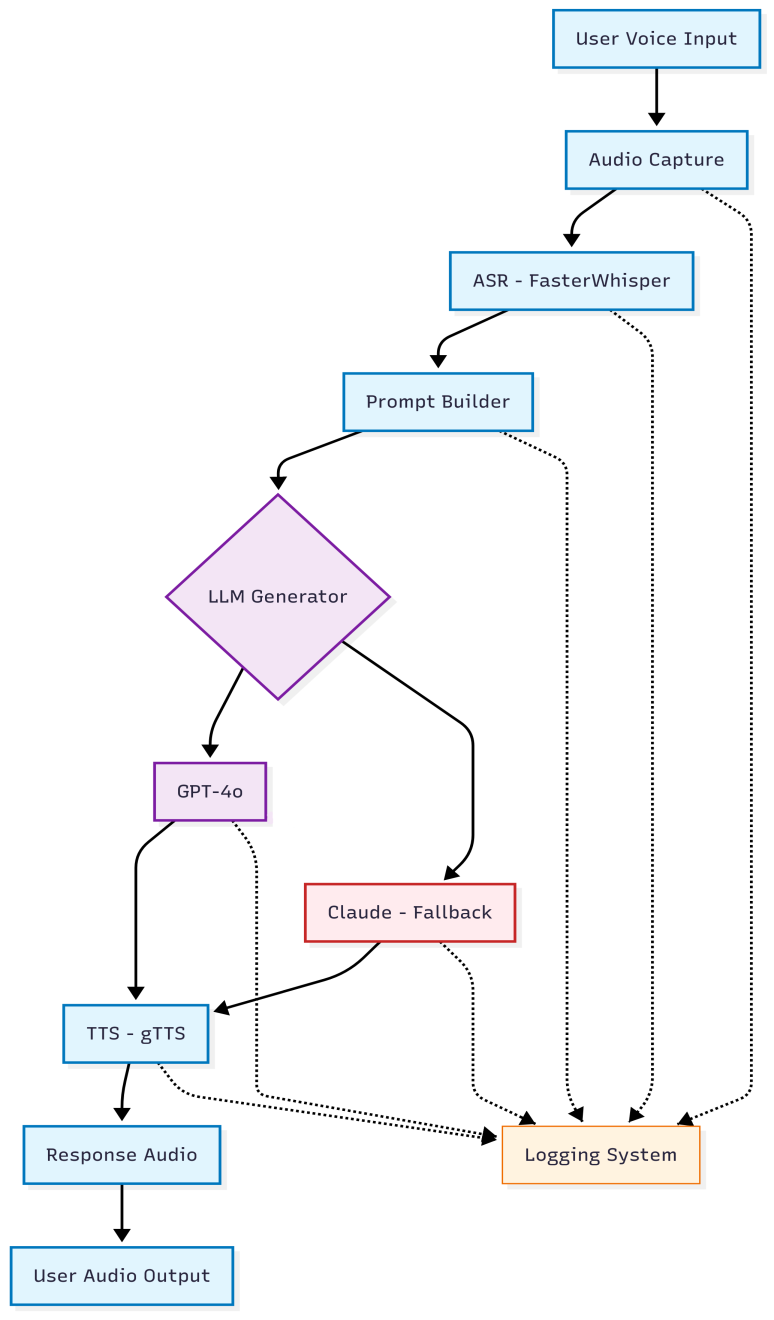
**Architecture**

**Overview**  
This document describes the system architecture for a culturally sensitive, therapeutic-grade, Omani Arabic voice assistant. The assistant listens to a user’s voice, understands their mental health needs, generates appropriate responses using large language models, and replies back in natural-sounding Omani Arabic.

**Component Breakdown**

* **Audio Capture**  
  Captures voice input from the user's microphone using Gradio or sounddevice. Records and saves a .wav file for further processing.
* **ASR (Automatic Speech Recognition)**  
  FasterWhisper (a faster implementation of OpenAI's Whisper) is used to transcribe Arabic audio into text. Language is specified as Arabic (ar), with preprocessing done to remove silence.
* **Prompt Builder**  
  Wraps the user’s transcribed text in a structured, culturally sensitive system prompt. Incorporates Islamic values and therapeutic tone.
* **LLM Generator**  
  Handles the main logic for generating replies. It first attempts to query GPT-4o. If it fails (e.g., due to timeout or quota), it automatically falls back to Claude Opus 4. Responses are returned in Arabic.
* **TTS (Text-to-Speech)**  
  Converts generated Arabic text back to speech using gTTS. This creates an audio file (response.wav) to be played back to the user.
* **Web Interface (Gradio)**  
  Presents a simple UI for users to speak, listen, and interact with the assistant. Also manages the audio input and output processes seamlessly.

**Data Flow**

1. Voice Input is recorded and saved.
2. ASR Module converts voice to text.
3. Text is passed to the prompt builder.
4. LLM Module generates a helpful, culturally sensitive response.
5. TTS Module synthesizes response back to speech.
6. Web Interface plays back the final output.
7. All interactions are logged for review, performance metrics, and future improvement.

**Model Integration**

* GPT-4o is queried first using OpenAI's Python API.
* If GPT-4o fails (e.g., timeout), Claude Opus 4 is triggered via Anthropic’s API.
* Both models are accessed using stored API keys in .env.
* A configurable timeout (e.g., 8 seconds) is used to decide fallback timing.

**Storage and Logging**

* Audio files are stored in data/samples/.
* Logs are saved in JSON format under data/log/, recording:
  + Input transcriptions
  + Model used
  + Final response
  + Latency
  + Timestamp

**Deployment Stack**

|  |  |
| --- | --- |
| **Component** | **Technology** |
| Language Model | GPT-4o + Claude Opus 4 |
| ASR | FasterWhisper |
| TTS | gTTS |
| Interface | Gradio (Python) |
| Audio Capture | sounddevice & Gradio |
| Logging | JSON |
| Hosting | Localhost for demo |
| Env Config | dotenv, .env files |