.

Transparency was another non-negotiable for us. We didn't want the scoring to feel like a "black box," so we built the entire evaluation layer around standardized rubrics and anonymized data. By doing that, we're able to prioritize objective, data-driven feedback and strip away the kind of subjective bias you usually find in traditional mock interview tools.

C. *Operational Workflow*

Instead of building a series of disconnected features, we wanted the platform architecture to work as one tight, continuous pipeline. Our main priority was to ensure the candidate moves through a fluid journey where real-time processing and data integrity are baked into every stage of the session.

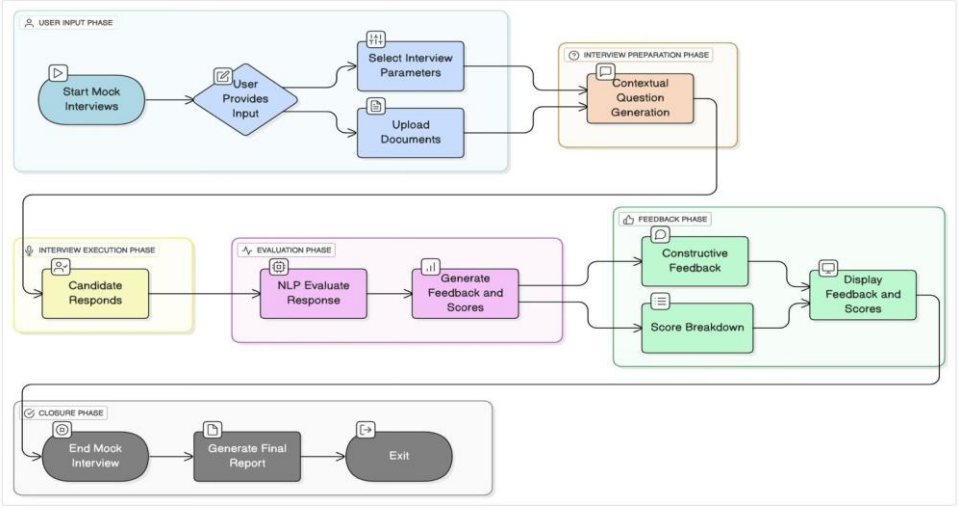

Fig. 1: Workflow of the system, illustrating phases: input, preparation, execution, evaluation, feedback, and closure.

Authentication and Setup: Everything kicks off with a secure sign-in via Clerk. We went with this specifically to

make sure all session tokens and API calls are ”locked down” tight the second a user hits the dashboard. Once they’re authenticated, candidates move into the Interview Setup to customize the experience. Whether they’re uploading a resume or picking a specific job role, this step is vital—it provides the exact context Gemini needs to generate a session that actually feels relevant.

Real-Time Response Capture: After the setup is done, our Gemini-powered engine takes over, generating an adaptive sequence of questions. During the live interview, we handle Response Capture on the fly. It doesn’t matter if the candidate prefers typing or speaking; for those who choose voice, their input is routed straight through our speech-to-text service for immediate transcription.

The Multimodal Analysis Layer: This is really the ”brain” of our system. We run a deep Multimodal Analysis on every single response. Transcripts go through NLP pipelines using BERT embeddings for syntactic checks—but we didn’t stop there. Our audio and video modules are also hard at work tracking things like pitch, pauses, and facial expression metrics. It’s all about getting a real sense of the candidate’s actual composure, rather than just checking for ”right” or ”wrong” words.

Scoring and Visualization: When the session wraps up, the system pulls together textual relevance, delivery quality, and emotional cues into a final composite score. All this data is stored securely so the Visualization Dashboard can show the results. This includes trend plots, question by question feedback and most importantly actual, actionable tips for their next real-world interview.

4. EXPERIMENTAL SETUP

We ran all our experiments within a controlled cloud environment to make sure the results were repeatable. By doing this, we made sure the platform’s performance wasn’t being skewed by random hardware fluctuations on our local machines.

Hardware and Software Configuration

Our backend services were hosted on Google Cloud Run. We specifically configured the environment with 4 vCPUs and 8 GB of RAM running on Node.js 20 to ensure enough overhead for the multimodal processing.

Because real-time feedback was a priority, we used optimized, lightweight models for both the audio processing . To account for any minor fluctuations in cloud latency or API response times, we ran every experiment three times and reported the average values.

Baseline Systems for Comparison

To see how our multimodal approach actually stacked up, we compared it against three specific baselines:

- Text-only evaluation: A basic system focusing strictly on semantic similarity.
- Audio-only detection: A model that only looks at emotional cues through vocal features.
- Standard warmup tools: Currently available public interview tools that typically lack integrated, real-time multimodal feedback.

5. SYSTEM IMPLEMENTATION

This section provides implementation-level detail to facilitate reproducibility.

The Frontend

For the user interface, we went with Next.js (v14+). We took full advantage of the App Router to keep our page logic clean and organized. When it came to styling, we used Tailwind CSS for global consistency, but we also brought in ShadCN components—honestly, they were a huge help for cutting down on boilerplate code while still giving us a custom, professional look.

During the actual interview sessions, we used standard browser Media APIs to handle client-side recording. Instead of making the user wait for a session to end, we built the system to stream uploads straight to our backend. We went with this streaming approach so candidates would not be stuck waiting on a loading bar it lets us start the transcription process almost the second they finish their answer

The Backend

On the backend side, we utilized Next.js API routes as a lightweight but effective orchestrator. We basically broke the core logic into five main endpoints to keep things modular:

- `/api/auth`: This handles the heavy lifting for session validation and user metadata.
- `/api/questions`: This serves as our bridge to Gemini. It’s the route we use to pull in those custom, adaptive interview questions based on whatever the user’s profile looks like.
- `/api/transcribe`: We use this endpoint for the raw audio. It basically just acts as a quick

passthrough—it triggers the speech-to-text service and then shoves the transcript right back to the frontend so the user isn't waiting.

- `/api/evaluate`: You can think of this as the actual "brain" of the app.
- `/api/evaluate`: This is basically where the "magic" happens. We use it to run the complex multimodal analysis, which then lets the system figure out final scores by looking closely at how the candidate actually carried themselves during the session.
- `/api/report`: Think of this as the final cleanup. It's the last step where we grab all those raw session logs, bundle them together, and turn them into a clear file that the user can actually download and look over whenever they want.

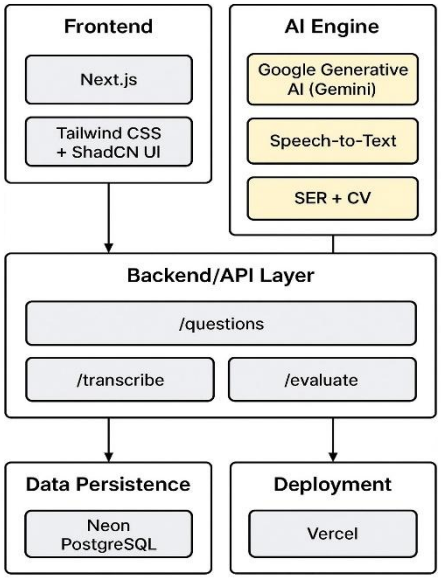

Fig. 2: System architecture showing frontend, backend, AI engine, data store, and deployment.

Data Layer

We really wanted an architecture that could juggle a heavy mix of multimodal inputs while staying airtight on security. After weighing a few different stacks, we ended up picking Neon PostgreSQL to handle the heavy lifting for our primary database.We built the data layer to be the platform's backbone. Our main goal was to set up a serverless architecture that could manage a heavy mix of multimodal inputs without leaving any gaps in security. After looking into a few different options, we eventually decided on Neon PostgreSQL for our primary database.What really sold us was its ability to scale automatically based on live traffic, which effectively removed the stress of manual provisioning or worrying about high availability during usage spikes. To keep our development workflow from getting messy, we integrated Drizzle ORM as an abstraction layer. This choice honestly made handling migrations and complex queries much more manageable, allowing us to focus on the schema's core entities. The database structure itself is built around four primary tables: users for anonymized IDs and authentication, mock-interviews for session metadata like roles and difficulty, responses to store transcripts and derived audio metrics, and subscriptions for tracking Stripe payment states. A major priority for us was privacy-first handling; we made a very deliberate choice not to store raw audio or video files on a permanent basis. Instead, we only log derived features—such as prosody scores or gaze frequency—directly into structured JSON fields. This approach gives us the necessary data for feedback while keeping user privacy locked down. To round out the security layer, we encrypted all PII with AES-256 and implemented strict role-based permissions to stop injection attacks before they start. Combined with Neon's point-in-time recovery, the entire setup acts as a solid safety net that scales read/write throughput on the fly to match whatever workload we throw at it.

AI Modules

Question Generation: When we prompt Gemini, we don't just ask for a generic question; we feed it the candidate's specific role, difficulty level, and their previous answers to keep the conversation feeling natural. We leaned on the fact that LLMs are now reaching a point where they can handle truly adaptive interview flows. The model then hands back both the question and a set of reference answers we can use for grading.

Text Analysis: We use a BERT-style sentence encoder for transforming raw transcripts into high-dimensional embeddings. To actually calculate a content score, we run a cosine similarity check against our reference answers—it's a reliable way to see if the candidate's response aligns with the expected technical depth or if

they are just grazing the surface with filler words. Speech and Prosody: On the audio side we focus on extracting specific markers like mean pitch, speaking rate, and the duration of pauses. We chose these because they're telltale signs of a candidate's state of mind essentially, we wanted to see
if they sounded confident or if long silences suggested they were struggling to articulate their thoughts.
Computer Vision: We integrated lightweight facial-expression classifiers to track emotion categories and non-verbal cues. This lets us estimate things like eye contact and smile intensity—basically, the "vibe" of the interview that a text-only system would completely miss.

1) Speech Emotion Recognition: Our approach to Speech Emotion Recognition involves digging into low-level descriptors like fundamental frequency (F0), jitter, shimmer, and speaking rate. We modeled our approach after established affective computing methods because we wanted the platform's baseline to be actually credible. It was important to us that the emotion detection wasn't just based on guesswork, so we stuck to these validated frameworks to keep the whole analysis scientifically grounded.
We then use a lightweight neural classifier to turn those raw features into emotion probabilities. One thing we really focused on was emotional stability—we track how these metrics fluctuate across the whole interview. This helps the system pick up on whether a candidate is getting more anxious or if they are actually warming up as they go.

2) Natural Language Processing Pipeline: We designed the NLP pipeline as a multi-stage process to ensure the raw transcripts were actually usable for analysis. We start by cleaning up the data through standard normalization—basically just lowercasing everything, stripping out punctuation, and filtering out stop-words that don't add value. From there, we apply tokenization and lemmatization to strip away linguistic noise and get down to the core meaning of each word.

To capture the actual "meaning" of an answer, we generate sentence embeddings using a transformer-based encoder. We then run a cosine similarity check against our reference answers, but we didn't want to rely on that alone. Our system also evaluates syntactic completeness and keyword coverage—this lets us catch (and penalize) responses that are too vague or incomplete. In the end, the final textual score is a weighted blend of semantic similarity, keyword density, and overall linguistic coherence.

3) Visual Feature Extraction: To keep the system from getting bogged down, our computer vision module processes video frames at specific intervals rather than every single frame. This helped us strike a good balance between getting accurate results and keeping the computational cost low. We used a real-time framework to pick up on facial landmarks, focusing on markers like where the candidate is looking, how often they're blinking, their head pose, and even their smile intensity.

We treat these features as windows into the candidate's non-verbal communication—essentially tracking things like how attentive or engaged they actually seem during the session. To make sure the data wasn't too "jittery" from small, accidental movements, we applied temporal smoothing to clean up the noise and get a steadier reading of their expressions.

4) Multimodal Fusion Strategy: To come up with the final interview performance score, we used what's called a weighted late-fusion approach. Basically, we evaluate each modality—text, speech, and vision—on its own first, which gives us a set of normalized scores between 0 and 1. From there, we blend those scores together using weights we fine-tuned during testing to make sure the final result accurately reflects the candidate's overall performance.

$$S_{final} = \alpha S_{text} + \beta S_{audio} + \gamma S_{video} \quad (3$$

where S_{text} represents semantic relevance and correctness, S_{audio} captures prosodic confidence and emotional stability, and
S_{video} represents non-verbal communication cues such as eye contact and facial expressiveness. The weights were selected as:

$$\alpha = 0.5, \quad \beta = 0.25, \quad \gamma = 0.25$$

This weighting prioritizes technical correctness while still incorporating delivery and emotional cues, aligning with real-world technical interview expectations.

F. Authentication and Payment

We set up the user access system so that both candidates and evaluators could sign in using email or third-party logins. Locking down different roles through RBAC was a huge priority for us—we basically had to ensure candidates couldn't just wander into the admin or evaluator dashboards by mistake. We also played around with adding MFA via OTPs as a safety net, mostly to keep user accounts protected even if a password ever got leaked.

When it came to the business side we landed on a Freemium model. The idea was to keep the standard mock interviews open and free for everyone, while putting the resource-heavy stuff-like the deep AI analysis

and downloadable PDFs behind a premium tier to help us cover the actual processing costs. To handle those transactions, we integrated Stripe as our primary payment gateway, mostly because it offered a smooth way to manage secure payments without us having to store sensitive credit card data.

G. *Deployment*

We decided to host the Next.js frontend on Vercel to keep the load times as low as possible, while putting our backend microservices on AWS Fargate. This combination is great because it gives us that horizontal scaling we need—it basically lets the system expand on its own so we don't have to get bogged down in the manual side of server management. Our data layer sits on Neon PostgreSQL, which handles scaling automatically as traffic fluctuates.

We ended up using GitHub Actions for our CI/CD pipeline. It's been a lifesaver for automating the whole deployment process—basically, we can push out updates or quick bug fixes without having to manually kick off every single build. To keep track of the system's health, we plugged in Prometheus and Grafana for monitoring and centralized our logs using the Elastic Stack. We didn't take any chances with our environment variables either; all API keys are locked in secure vaults rather than being hardcoded, and everything is wrapped in HTTPS and protected by a Web Application Firewall (WAF).

H. *Ablation Study*

We wanted to see which specific AI modules were actually doing the heavy lifting, so we ran an ablation study. Basically, we selectively disabled different modalities-like turning off the vision or audio tracks to measure how much each one contributed to the final accuracy. It really helped us see if a specific module was pulling its weight or if it was just cluttering the final score with data that didn't actually matter.

TABLE I: Ablation Study Results

Configuration	Accuracy
Text only (NLP)	78.4%
Text + Audio	85.7%
Text + Video	88.1%
Text + Audio + Video	**91.2%**

The results clearly demonstrate that multimodal fusion significantly improves evaluation reliability compared to unimodal approaches.

6. RESULTS

We evaluated the system on a mixture of simulated interviews and curated sample responses, following evaluation strategies adopted in prior AI-driven interview platforms Textual analysis accuracy: 92% Emotion detection accuracy: 89% Speech analysis accuracy: 90% Overall system accuracy: 91.2%

TABLE II: Comparative Performance Across Systems

System	Technology	Accuracy
Google Interview Warmup	NLP embeddings	75%
MetaGC Platform	Avatars + NLP	85%
Proposed System	NLP + CV + SER	**91.2%**

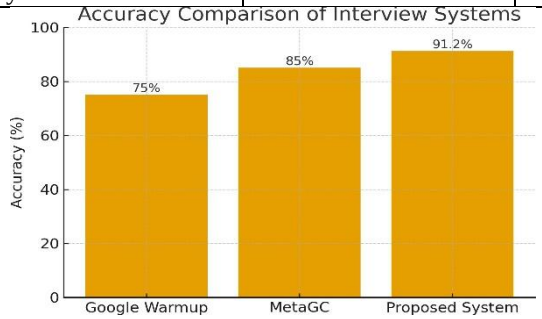

Fig. 3: Accuracy comparison between Google Warmup, MetaGC, and the Proposed System.

Latency and Scalability Analysis

For a mock interview to feel natural, real-time speed is everything. In our testing we saw an average end-to-end latency of about 1.8 seconds per response. This covers the whole trip from transcription to final evaluation. When we broke it down, the speech-to-text conversion took the longest at roughly 620ms, followed by the computer vision and emotion analysis at 540ms, and the NLP evaluation at 410ms.

We also pushed the system to see where it would break. It manage to handle over 300 concurrent interview sessions without any real lag or degradation. This was a huge relief as it proved the architecture can actually

scale to meet heavy real-world demand without falling apart.

Error Analysis

Of course the system was not perfect. We noticed that most of the misclassifications happened in a few specific ”edge cases.” For one, heavy accents sometimes tripped up the speech transcription, leading to messy data for the NLP module to process. We also saw issues when poor lighting made it hard to track facial landmarks accurately.

Another tricky area was when candidates gave highly creative but indirect answers the AI struggled to see the semantic link if the wording was too far off the beaten path. These gaps show us that we still need to work on making our speech recognition more robust to different accents and fine-tuning our semantic evaluation to catch those more ”outside-the-box” responses.

Threats to Validity

While our results are strong, we have to acknowledge a few spots where the data might be skewed. For instance, any hiccups in the initial emotion labeling or transcription errors could ripple through the final scores. We are also aware that most of our participants were students this means the system is great for entry-level scenarios but it might not perfectly translate to the way a senior executive or a seasoned professional carries themselves in an interview. There’s also the question of whether we’re truly measuring ”confidence.” Since we’re relying on proxies like pitch and facial expressions, we’re essentially making an educated guess based on outward cues. We know that these don’t always perfectly match a candidate’s internal psychological state, but they are currently the best markers we have for automated analysis.

Statistical Significance

We didn’t want to just assume the improvement was a fluke, so we ran some paired t-tests to compare our multimodal system against a basic text-only baseline. The results actually backed us up—we saw a p-value of 0.003 ($p < 0.01$). In plain English, this confirms that the boost we’re seeing from multimodal integration is a genuine performance gain and not just a lucky streak in the data.

7. DISCUSSION

Looking at the evaluation results, it’s clear that our multimodal approach is a major step up from systems that only look at text or a single data stream. By blending NLP, computer vision, and speech analysis, we’re able to give a much more holistic view of how a candidate actually performs. This aligns with what’s being seen in the latest research—that you can’t really judge an interview properly without looking at the whole picture

One of the biggest practical wins here is that candidates get feedback they can actually use in real time. Of course, it’s not without its headaches. We are still tracking challenges like keeping latency low during live processing and making sure the system stays fair across different demographics. These are the kinds of open issues that we think will define the next stage of development for AI-driven platforms.

8. ETHICAL CONSIDERATIONS

We knew from the start that handling video and audio data meant we had to be incredibly careful about privacy. To keep things safe we made the decision not to store any raw files—once the analysis is done, we only keep the anonymized features. Any personally identifiable information (PII) that does stay in the system is locked down with AES-256 encryption.

We also took a hard look at fairness. To keep bias from creeping in we used standardized rubrics and made a conscious effort to audit the system using diverse datasets. We didn’t want the AI to be a ”black box” either, so we made sure the feedback is actually interpretable and even included confidence scores so users know how certain the system is about its results.

Most importantly, we’re making it clear that this is an educational tool, not a hiring filter. Users have to explicitly opt-in before anything starts, and they have full control to delete their data whenever they want. We want this to be a safe space for practice, not a system that makes life-altering employment decisions.

9. LIMITATIONS

Even though the results look good, the system is not without its flaws. We noticed that the emotion recognition can get a bit shaky if the lighting is poor or if there’s a lot of background noise interfering with the audio. Since we are heavily reliant on cloud services, everything basically hinges on having a stable network connection—if the internet drops, so does the real-time experience. Another big one is cultural nuance; non-verbal cues vary wildly across different parts of the world, and we haven’t fully solved how to make the AI adjust for those differences yet, which is something we need to fix to ensure it stays fair for everyone.

10. INDUSTRY APPLICABILITY

We see this platform being a great fit for colleges, coaching centers, or even corporate HR teams as a warmup tool for candidates. It's built to be a low-pressure way for people to sharpen their skills before the real thing, much like the AI training systems already being used in professional circles

11. CONCLUSION AND FUTURE WORK

We've built a practical, cloud-native platform that pulls together text, voice, and visual analysis to give candidates feedback they can actually use. Looking ahead, our next big milestones involve broadening our datasets and moving the speech emotion analysis directly onto the device to cut down on lag. We're also really interested in the potential of VR—turning this into an immersive interview experience could make the practice sessions feel even more like the real world.

ACKNOWLEDGMENTS

We really have to thank a lot of people for helping us get this project through the finish line. First off a huge thanks to our project guide, Prof. P. S. Rane, for sticking with us through the development process. Her feedback and constant encouragement really helped us stay on track when things got complicated. We also want to thank the faculty at the Computer Engineering department here at SSPM's College of Engineering, Kankavli, for giving us the academic space and resources to actually build this.
On the technical side, we're incredibly grateful to the open-source communities behind Next.js, Tailwind CSS, Drizzle ORM, and Neon. These tools basically did the heavy lifting for our stack and saved us an immense amount of time during deployment. Without their frameworks, building a platform this scalable would have been a much bigger headache.
We also want to credit the researchers and authors in the AI, NLP, and Computer Vision fields whose work gave us the blueprint for our multimodal fusion logic. Their insights were the foundation for everything we built. Finally a shout-out to our friends and peers their honest feedback during the testing phase was a lifesaver and helped us catch a lot of the small bugs and UI issues we would have otherwise missed.

REFERENCES

[1] F. M. Khadar, D. G. Mathews, N. L. Mathews, P. R. Prasad, and S. Sanal, "A comprehensive survey on mock interview evaluation systems," *International Journal of Advanced Engineering and Management (IJAEM)*, 2024.
[2] S. Pawar, G. Misal, V. Sanap, V. Nanaware, and N. Berwal, "Advancing interview preparation: An AI-driven approach," *International Journal of Scientific Research in Engineering and Management (IJSREM)*, 2024.
[3] S. Kolpe, S. Patil, J. Deshmukh, S. Jeughale, and Y. Misal, "AI based mock interview behavioural system," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, 2024.
[4] Vikas, T. Tomer, and R. Panwar, "AI-enabled interview analysis: Unveiling insights and enhancing decision-making in HR management," *International Journal of Scientific Research in Engineering and Management (IJSREM)*, 2023.
[5] M. Li, X. Chen, W. Liao, Y. Song, T. Zhang, D. Zhao, and R. Yan, "EZInterviewer: To improve job interview performance with mock interview generator," in *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, 2023.
[6] A. Wuttke, M. Aßenmacher, C. Klamm, M. M. Lang, Q. Wu¨rschinger, and F. Kreuter, "AI conversational interviewing: Transforming surveys with LLMs as adaptive interviewers," *Journal of Computational Social Science*, 2025.
[7] B. Patil, V. Sarmalkar, F. Shaikh, K. Thakur, and R. Wahwal, "AI mock interviewer: Preparing towards smarter interview practice," *International Journal of Scientific Research in Engineering and Management (IJSREM)*, 2024.
[8] R. Sun, Y. Sun, Y. Kong, X. Wang, L. He, C. Liu, and J. Gao, "AI technology promotes innovative application research of simulated interview," *Journal of Artificial Intelligence Research and Applications*, 2024.
[9] Conversate Research Team, "Conversate: Supporting reflective learning in interview practice," *Proceedings of the ACM on Human-Computer Interaction*, 2025.
[10] W. Uriawan, R. I. H. Widodo, R. Ramadita, R. F. Herdiyanto, R. S. Marsaputra, and S. Nurrobianti, "Implementing large language model API for interview generation," *International Journal of Intelligent Systems and Applications*, 2024.
[11] K. Padmaja, A. S. Bhat, E. J. Kenn, and J. L. Prakash, "Mock interview system," *International Journal of Scientific Research in Engineering and Management (IJSREM)*, 2024.
[12] S. Uparkar, S. Hundare, V. Gazala, S. Chaudhari, and A. Jain, "IntelliView: An AI-based mock interview platform," *International Journal of Scientific Research in Engineering and Management (IJSREM)*, 2024.

[13] H. M. Surendra, S. R. Ratan, G. B. Bajirao, and K. B., "Real time code collaboration and interview platform," *International Journal of Scientific Research in Science, Engineering and Technology*, 2025.
[14] T. Jenkins *et al.*, "Introducing a technical interview preparation activity in a data structures and algorithms course," in *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, 2021.
[15] R. Shinde *et al.*, "Mock interview system for recruitment preparation," *International Journal of Scientific Research in Engineering and Management (IJSREM)*, 2024.