# OVERVIEW OF THE IMPLEMENTATION

April 8, 2025

# Contents

You are building an app that simulates a realistic human interviewer using an AI-powered avatar. The avatar speaks interview questions, listens to the user's answers, analyzes their responses (both behavioral and technical), gives instant feedback, and can simulate real interview scenarios. The experience needs to be immersive, human-like, interactive, and intelligent.

---

# 1 SYSTEM ARCHITECTURE OVERVIEW

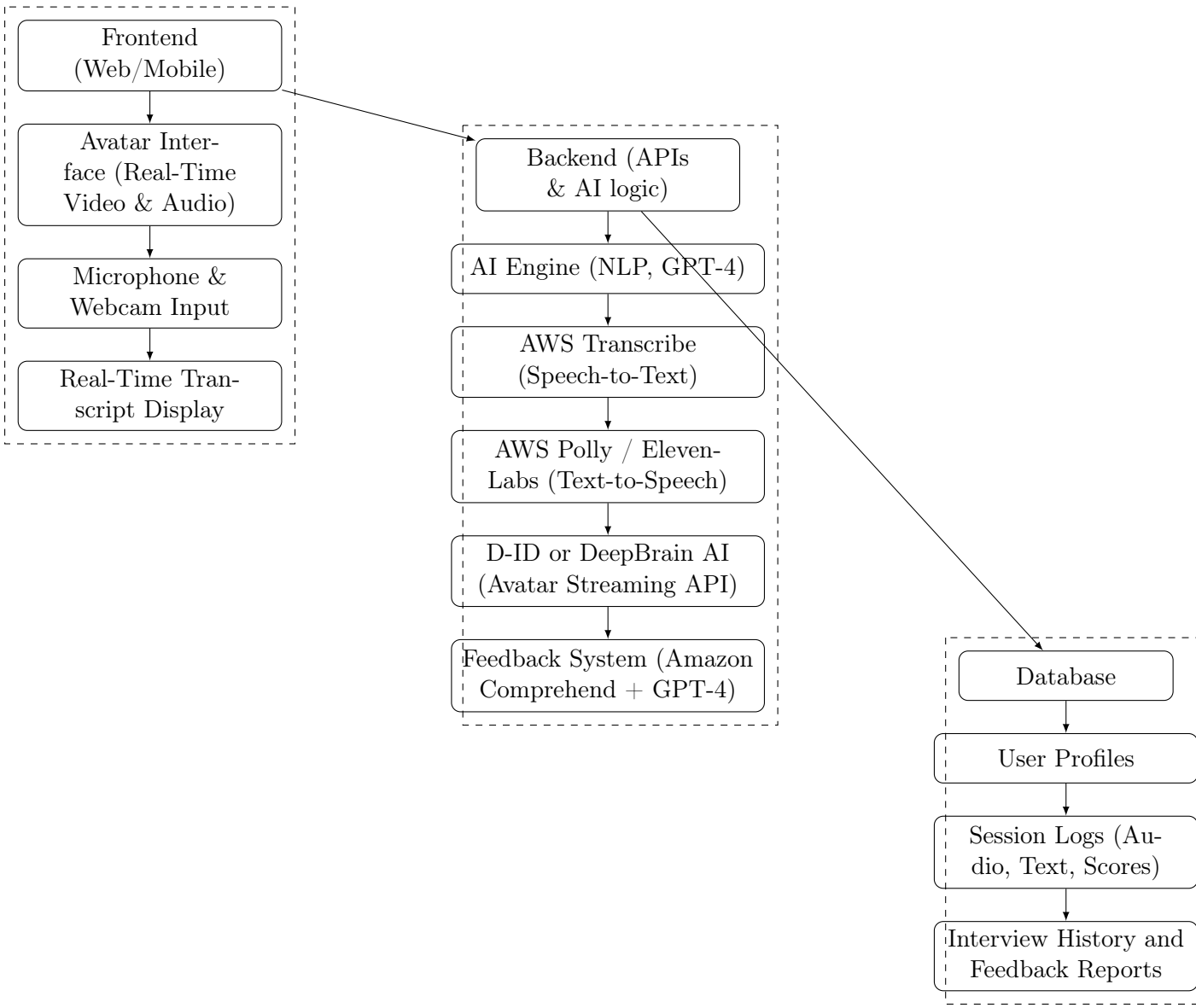


Figure 1: High-Level System Architecture

---

# 2 IMPLEMENTING A REALISTIC AI AVATAR

This is one of the most critical parts of the product. You need an avatar that:

- Looks human and talks naturally.
- Can respond in real time.
- Has facial expressions and gestures.
- Feels like a live person, not a robot.

### Tools & Technologies:

**D-ID Live Streaming Avatar:** Supports real-time lip-synced avatars using text input or speech.

**DeepBrain AI:** Offers highly realistic avatars (used by banks and newsrooms in Korea).

**Synthesia (Limited Interactivity):** Great video generation but less suited for real-time interaction.

**Unreal Engine + Metahuman:** For building your own custom avatars (needs a 3D dev team).

### Suggested Implementation Path:

For MVP, use D-ID Live or DeepBrain API to stream a photorealistic avatar directly in your web or mobile app.

### Implementation Steps:

1. **Design or Select Avatar:**
   - Choose a realistic human avatar from D-ID or DeepBrain's library (male/female, ethnic diversity).
   - Customize clothing, background, and facial features to suit a professional environment.
2. **Integrate Live Avatar API:**
   - Use D-ID's live API (via WebSocket or REST).
   - Send the avatar the text you want it to speak (from GPT-4 or pre-set scripts). The avatar speaks, and the video renders in real-time.
3. **Sync with Voice (Text-to-Speech):**
   - Use ElevenLabs for hyper-realistic speech (more human than Amazon Polly).
   - Convert GPT-4 output or pre-written interview questions into voice.
   - Send voice to D-ID for facial motion sync.
4. **Render in Frontend:**
   - Use a React or Flutter frontend to embed the avatar using D-ID's iframe or video stream.
   - Ensure user microphone and camera access are enabled.
5. **User Interaction:**
   - When the avatar asks a question, start capturing the user's voice.
   - Real-time speech-to-text transcribes their response.

---

# 3 REAL-TIME INTERACTION FLOW

Here's what the real-time session will feel like and how it runs in the background.

**Example Scenario:** User logs in > Chooses job role > Starts session > Avatar appears and begins interviewing

### Step-by-Step Flow:

A. **Avatar Starts the Interview:**

- GPT-4 or prompt-based logic generates the first interview question.
- ElevenLabs converts the question into realistic voice.
- D-ID animates the avatar to speak the question.
- Avatar video is streamed to the user in the app.

B. **User Responds:**

- User speaks answer via microphone.
- Audio is streamed to AWS Transcribe (or Google Speech-to-Text).
- Transcription is sent to backend in near real-time.
- Text is logged and sent to GPT-4 for feedback generation.

C. **Real-Time Feedback (Optional Mid-Session):**

- Amazon Comprehend analyzes tone and emotion.
- GPT-4 assesses clarity, structure, and technical knowledge.
- Optional: Display soft prompts like "Good answer, now let's talk about leadership."

D. **Continue Interview Loop:**

- Avatar continues next question.
- Session follows a question-response-feedback loop for 5–10 minutes.

E. **End Session with Feedback:**

- GPT-4 summarizes performance.
- System generates a report:
  - Communication score
  - Confidence and fluency
  - Technical answer quality
  - Suggested improvements

---

# 4 TECHNICAL INTERVIEW MODULE IMPLEMENTATION

To support technical interviews, especially for engineering or developer roles:

### Additions to the MVP:

- **Code Editor (Frontend):**
  - Use Monaco Editor (used by VSCode).
  - Embed in browser for user to write code during interviews.
- **Coding Question Generator:**
  - GPT-4 can generate coding challenges or use preset challenges from a library.
  - Questions like "Reverse a linked list in Python."
- **Code Evaluation:**
  - Use Judge0 API or integrate Python/Pyodide for browser-side execution.
  - GPT-4 can also "review" code for correctness, readability, and performance.
- **Live Feedback:**
  - After submission, GPT-4 provides:
    - Whether the solution is correct
    - Time/space complexity
    - Suggestions
- **Scoreboard:**
  - Add a scoring engine to track interview performance.

---

## 5 FRONTEND EXPERIENCE

**Built With:** React.js + Tailwind + Three.js (for UI polish)

### Components:

- Landing Page (Pitch the product)
- Onboarding Flow (Choose role, upload resume, etc.)
- Interview Screen
  - Live avatar window
  - Speech-to-text transcript pane
  - Timer and question progress bar
  - Optional webcam feedback window
- Feedback Page
  - Session summary
  - Downloadable report
  - Visual performance charts

---

# 6 DATA FLOW DIAGRAM (SIMPLIFIED)

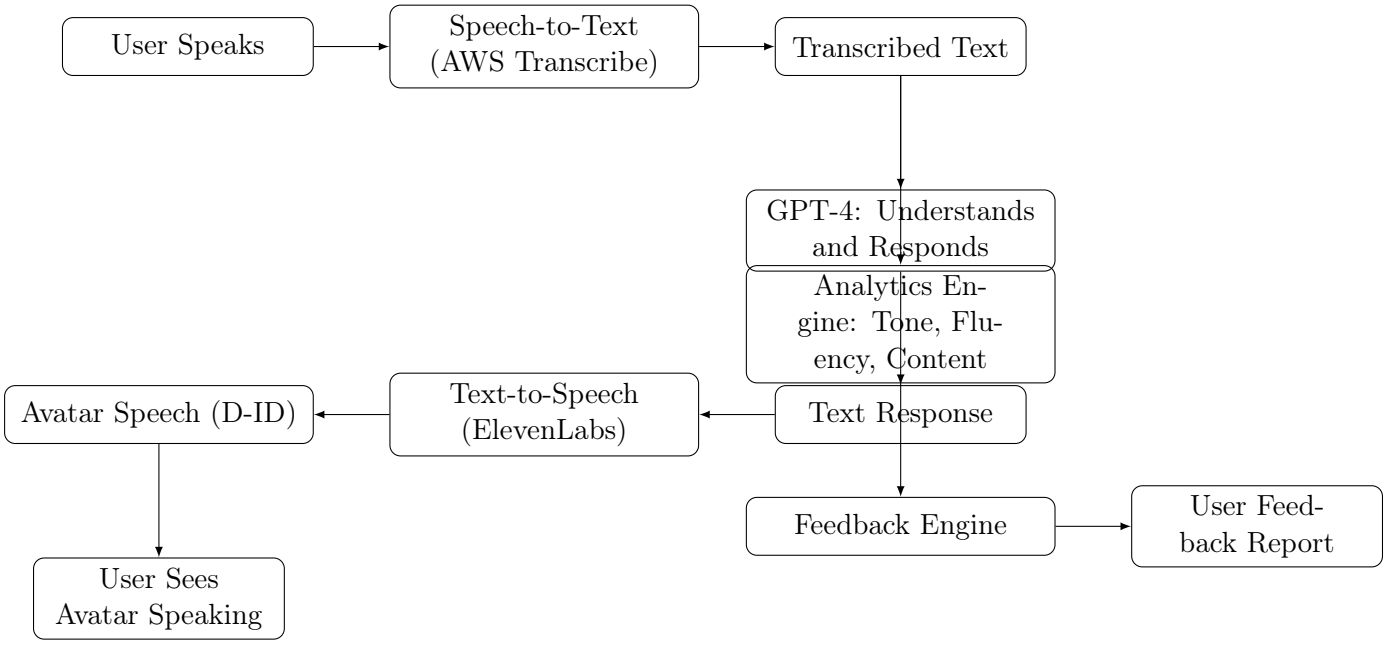


Figure 2: Simplified Data Flow

---

# 7 STORAGE & INFRASTRUCTURE

**Services:**

- **User Data & Authentication:** Firebase Auth or AWS Cognito
- **Interview Logs:** Stored in AWS S3 or Firebase Firestore
- **Session Data:** MongoDB or DynamoDB
- **Media Files (optional):** Store voice or video samples (with consent)

**Hosting:**

- AWS Amplify or Vercel for frontend
- AWS Lambda or Node backend via Express/FastAPI
- CI/CD with GitHub Actions

---

# 8 FUTURE EXPANSIONS

- Mobile App: React Native or Flutter version
- Multilingual Support: Add support for Spanish, French, etc.

- Custom Avatars: Users can choose their interviewer's gender, voice, accent
- Interview Coaching: Personalized learning paths based on past interview performance
- Enterprise Dashboard: For schools, bootcamps, or HR teams

---

## CLOSING THOUGHTS

This implementation combines AI, avatar tech, real-time NLP, and user-centric UX to create a seamless interview simulation experience. It's both technically viable and commercially scalable with the right execution.