📘 Full Project Documentation

Real Estate AI Voice Assistant

👥 Team Name: Wifi

1️⃣ Approach to the Problem

🔍 Problem Statement

Real estate businesses handle a high volume of customer inquiries daily. Traditional support systems face challenges such as:

- Delayed responses
- Inconsistent lead qualification
- Lack of structured information capture
- No automated conversation intelligence
- Manual post-call reporting

The objective of this project was to design an AI-driven conversational system that:

- Conducts structured real-time voice conversations
- Qualifies leads intelligently
- Extracts structured business information
- Tracks conversation progression
- Generates executive-grade Minutes of Meeting (MoM)

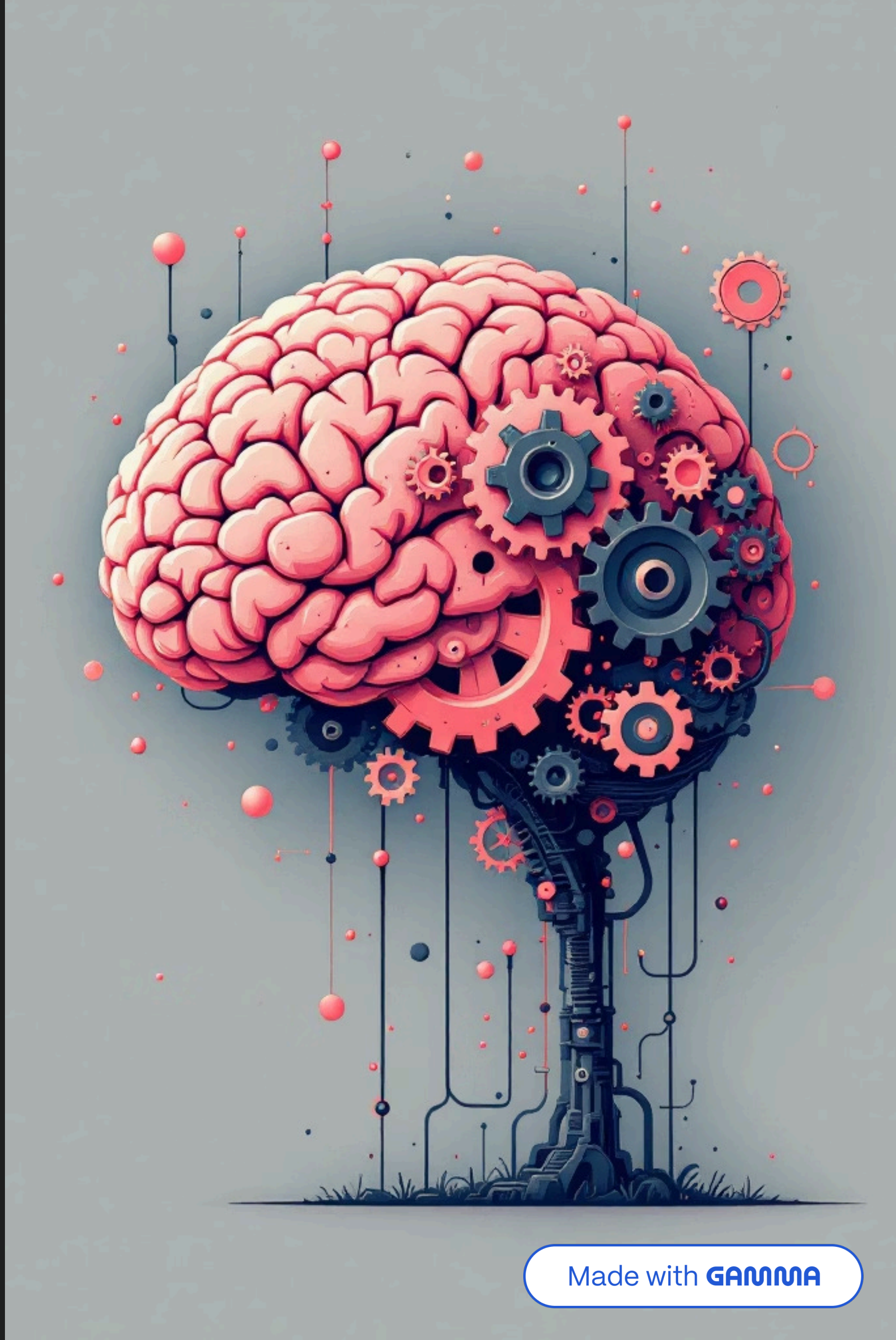
The system is designed as a **Sales Intelligence Engine**, not just a chatbot.

🧠 Core Design Philosophy

Instead of creating a simple Q&A chatbot, the assistant was designed to simulate a trained real estate executive by:

- Understanding customer intent
- Identifying missing qualification data
- Progressively upgrading lead stage
- Handling objections logically
- Maintaining CRM-style structured memory
- Generating business analytics after call completion

The system emphasizes structured reasoning and memory management over free-form conversation.



2️⃣ Justification for Technical Selection

🔹 Faster-Whisper (Speech-to-Text)

Selected because:

- High transcription accuracy
- Optimized performance
- Low latency for conversational speech
- Local processing (reduces cloud dependency)

It ensures reliable speech recognition for real-time interaction.

🔹 Voice Activity Detection (VAD)

VAD enables:

- Efficient speech detection
- Natural turn-taking
- Reduced processing of silence
- Improved interruption handling

Without VAD, the system would process silence and increase latency.

🔹 Gemini API (Reasoning Engine)

Gemini was selected for:

- Strong contextual reasoning
- Reliable structured output formatting
- Controlled JSON responses
- Balanced cost-performance ratio

Instead of using multiple LLM agents, a single structured reasoning agent was implemented to:

- Reduce latency
- Minimize API calls
- Maintain consistent logic
- Produce strict JSON output

🔹 Piper (Text-to-Speech)

Chosen because:

- Local inference capability
- Low latency
- Lightweight execution
- Supports interruption stopping
- No cloud dependency

This improves system responsiveness and cost efficiency.

🔹 FastAPI

Used to:

- Expose AI engine as an API
- Enable microservice-style architecture
- Provide automatic API documentation
- Support scalable backend deployment

3️⃣ Advantages Over Alternative Solutions

🆚 Compared to Rule-Based Systems

Rule-based chatbots:

- Break under unpredictable input
- Cannot adapt dynamically
- Lack semantic understanding

This system:

- Uses AI reasoning
- Understands semantic context
- Dynamically adapts conversation flow

🆚 Compared to Basic LLM Chatbots

Simple LLM chatbots:

- Do not maintain structured CRM memory
- Lack lead stage tracking
- Provide generic responses

This system:

- Maintains structured conversation state
- Tracks business entities
- Upgrades lead stage dynamically
- Generates executive-level MoM

🆚 Compared to Cloud-Only Voice Assistants

Cloud-based voice systems:

- High operational cost
- Network latency dependency
- Limited customization

This system:

- Uses local STT and TTS
- Minimizes external dependency
- Provides customizable business logic

4️⃣ Technical Implementation Details

🧩 Module Breakdown

1️⃣ Audio Processing Layer

Components:

- Audio Stream Handler
- Voice Activity Detection
- Speech-to-Text Engine
- Text-to-Speech Engine

Workflow:

1. Audio chunk received
2. VAD detects speech activity
3. Speech sent to Whisper
4. Text generated

2️⃣ Intelligence Layer

Single reasoning agent performs:

- Intent detection
- Entity extraction
- Sentiment analysis
- Lead stage classification
- Response generation
- Call termination decision

Returns strict JSON structure:

```
{ "intent": "", "entities": {}, "sentiment": "", "final_response": "", "lead_stage": "", "end_call": false }
```

3️⃣ Session Management

Each session maintains:

- Complete transcript
- Business state (budget, location, configuration)
- Lead stage progression
- Sentiment timeline
- Objections
- Action items

Memory Types Used:

- Structured summary (long-term memory)
- Recent conversation turns (short-term memory)
- Business CRM state
- Current user message

This prevents repetitive responses and maintains logical flow.

4️⃣ Interruption Handling

If the user speaks while AI is responding:

- Text-to-speech stops immediately
- System switches to listening mode
- Conversation continues naturally

This ensures realistic conversational behavior.

5️⃣ High-Level Architectural Design

End-to-End Workflow

🔄 System Flow

1. Assistant greets user.
2. User speaks.
3. VAD detects speech.
4. Whisper converts speech to text.
5. Context builder prepares:
    - Structured summary
    - Recent conversation turns
    - Business state
6. Gemini reasoning agent processes:
    - Intent
    - Entities
    - Sentiment
    - Lead stage
7. Response generated.
8. Piper converts response to speech.
9. Audio played back.
10. Session memory updated.
11. If end_call flag true → MoM generated.

🏗️ Logical Architecture

User Voice

↓

Voice Activity Detection

↓

Faster-Whisper (STT)

↓

Context Builder

↓

Gemini Reasoning Engine

↓

Lead Stage Logic

↓

Response Generation

↓

Piper TTS

↓

User Hears Response

↓

MoM Generator (Post-call)

🎯 System Strengths

- Real-time conversational intelligence
- Structured lead qualification
- CRM-style memory management
- Interruption-aware interaction
- Executive-grade reporting
- Scalable backend architecture

📌 Conclusion

The Real Estate AI Voice Assistant is a structured conversational intelligence system designed to simulate a professional real estate executive.

It integrates speech processing, AI reasoning, session management, and business analytics into a unified workflow.

The project demonstrates:

- Applied AI engineering
- Structured prompt design
- Real-time systems architecture
- Business-focused conversational modeling
- Scalable service-oriented backend design

This is not a basic chatbot implementation, but a structured AI-powered sales intelligence engine.