Interaura: AI Mock Interview Platform

Team: AI Forge

1. Problem Statement

Job seekers today face a "Black Box" hiring process characterized by high anxiety and a lack of transparency. Key pain points include:

- **Resume Uncertainty:** Candidates often do not know if their resume meets Applicant Tracking System (ATS) criteria.
- **Lack of Feedback:** Rejections rarely come with actionable insights, leaving candidates unaware of their specific weaknesses.
- **Limited Access:** Quality mock interviews with human mentors are often expensive, hard to schedule, or unavailable.
- **Interview Anxiety:** Many candidates struggle with confidence and "freeze up" during technical questioning due to a lack of low-pressure practice environments.

2. Solution & Key Innovations

Interaura is a full-stack, AI-powered platform designed to democratize interview preparation. It simulates real-world interviews and provides objective, data-driven feedback.

- **AI Mock Interviews:** Real-time audio/video simulations where an AI interviewer asks role-specific questions (Technical, Behavioral, System Design).
- **Multi-Modal Analysis:** A holistic feedback system that evaluates *what* you say (Content), *how* you say it (Speech/Tone), and *how* you present yourself (Visual Presence).
- **Smart Resume Analysis:** A hybrid parsing engine that decodes ATS logic, providing a "Match Score" and highlighting missing keywords relative to a target job description.
- **Real-Time Processing:** Low-latency architecture ensures the AI responds naturally (< 2 seconds), mimicking human conversation flow.

3. Technical Architecture

The platform utilizes a modern, event-driven architecture designed for scalability and speed.

Frontend (Client)

- **Framework:** Built with **React 18** and **Vite** for a responsive user interface.
- **Media Capture:** Leverages **WebRTC** and the **MediaRecorder API** to capture high-quality audio and video directly in the browser.
- **Styling: Tailwind CSS** provides utility-first styling for a clean, professional aesthetic.

Backend (API Gateway)

- **Core: FastAPI (Python)** serves as the core backend, chosen for its high-performance asynchronous capabilities needed for handling simultaneous media streams.
- **Communication: WebSockets** enable bi-directional, real-time communication between the client and AI services.

AI Processing Layer

- **LLM (Logic): Groq (Llama 3.1 70B)** is used for ultra-low latency question generation and content evaluation. **OpenAI GPT-4o** serves as a fallback for complex reasoning tasks.
- **Speech Services: Deepgram Nova-2** provides industry-leading real-time Speech-to-Text

(STT) with filler word detection.
- **Computer Vision: OpenCV** analyzes video frames to track eye contact, head pose, and movement stability.
- **Resume Engine: Google Gemini 1.5 Flash** combined with **scikit-learn (TF-IDF)** performs semantic and keyword matching for resumes.

Data & Storage

- **Database: Supabase (PostgreSQL)** stores user profiles, interview transcripts, and feedback reports.
- **Security: Row Level Security (RLS)** ensures strict data isolation for every user.

Frontend (Next.js)

|

Backend (FastAPI)

├── Resume Module

├── Interview Engine

├── Media Service (FFMPEG + Deepgram)

├── Evaluation Layer (Groq)

└── Supabase Database

4. Approach and AI Components

The system's distinct advantage is its **Multi-Modal Analysis**:

- **Content Quality (LLM):** The system transcribes the user's answer and feeds it to the Groq LLM. It evaluates technical accuracy, depth, and structure against the specific question asked, assigning a score (1-10).
- **Speech Delivery (Audio):** Deepgram analyzes the audio stream to calculate pace (Words Per Minute) and identify hesitation markers (filler words like "um", "uh"), which are key indicators of confidence.
- **Visual Presence (Vision):** OpenCV processes video frames to estimate gaze direction. It calculates an "Eye Contact Score" by tracking how often the user looks at the camera versus looking away.

Scoring Algorithm:

Overall Score = (Content × 0.40) + (Visual × 0.30) + (Speech × 0.20) + (Communication × 0.10)

Interaura: AI Mock Interview Platform

5. Challenges & Mitigations

Challenge 1: Real-Time Video Latency

- **Problem:** Processing every video frame caused significant lag (30s processing for 2 mins video).
- **Mitigation:** Implemented frame sampling (processing every 5th frame) and asynchronous pipelines, reducing processing time to ~8 seconds.

Challenge 2: LLM Rate Limits

- **Problem:** High API usage led to quota errors on the free tier.
- **Mitigation:** Developed a request queuing system with exponential backoff and a fallback mechanism that switches to rule-based analysis if the AI service fails.

Challenge 3: Eye Contact Accuracy

- **Problem:** Poor lighting resulted in false positives for gaze tracking.
- **Mitigation:** Created a multi-stage validation (Face → Eyes → Gaze) with confidence thresholding, improving accuracy from 65% to 92%.

Challenge 4: Browser Compatibility

- **Problem:** MediaRecorder support varied across browsers (Chrome vs Safari).

- **Mitigation:** Added robust feature detection and codec negotiation (WebM/MP4) to ensure functionality across all modern browsers.

6. Roadmap to Final Build (4 Weeks)

- **Phase 1: Foundations (Weeks 1)**
  - Finalize Figma designs, database schema, and API contracts. Setup CI/CD pipelines.
- **Phase 2: Core Text-Only MVP (Weeks 2-3)**
  - Build authentication and text-only interview interface to validate prompt engineering and logic.
- **Phase 3: Audio & Real-time Integration (Weeks 3)**
  - Implement real-time audio capture, STT (Deepgram), and TTS integration via WebSockets.
- **Phase 4: Feedback Engine & Polish (Weeks 3-4)**
  - Develop the asynchronous report generation service, polish the UI, and conduct internal alpha testing.
- **Phase 5: Launch**
  - Deploy to production, setup monitoring (Sentry), and release for public use.