AI Mock Interview Simulator – Technical Documentation
Round 2 Submission

Team – 403 Forbidden – 70% Completed

1. Introduction

The AI Mock Interview Simulator is designed to provide students with an intelligent, real-time, interactive interview experience powered by advanced AI models. The system replicates a human interviewer through voice, video avatar, and AI-driven evaluation, offering personalized feedback on technical skills, communication, confidence, and behavioural performance.

This project aims to address the gap between theoretical preparation and real-world interview experience by providing a scalable, adaptive, and fully automated interview tool accessible through the web.

In Round 2 (Prototype Development, 60–80% completion), the focus has been on establishing the **core AI pipeline, backend architecture, model integration, persona-based media functioning, and complete frontend–backend communication**. A cloud-based deployment has been completed to ensure accessibility and testing readiness for mentors.

2. Problem Statement

Students often struggle to prepare effectively for interviews due to limited access to expert mentors, lack of personalized feedback, and the absence of real-time mock evaluation systems. Traditional mock interview platforms typically lack adaptability, multimodal inputs, and AI-driven scoring mechanisms.

The objective of this system is to build an AI-powered tool that:

- Conducts realistic mock interviews using voice and avatar personas
- Generates personalized interview questions based on a user's resume/profile
- Evaluates answers using technical, communication, and behavioural metrics
- Produces structured reports and improvement recommendations
- Works both online (cloud inference) and offline (local model inference in Round 3)

3. Technical Architecture Overview

We finalized a modular, scalable architecture that separates the system into independent layers:

3.1 Frontend (React + Tailwind)

- Responsive UI for interview flow, dashboard, and reports
- Real-time camera and microphone integration
- Human-like interviewer avatar video rendering (male/female persona)
- Speech recognition for capturing user answers
- Persona-based TTS output (male/female voice)

3.2 Backend (FastAPI)

- API for profile parsing, question generation, answer submission, and report synthesis
- Integration with OpenRouter for cloud inference
- Session management, scoring pipelines, and report generation
- Secure communication with the frontend

3.3 AI Pipeline

- **Cloud AI in Round 2:**
  Using **OpenRouter** models (e.g., Claude, Mixtral, Amazon Nova Lite) for:
  - Question generation
  - Answer evaluation
  - Report summarization
  - Answer improvement suggestions
- **Offline AI in Round 3:**
  We will run optimized small-to-medium LLMs locally (e.g., Llama-3-8B, Mistral 7B) with GPU inference to comply with the offline requirement.

3.4 Database

- SQLite for storing sessions during prototype development
- Tracks interview sessions, scores, and timestamps

3.5 Deployment

- **Frontend:** Vercel
- **Backend:** Render / Railway
- Integrated cloud build pipeline and Downloadable PDF reports integrated

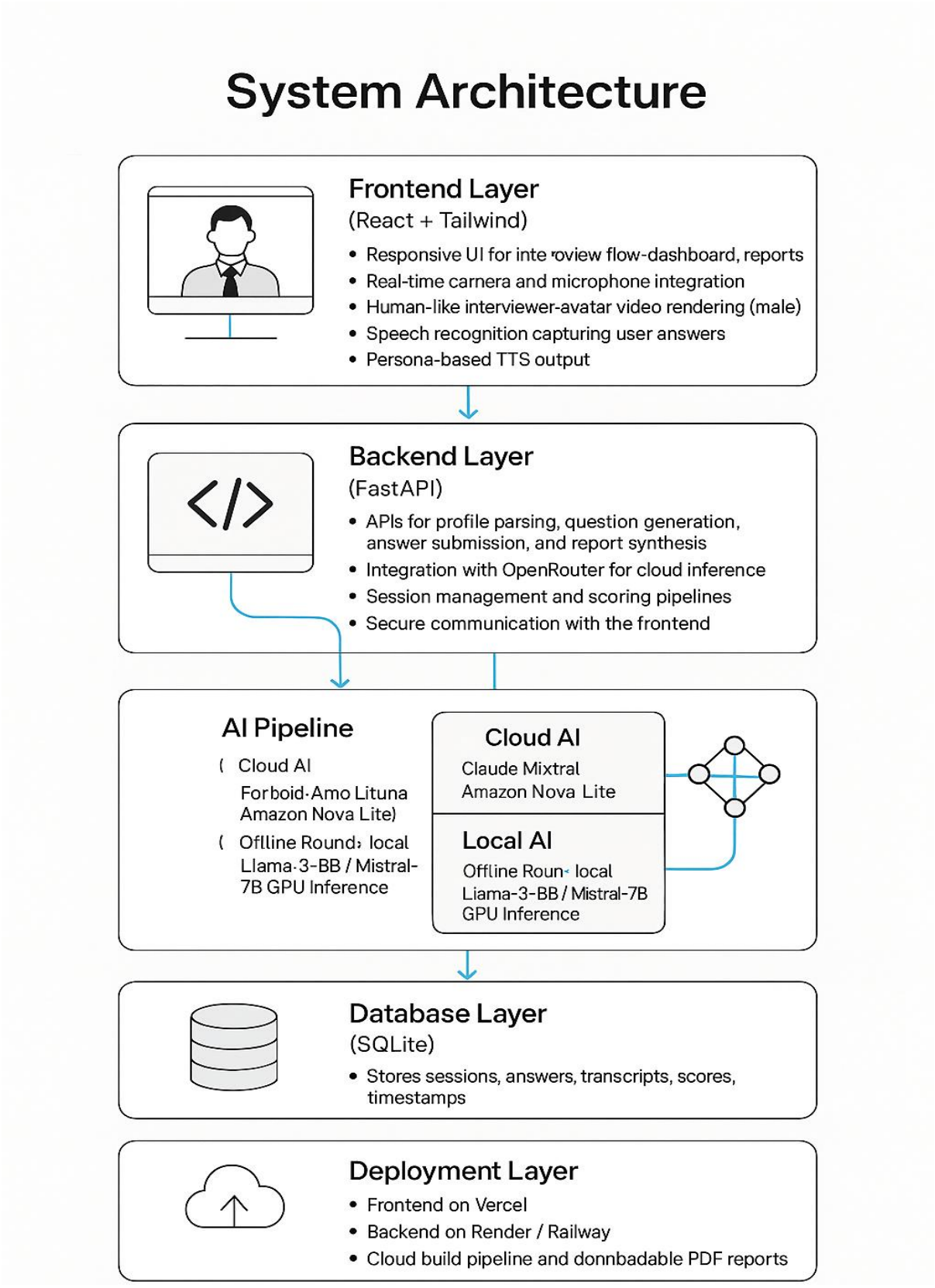

Figure 1: This Diagram represents System Architecture of AI Interview

4. Week-by-Week Development Summary

Week 1: Architecture Finalization & Setup

- Established complete architecture diagrams and workflow design
- Set up FastAPI backend with modular routing
- Created database schema for storing interview records
- Developed initial UI wireframes and landing page
- Configured GitHub repository with README, environment setup, and instructions
- Integrated resume parsing workflow

Deliverable Achieved: Architecture document + repository ready for mentor review

---

Week 2: Core AI Model Integration

- Integrated OpenRouter API for question generation and evaluation
- Implemented the interview pipeline:
  Profile → Question Generation → Answer Capture → AI Evaluation → Report Storage
- Developed fallback mechanisms for offline and error scenarios
- Performed internal testing with sample users
- Created API documentation and testing suite

Deliverable Achieved: Functional backend with working AI inference pipeline

---

Week 3: Frontend & Workflow Integration

- Connected frontend with backend endpoints
- Implemented interviewer persona selection (male/female/bossy female)
- Added **male interviewer video + male voice** and **female interviewer video + female voice**
- Fully functional interview flow with seamless transitions
- Speech-to-text capture, live recording indicator, and persona-based TTS
- PDF report generator implemented and deployed
- End-to-end testing completed on cloud deployment

Deliverable Achieved: Complete working prototype with frontend-backend-AI integration.

5. Model Training & Accuracy

Although the current prototype uses **cloud inference**, we have prepared datasets and scripts for offline fine-tuning in Round 3.

Round 2 (Cloud Models)

- Used OpenRouter inference (Claude, Mixtral, Nova Lite)
- No on-device training required
- Evaluation and scoring performed via deterministic prompts

Round 3 (Offline Fine-Tuning)

We will shift to offline models to comply with the onsite rules. Planned steps:

- Train/fine-tune Llama-3-8B or Mistral-7B on interview Q&A datasets
- Build local vector database for skill-based question generation
- Run quantized models for fast inference on CPU/GPU
- Technical accuracy and scoring calibration will be improved using rubric-based supervised fine-tuning

The final model will operate **completely offline**, including question generation, evaluation, and report generation.

6. Functional Prototype Summary (60–80% Completion Achieved)

Working Features

1. Resume parsing
2. AI-generated interview questions
3. Persona-based voice and video interviewer
4. Real-time interview simulation
5. Live transcript and recording
6. AI evaluation with score distribution
7. Automatic report generation (PDF enabled)
8. Cloud deployment for demonstration

Stability & Technical Feasibility

- Backend handles session-based question flow without crashes
- AI calls are optimized with caching and fallbacks
- Video and audio systems stable across browsers
- End-to-end flow verified on remote deployment

7. Challenges and Mitigation

Challenge 1: Real-time audio/video synchronization

Solution: Implemented robust state management and media hooks.

Challenge 2: Cloud model reliability / rate limits

Solution: Added local fallback templates & caching.

Challenge 3: Persona mismatches (video/voice)

Solution: Reworked persona logic to switch between male and female videos + voices seamlessly.

Challenge 4: Large prompt cost & latency

Solution: Optimized prompts to ≤900 tokens and enabled partial response usage.

8. Roadmap to Final Build (Round 3)

Planned Enhancements

- Offline LLM inference pipeline
- Improved scoring algorithms
- Stronger datasets for training
- More interviewer personas
- UI polishing and animations
- Model analytics dashboard
- Enhanced facial analysis and sentiment scoring

Final Goal

Deliver a full offline-capable AI Interview Simulator running completely on-device during the IIT Bombay finale.

9. Conclusion

Our team has successfully completed **over 70% of the system**, with a fully working AI-powered prototype deployed online, featuring real-time interviews, persona-based avatars, AI evaluation, scoring, and report generation.

The system demonstrates strong technical feasibility, scalability, and educational impact. With the foundation complete, the Round 3 offline implementation will focus on refining accuracy, optimizing local inference, and delivering a polished final product.