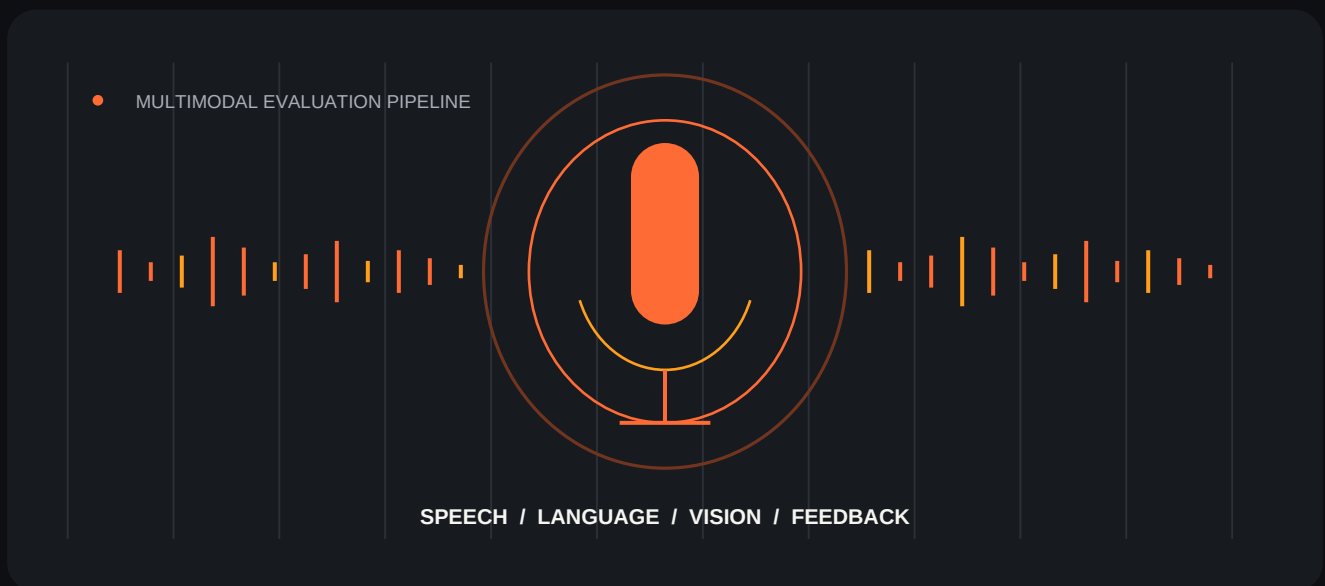


MOCKAI

AI-POWERED INTERVIEW EVALUATION PLATFORM

Turning interview practice into structured, multimodal feedback.



MULTIMODAL

Text • speech • facial signals

FULL STACK

Candidate + admin portals

AI PIPELINE

Analysis → report

ENGINEERING

Performance + reliability

A full-stack project exploring how speech processing, language analysis, and available facial-expression signals can support a more structured interview-practice experience.

PRODUCT
MockAI

FOCUS
AI + full-stack engineering

DELIVERABLE
Candidate experience, admin portal, evaluation reports

01 / THE PRODUCT

The challenge: interview practice can be difficult to improve when feedback is vague or inconsistent. MockAI was built to make the experience more structured—from completing a session to reviewing a multi-dimensional report.

PRODUCT GOALS

REPEATABLE PRACTICE

Candidate sign-in, interview sessions, response submission, and access to previous results.

MORE THAN A TRANSCRIPT

Language, speech-delivery, and optional facial-video signals contribute different perspectives.

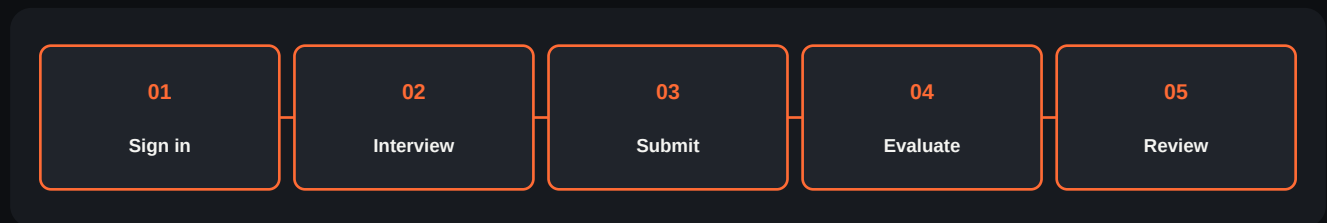
ACTIONABLE PRESENTATION

Dimension summaries, charts, and feedback help candidates review performance.

ADMIN OVERSIGHT

An admin interface supports question-bank and platform-level review workflows.

CANDIDATE JOURNEY

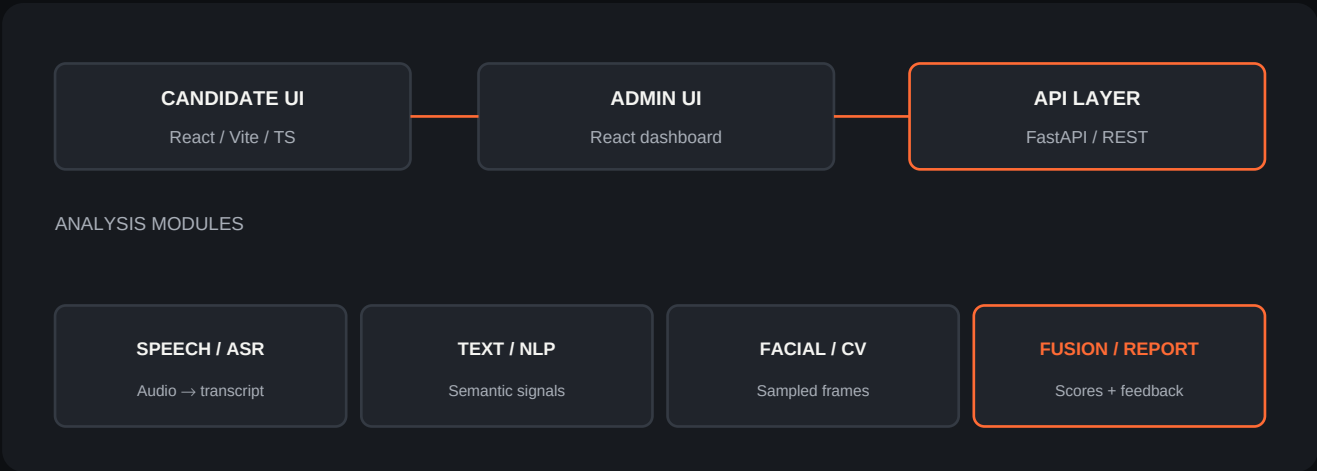


The backend coordinates the interview and evaluation lifecycle, persists records, and exposes results to the candidate-facing experience. The admin portal provides separate management and review surfaces.

Scope note: this case study describes the implemented workflows and reported local validation. It does not claim independently measured model accuracy or production-scale performance.

02 / SYSTEM ARCHITECTURE

The application separates candidate and admin experiences from API orchestration and individual analysis modules. This modular structure makes it easier to validate and optimize different stages of the workflow.



TECHNOLOGY STACK

LAYER	TECHNOLOGY / APPROACH	RESPONSIBILITY
Frontend	React, Vite, TypeScript; Tailwind/CSS styling	Candidate interview flow, results, and admin UI
API backend	Python, FastAPI, REST endpoints	Authentication, orchestration, and result APIs
Data	MongoDB	User, interview, question-bank, and evaluation records
Speech	Speech-to-text / ASR pipeline	Converts recorded speech into transcript data
Language	all-MiniLM-L6-v2 embeddings + scoring heuristics	Semantic alignment, coverage, relevance, and completeness signals
Facial analysis	YuNet + FERPlus ONNX; Haar fallback	Face detection and predicted expression labels from sampled frames
Media	FFmpeg; local filesystem storage	Normalizes and processes interview recordings

The architecture graphic is conceptual. In the described setup, interview media is handled through local filesystem storage; deployment topology may vary by environment.

03 / MULTIMODAL EVALUATION

The pipeline combines evidence sources when available. Each modality contributes a different perspective, and unavailable evidence should be shown as unassessed rather than silently converted into a low score.

01 TEXT & ANSWER QUALITY

Transcript analysis considers semantic alignment, coverage, relevance, and completeness. Heuristics flag short fragments, question echoes, and keyword-only responses.

02 SPEECH & DELIVERY

Speech processing supplies transcript data and delivery-related features such as speaking pace and pause patterns. These are guidance signals, not universal rules.

03 FACIAL SIGNALS

When video is available, sampled frames are analyzed for faces and predicted expression labels. Missing video or no detected face should remain explicit.

INPUT → PROCESSING → REPORT

INPUT	PROCESSING	OUTPUT
Recorded answer, transcript, optional video	ASR, semantic analysis, delivery features, sampled-frame analysis	Dimension scores, summary, charts, feedback

RELIABILITY & INTERPRETATION

Result handling was updated to preserve missing-value states and avoid showing unassessed confidence or stress as measured results. Facial-expression labels and heuristic confidence/stress indicators are model-derived signals—not reliable measurements of internal emotional state.

MockAI is designed for interview practice and reflection, not as a standalone hiring decision system or clinical assessment tool.

04 / ENGINEERING & OUTCOMES

The project includes performance work, edge-case handling, and consistency checks between backend result fields and the candidate-facing dashboard.

IMPLEMENTATION HIGHLIGHTS

NLP inference reuse

Model resources are loaded once and embedding work is batched to reduce repeated setup and inference overhead.

Speech feature optimization

Acoustic pause analysis was optimized; reported local timing for that step improved from approximately 1,779 ms to 19 ms.

Facial-video processing

Frame stepping, resizing, ONNX inference optimizations, and normalized media reuse reduced reported frame-stepping time from about 1,109 ms to 267 ms and FER inference from about 841 ms to 385 ms.

Result-state correctness

Dashboard dimension mappings were aligned with API output; missing vision and pending evaluation states preserve null/unassessed values.

Answer-quality heuristics

Checks for short fragments, question echoes, and keyword fragments were added to reduce misleadingly strong feedback for obviously weak answers.

VALIDATION SNAPSHOT

6/6

Focused dashboard/UI tests

14

Facial optimization tests added

23

Backend test modules in inventory

0

Errors in reported production build

Reported validation also covered authentication, interview lifecycle, background evaluation, persistence, result APIs, dashboard statistics, and edge cases. Timings are environment-specific engineering observations—not a benchmark study or a labeled-dataset accuracy evaluation.

WHAT THIS PROJECT DEMONSTRATES

FULL-STACK DELIVERY

Connected candidate experience and admin interface backed by APIs and persisted evaluation records.

MULTIMODAL SYSTEM DESIGN

Language, speech-delivery, and optional facial signals flow into a structured report.

RELIABILITY ENGINEERING

Attention to inference cost, media handling, edge cases, and API/UI consistency.

NEXT VALIDATION STEP

Measure scoring quality with a labeled, consented dataset and test production-like load.

MockAI turns interview practice into a structured feedback loop—combining a usable web product with a modular evaluation pipeline and a focus on honest, interpretable outputs.

TECHNOLOGIES React • Vite • TypeScript • FastAPI • Python • MongoDB • ONNX Runtime • FFmpeg

This case study reflects implementation details and local test results reported in the project context. It does not claim independently measured model accuracy, production-scale capacity, or improved hiring outcomes.