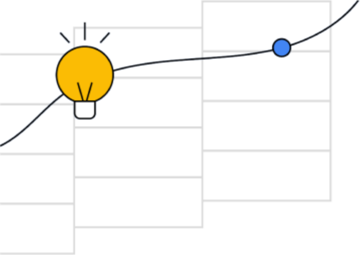
Guidelines

- Kindly use the given template for submitting your project (Make a copy of the template).
- One team is only required to submit one project.
- The ideal size of the presentation should not be more than 10-12 slides.
- You are welcome to add as many POCs and design concepts to support your project.
- It is mandatory to use Gemini APIs while building your prototype.
- In case of queries, kindly reach out to us at apacsolutionchallenge@hack2skill.com

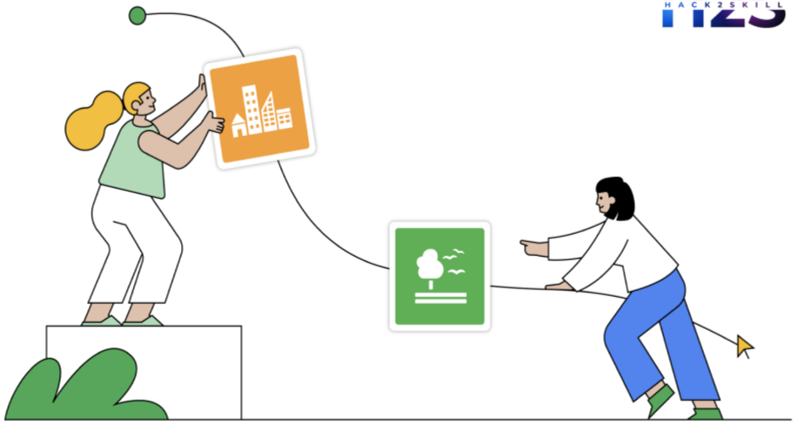


Google Developer Groups | ADB

Powered by H2S

Solution Challenge

AI for a better tomorrow



Team Details

a. **Team name:** onemanarmy
b. **Team leader name:** Trần Võ Minh Hùng
c. **Problem Statement:** The problem is to design and implement an AI assistant that can effectively understand and process multimodal user inputs (text, voice, images) related to plant health issues, utilize different LLM architectures (local Gemma and cloud Gemini) for analysis and response generation, and maintain conversational context through summarization and RAG to provide users with relevant preliminary diagnoses and actionable recommendations.

Brief about your solution

Solution Name: "AI-Powered Plant Health Assistant,"
Brief: This project is an AI-powered, multimodal chatbot assistant designed to help users identify potential plant health issues. Users can interact with the assistant via text or voice and can upload images of affected plants. The system leverages Large Language Models (LLMs) – specifically a local Gemma-3-4B-IT model and the cloud-based Gemini 2.0 Flash API – to understand user queries, analyze image descriptions, provide preliminary diagnoses, and offer actionable advice. Key features include session management, conversation history, image analysis, voice-to-text, text-to-speech, and a RAG (Retrieval Augmented Generation) system for contextually relevant information retrieval from past conversation summaries and image descriptions. The system architecture allows for dynamic switching between different LLMs for response generation and includes an "evaluation mode" to compare or enhance responses using a more powerful model (Gemini).

Opportunities

How different is it from any of the other existing ideas?

Multimodal & Multi-LLM Flexibility: While many plant disease apps exist, this solution uniquely integrates multiple LLMs (local Gemma and cloud Gemini) and provides a framework to switch between them or use them in tandem (e.g., Gemini for evaluation). This allows for experimentation and leveraging the strengths of different models. The multimodal input (text, voice, image) is comprehensive.

Conversational & Iterative Diagnosis: Unlike static image lookup tools or simple Q&A bots, this system engages in an iterative diagnostic conversation, asking clarifying questions based on initial input (including AI-generated image descriptions) to gather more details, mimicking a consultation with an expert.

Integrated RAG and Summarization: The continuous summarization of the conversation and embedding of key information (including image descriptions) into a vector database for RAG ensures that the LLM has relevant, session-specific long-term memory, leading to more coherent and context-aware interactions over extended conversations.

Developer-Focused Features (Toggles): The inclusion of toggles for switching LLMs and activating an "evaluation mode" makes it a powerful platform for research, development, and comparative analysis of LLM performance in this specific domain, which is not typically a feature of end-user plant apps.

Opportunities

How will it be able to solve the problem?

Accessibility: Provides users (e.g., home gardeners, small-scale farmers) with an accessible first line of support for plant health concerns, available 24/7.

Information Gathering: Systematically guides users to provide relevant information (plant type, symptoms, environment) that is crucial for diagnosis, which users might not initially know to offer.

Preliminary Analysis: Offers potential causes for plant issues based on the provided information, helping users narrow down possibilities.

Actionable Advice: Suggests immediate, safe management steps and preventative measures.

Reduces Diagnostic Uncertainty: While not a replacement for lab tests, it helps users understand potential issues better and make more informed decisions about plant care or when to seek professional help.

Opportunities

USP of the proposed solution

Adaptive AI Core: The ability to switch and evaluate different LLM backbones (Gemma vs. Gemini) for the core diagnostic task.

Contextual Conversational Diagnosis: Deeply contextual conversations powered by RAG over session summaries and image descriptions.

Multimodal Interaction: Seamless handling of text, voice, and image inputs for a richer user experience.

Developer/Researcher-Friendly: Built-in toggles and evaluation mechanisms facilitate ongoing improvement and experimentation with AI models.

List of features offered by the solution

User Interaction:

+ Text-based chat interface.
+ Voice input (Speech-to-Text via Whisper-small).
+ Text-to-Speech output (via Kokoro TTS).
+ Image upload capability for visual diagnosis.
+ Live camera capture for image input.

AI Capabilities:

+ Primary Response Generation:
+ Using local Gemma-3-4B-IT model.
+ Option to switch to Google's Gemini 2.0 Flash API.
+ Image Description Generation: AI (Gemma, or potentially Gemini if main is Gemini) describes uploaded images to extract visual symptoms.
+ Clarification Question Generation: AI asks targeted questions to gather more diagnostic details.
+ Preliminary Diagnosis & Advice: AI suggests potential plant issues and management strategies.
+ Conversational Context & History Management:
  ++Stores and retrieves chat session history (SQLite).
  ++Maintains short-term memory for LLM context.

.

List of features offered by the solution

+RAG (Retrieval Augmented Generation):
  ++Embeds user messages, AI responses, and image descriptions into a vector database (ChromaDB) per session.
  ++Retrieves relevant past conversation snippets and image context to augment prompts for the LLM.
+ Conversation Summarization:
  ++Periodically summarizes the ongoing conversation using Qwen2.5-0.5B-Instruct with a structured template.
  ++Uses this summary as part of the RAG context.
+ "Eval Mode":
  ++Allows triggering an additional, enhanced response from Gemini based on the current conversation summary and latest image, for comparison or improved insights.
+ Session Management (Frontend & Backend):
  ++Create, rename, delete, and switch between chat sessions.
  ++Persistence of sessions and messages.
+ Backend & System:
  ++WebSocket server for real-time communication.
  ++Asynchronous task handling.
  ++Persistent storage for API keys (via settings.json).
  ++Developer toggles for switching LLM and activating evaluation mode.

Process flow diagram - a link to view:
https://drive.google.com/file/d/1M6jVVndR50O_sY4-gxjuyC9pPsAa9dl9/view?usp=sharing

Process flow diagram or Use-case diagram

1. User Interaction (Frontend):

+ The user interacts with the React frontend (e.g., MainChatView.tsx).

+ They can type a text message, upload an image file, or capture an image using the camera. Voice input is also possible, which is first transcribed to text by Whisper on the backend.

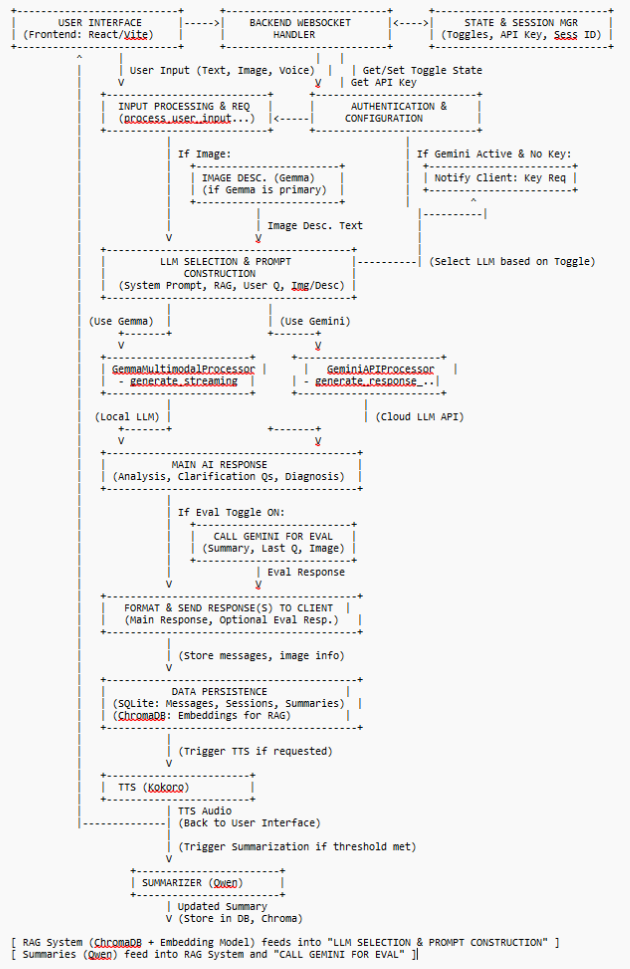
+ The frontend client also manages UI elements for toggles (Gemini, Eval, Grounding) and an API key input mechanism.

2. Input Transmission (Frontend to Backend):

+ When the user sends a message (potentially with an image) or changes a toggle state, or submits an API key, the frontend formats a JSON message.

+ This message is sent over a WebSocket connection (managed by WebSocketProvider.tsx) to the Python backend server.

+ Relevant sessionId, message type (e.g., text_input, update_toggle_state, set_api_key), and data (text, image base64, toggle name/state, API key) are included.

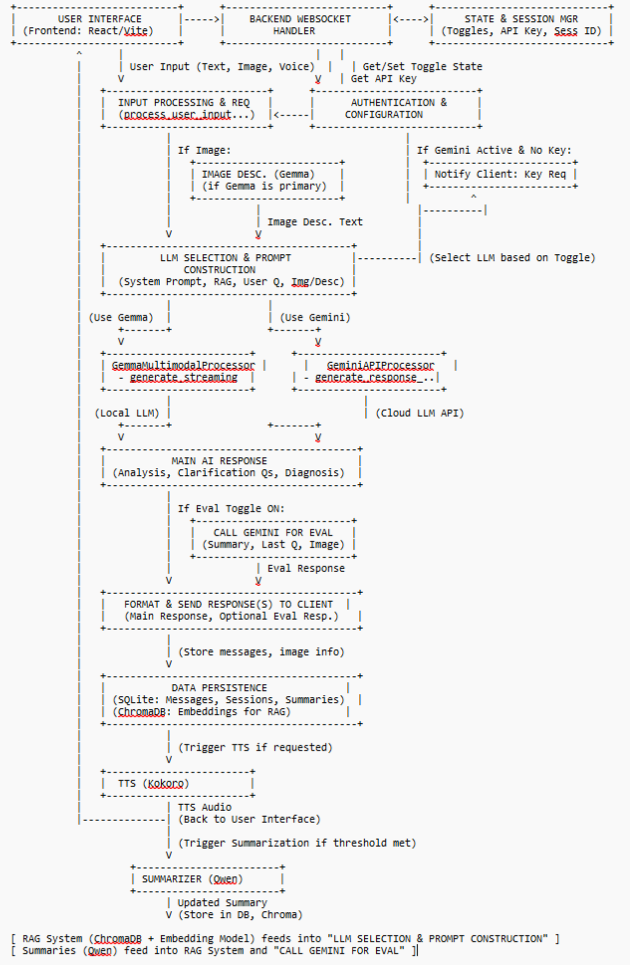


```
+---------------------------+      +---------------------------+      +-----------------------------+
|       USER INTERFACE      |----->|    BACKEND WEBSOCKET      |<---->|     STATE & SESSION MGR     |
| (Frontend: React/Vite)    |      |        HANDLER            |      | (Toggles, API Key, Sess ID) |
+---------------------------+      +---------------------------+      +-----------------------------+
           ^      |                                |   |
           |      | User Input (Text, Image, Voice)  |   | Get/Set Toggle State
           |      V                                V   | Get API Key
           |   +---------------------------+      +---------------------------+
           |   |  INPUT PROCESSING & REQ   |      |      AUTHENTICATION &     |
           |   |  (process_user_input...)  |<-----|     CONFIGURATION         |
           |   +---------------------------+      +---------------------------+
           |              |                                       |
           |              | If Image:                             | If Gemini Active & No Key:
           |              |   +------------------------+          |  +------------------------+
           |              |   | IMAGE DESC. (Gemma)    |          |  | Notify Client: Key Req |
           |              |   | (if Gemma is primary)  |          |  +------------------------+
           |              |   +------------------------+          |          ^
           |              |              |                          |----------|
           |              |              | Image Desc. Text         |
           |              V              V                          |
           |   +-----------------------------------------+          |
           |   |         LLM SELECTION & PROMPT          |----------| (Select LLM based on Toggle)
           |   |             CONSTRUCTION                |
           |   |  (System Prompt, RAG, User Q, Img/Desc) |
           |   +-----------------------------------------+
           |              |                |
           | (Use Gemma)  |                | (Use Gemini)
           |      +-------+                +-------+
           |      V                                V
           |   +------------------------+      +------------------------+
           |   | GemmaMultimodalProcessor |      |   GeminiAPIProcessor   |
           |   |  - generate_streaming  |      | - generate_response_..|
           |   +------------------------+      +------------------------+
           |              |                                |
           |  (Local LLM) |                                | (Cloud LLM API)
           |      +-------+                +-------+
           |      V                                V
           |   +------------------------------------------+
           |   |           MAIN AI RESPONSE               |
           |   | (Analysis, Clarification Qs, Diagnosis)  |
           |   +------------------------------------------+
           |              |
           |              | If Eval Toggle ON:
           |              |   +--------------------------+
           |              |   |   CALL GEMINI FOR EVAL   |
           |              |   | (Summary, Last Q, Image) |
           |              |   +--------------------------+
           |              |              | Eval Response
           |              V              V
           |   +------------------------------------------+
           |   |   FORMAT & SEND RESPONSE(S) TO CLIENT  |
           |   |   (Main Response, Optional Eval Resp.)   |
           |   +------------------------------------------+
           |              |
           |              | (Store messages, image info)
           |              V
           |   +------------------------------------------+
           |   |           DATA PERSISTENCE             |
           |   | (SQLite: Messages, Sessions, Summaries)  |
           |   | (ChromaDB: Embeddings for RAG)         |
           |   +------------------------------------------+
           |              |
           |              | (Trigger TTS if requested)
           |              V
           |   +------------------------+
           |   |  TTS (Kokoro)          |
           |   +------------------------+
           |              | TTS Audio
           |--------------| (Back to User Interface)
                          |
                          | (Trigger Summarization if threshold met)
                          V
                    +------------------------+
                    | SUMMARIZER (Qwen)      |
                    +------------------------+
                          | Updated Summary
                          V (Store in DB, Chroma)

[ RAG System (ChromaDB + Embedding Model) feeds into "LLM SELECTION & PROMPT CONSTRUCTION" ]
[ Summaries (Qwen) feed into RAG System and "CALL GEMINI FOR EVAL" ]
```

Process flow diagram or Use-case diagram

3. Backend Reception & Initial Handling (handle_client & receive_data_from_client):

+ The backend's handle_client function manages the WebSocket connection for each client.
+ The nested receive_data_from_client task listens for incoming messages.
+ Session Management: The handler identifies or establishes the active_client_session_id.
+ Toggle/API Key State Update: If the message is update_toggle_state or set_api_key:
  ++The backend updates its global state variables (e.g., GLOBAL_USE_GEMINI_MODEL, GEMINI_API_KEY_STORE).
  ++The API key is saved persistently (e.g., to settings.json).
  ++An acknowledgment (..._ack) is sent back to the client. If enabling a Gemini feature without a key, an error/prompt for key is sent.
  ++If an API key is set, the gemini_processor_instance for that client handler is initialized.



```
+---------------------------+      +---------------------------+      +-----------------------------+
|       USER INTERFACE      |----->|    BACKEND WEBSOCKET      |<---->|     STATE & SESSION MGR     |
| (Frontend: React/Vite)    |      |        HANDLER            |      | (Toggles, API Key, Sess ID) |
+---------------------------+      +---------------------------+      +-----------------------------+
           ^      |                                |   |
           |      | User Input (Text, Image, Voice) |   | Get/Set Toggle State
           |      V                                V   | Get API Key
           |   +---------------------------+      +---------------------------+
           |   |  INPUT PROCESSING & REQ   |      |      AUTHENTICATION &     |
           |   |  (process_user_input...)  |<-----|     CONFIGURATION         |
           |   +---------------------------+      +---------------------------+
           |              |                                       |
           |              | If Image:                             | If Gemini Active & No Key:
           |              |   +------------------------+          |  +------------------------+
           |              |   | IMAGE DESC. (Gemma)    |          |  | Notify Client: Key Req |
           |              |   | (if Gemma is primary)  |          |  +------------------------+
           |              |   +------------------------+          |          ^
           |              |              |                          |----------|
           |              |              | Image Desc. Text         |
           |              V              V                          |
           |   +-----------------------------------------+          |
           |   |         LLM SELECTION & PROMPT          |----------| (Select LLM based on Toggle)
           |   |             CONSTRUCTION                |
           |   |  (System Prompt, RAG, User Q, Img/Desc) |
           |   +-----------------------------------------+
           |              |                |
           | (Use Gemma)  |                | (Use Gemini)
           |      +-------+                +-------+
           |      V                                V
           |   +------------------------+      +------------------------+
           |   | GemmaMultimodalProcessor |      |   GeminiAPIProcessor   |
           |   |  - generate_streaming  |      | - generate_response_..|
           |   +------------------------+      +------------------------+
           |              |                                |
           |  (Local LLM) |                                | (Cloud LLM API)
           |      +-------+                +-------+
           |      V                                V
           |   +------------------------------------------+
           |   |           MAIN AI RESPONSE               |
           |   | (Analysis, Clarification Qs, Diagnosis)  |
           |   +------------------------------------------+
           |              |
           |              | If Eval Toggle ON:
           |              |   +--------------------------+
           |              |   |   CALL GEMINI FOR EVAL   |
           |              |   | (Summary, Last Q, Image) |
           |              |   +--------------------------+
           |              |              | Eval Response
           |              V              V
           |   +------------------------------------------+
           |   |   FORMAT & SEND RESPONSE(S) TO CLIENT  |
           |   |   (Main Response, Optional Eval Resp.)   |
           |   +------------------------------------------+
           |              |
           |              | (Store messages, image info)
           |              V
           |   +------------------------------------------+
           |   |           DATA PERSISTENCE             |
           |   | (SQLite: Messages, Sessions, Summaries)  |
           |   | (ChromaDB: Embeddings for RAG)         |
           |   +------------------------------------------+
           |              |
           |              | (Trigger TTS if requested)
           |              V
           |   +------------------------+
           |   |  TTS (Kokoro)          |
           |   +------------------------+
           |              | TTS Audio
           |--------------| (Back to User Interface)
                          |
                          | (Trigger Summarization if threshold met)
                          V
                    +------------------------+
                    | SUMMARIZER (Qwen)      |
                    +------------------------+
                          | Updated Summary
                          V (Store in DB, Chroma)

[ RAG System (ChromaDB + Embedding Model) feeds into "LLM SELECTION & PROMPT CONSTRUCTION" ]
[ Summaries (Qwen) feed into RAG System and "CALL GEMINI FOR EVAL" ]
```

Process flow diagram or Use-case diagram

4. Main Input Processing (process_user_input_and_respond):

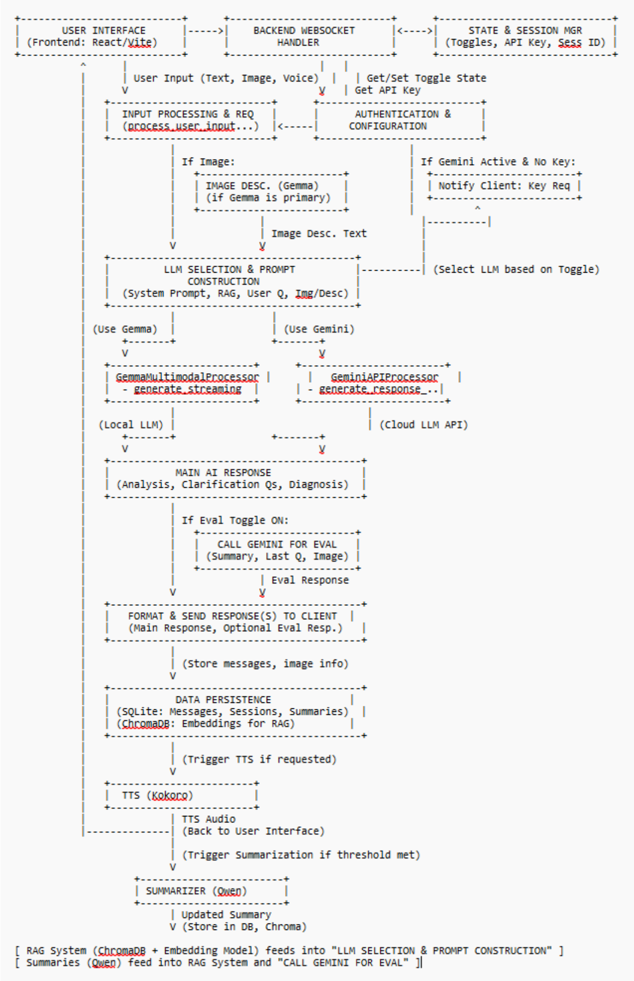
+ For a text_input message:

++The user's message ID is generated, and text/image data is extracted.

++If an image is present, it's decoded. current_image_bytes_for_gemini and current_image_pil_for_turn (for Gemma) are prepared. The image file is saved to media_storage.

++The user's message (text and image filename reference) is saved to the SQLite database (messages table).

++Gemma's short-term history is updated with the user's turn (this aids RAG even if Gemini is used later).



```
+---------------------------+      +---------------------------+      +-----------------------------+
|       USER INTERFACE      |----->|    BACKEND WEBSOCKET      |<---->|     STATE & SESSION MGR     |
| (Frontend: React/Vite)    |      |        HANDLER            |      | (Toggles, API Key, Sess ID) |
+---------------------------+      +---------------------------+      +-----------------------------+
           ^      |                                |   |
           |      | User Input (Text, Image, Voice)  |   | Get/Set Toggle State
           |      V                                V   | Get API Key
           |   +---------------------------+      +---------------------------+
           |   |  INPUT PROCESSING & REQ   |      |      AUTHENTICATION &     |
           |   |  (process_user_input...)  |<-----|     CONFIGURATION         |
           |   +---------------------------+      +---------------------------+
           |              |                                       |
           |              | If Image:                             | If Gemini Active & No Key:
           |              |   +------------------------+          |  +-------------------------+
           |              |   | IMAGE DESC. (Gemma)    |          |  | Notify Client: Key Req  |
           |              |   | (if Gemma is primary)  |          |  +-------------------------+
           |              |   +------------------------+          |          ^
           |              |              |                          |----------|
           |              |              | Image Desc. Text         |
           |              V              V                          |
           |   +-----------------------------------------+          |
           |   |         LLM SELECTION & PROMPT          |----------| (Select LLM based on Toggle)
           |   |             CONSTRUCTION                |
           |   |  (System Prompt, RAG, User Q, Img/Desc) |
           |   +-----------------------------------------+
           |              |                |
           | (Use Gemma)  |                | (Use Gemini)
           |      +-------+                +-------+
           |      V                                V
           |   +------------------------+      +------------------------+
           |   | GemmaMultimodalProcessor |      |   GeminiAPIProcessor   |
           |   |  - generate_streaming  |      | - generate_response_..|
           |   +------------------------+      +------------------------+
           |              |                                |
           |  (Local LLM) |                                | (Cloud LLM API)
           |      +-------+                +-------+
           |      V                                V
           |   +------------------------------------------+
           |   |           MAIN AI RESPONSE               |
           |   | (Analysis, Clarification Qs, Diagnosis)  |
           |   +------------------------------------------+
           |              |
           |              | If Eval Toggle ON:
           |              |   +--------------------------+
           |              |   |   CALL GEMINI FOR EVAL   |
           |              |   | (Summary, Last Q, Image) |
           |              |   +--------------------------+
           |              |              | Eval Response
           |              V              V
           |   +------------------------------------------+
           |   |   FORMAT & SEND RESPONSE(S) TO CLIENT  |
           |   |   (Main Response, Optional Eval Resp.)   |
           |   +------------------------------------------+
           |              |
           |              | (Store messages, image info)
           |              V
           |   +------------------------------------------+
           |   |           DATA PERSISTENCE             |
           |   | (SQLite: Messages, Sessions, Summaries)  |
           |   | (ChromaDB: Embeddings for RAG)         |
           |   +------------------------------------------+
           |              |
           |              | (Trigger TTS if requested)
           |              V
           |   +------------------------+
           |   |  TTS (Kokoro)          |
           |   +------------------------+
           |              | TTS Audio
           |--------------| (Back to User Interface)
                          |
                          | (Trigger Summarization if threshold met)
                          V
                    +------------------------+
                    | SUMMARIZER (Qwen)      |
                    +------------------------+
                          | Updated Summary
                          V (Store in DB, Chroma)

[ RAG System (ChromaDB + Embedding Model) feeds into "LLM SELECTION & PROMPT CONSTRUCTION" ]
[ Summaries (Qwen) feed into RAG System and "CALL GEMINI FOR EVAL" ]
```

Process flow diagram or Use-case diagram

5. LLM Selection & Pre-processing:

+ The function checks the GLOBAL_USE_GEMINI_MODEL toggle and if a gemini_processor instance is available (i.e., API key is set and processor initialized).

+ If Gemini Path:

++active_llm_service is set to "Gemini".

++RAG context (existing conversation summary from Qwen + relevant snippets from ChromaDB) is retrieved via gemma_processor._retrieve_rag_context().

++The detailed GEMINI_CHAT_SYSTEM_PROMPT_FOR_MAIN_RESPONSE is prepared.

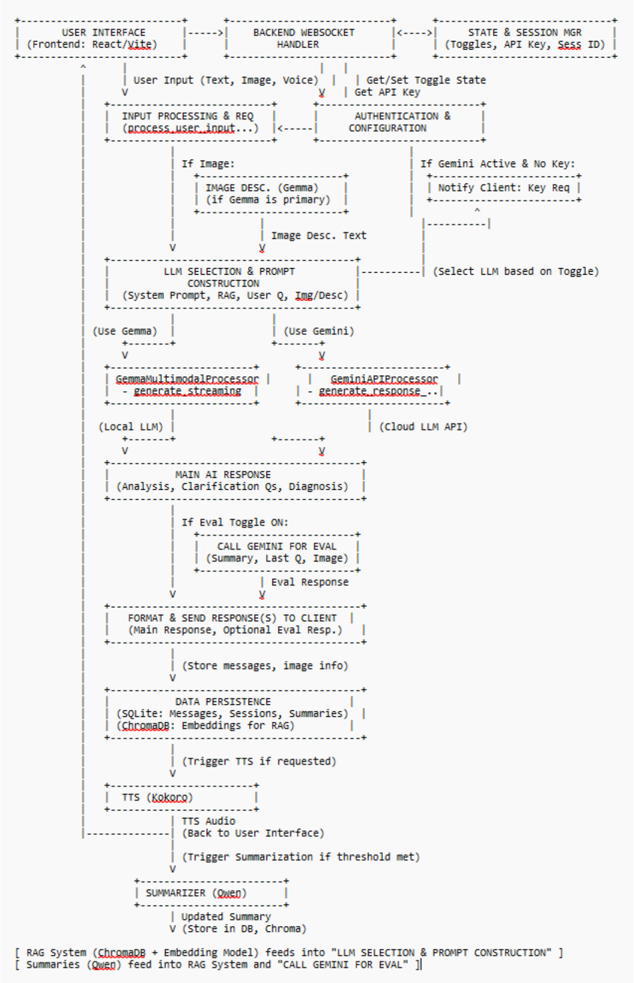
+ If Gemma Path (Fallback or Default):

++active_llm_service is set to "Gemma".

++Gemma Image Description (if image present): gemma_processor.describe_image() is called. The generated description is sent to the client immediately, saved to the DB, and embedded into ChromaDB. This description (image_description_for_gemma_context) is then available for Gemma's main diagnostic step.

++RAG context is retrieved (implicitly within gemma_processor.generate_streaming which calls _retrieve_rag_context).

++The REVISED_SIMPLIFIED_SYSTEM_PROMPT is used.

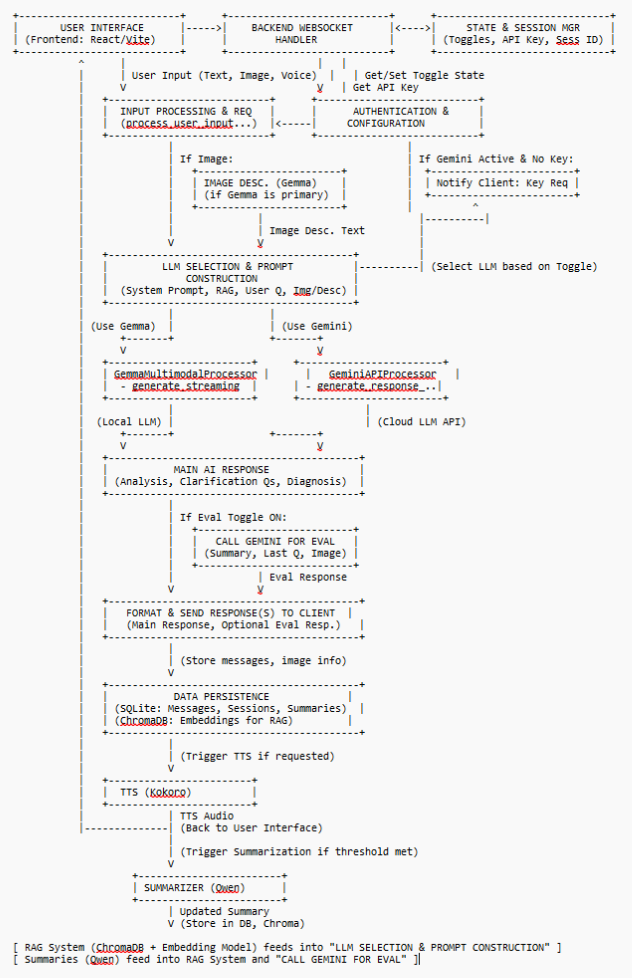


```
+---------------------------+      +---------------------------+      +-----------------------------+
|       USER INTERFACE      |----->|    BACKEND WEBSOCKET      |<---->|     STATE & SESSION MGR     |
| (Frontend: React/Vite)    |      |        HANDLER            |      | (Toggles, API Key, Sess ID) |
+---------------------------+      +---------------------------+      +-----------------------------+
           ^      |                                |   |
           |      | User Input (Text, Image, Voice) |   | Get/Set Toggle State
           |      V                                V   | Get API Key
           |   +---------------------------+      +---------------------------+
           |   |  INPUT PROCESSING & REQ   |      |      AUTHENTICATION &     |
           |   |  (process_user_input...)  |<-----|     CONFIGURATION         |
           |   +---------------------------+      +---------------------------+
           |              |                                       |
           |              | If Image:                             | If Gemini Active & No Key:
           |              |   +------------------------+          |  +------------------------+
           |              |   | IMAGE DESC. (Gemma)    |          |  | Notify Client: Key Req |
           |              |   | (if Gemma is primary)  |          |  +------------------------+
           |              |   +------------------------+          |          ^
           |              |              |                          |----------|
           |              |              | Image Desc. Text         |
           |              V              V                          |
           |   +-----------------------------------------+          |
           |   |         LLM SELECTION & PROMPT          |----------| (Select LLM based on Toggle)
           |   |             CONSTRUCTION                |
           |   |  (System Prompt, RAG, User Q, Img/Desc) |
           |   +-----------------------------------------+
           |              |                |
           | (Use Gemma)  |                | (Use Gemini)
           |      +-------+                +-------+
           |      V                                V
           |   +------------------------+      +------------------------+
           |   | GemmaMultimodalProcessor |      |  GeminiAPIProcessor    |
           |   |  - generate_streaming  |      | - generate_response_..|
           |   +------------------------+      +------------------------+
           |              |                                |
           |  (Local LLM) |                                | (Cloud LLM API)
           |      +-------+                +-------+
           |      V                                V
           |   +------------------------------------------+
           |   |          MAIN AI RESPONSE                |
           |   | (Analysis, Clarification Qs, Diagnosis)  |
           |   +------------------------------------------+
           |              |
           |              | If Eval Toggle ON:
           |              |   +--------------------------+
           |              |   |  CALL GEMINI FOR EVAL    |
           |              |   | (Summary, Last Q, Image) |
           |              |   +--------------------------+
           |              |              | Eval Response
           |              V              V
           |   +------------------------------------------+
           |   |   FORMAT & SEND RESPONSE(S) TO CLIENT  |
           |   |   (Main Response, Optional Eval Resp.)   |
           |   +------------------------------------------+
           |              |
           |              | (Store messages, image info)
           |              V
           |   +------------------------------------------+
           |   |           DATA PERSISTENCE             |
           |   | (SQLite: Messages, Sessions, Summaries)  |
           |   | (ChromaDB: Embeddings for RAG)         |
           |   +------------------------------------------+
           |              |
           |              | (Trigger TTS if requested)
           |              V
           |   +------------------------+
           |   |  TTS (Kokoro)          |
           |   +------------------------+
           |              | TTS Audio
           |--------------| (Back to User Interface)
                          |
                          | (Trigger Summarization if threshold met)
                          V
                    +------------------------+
                    | SUMMARIZER (Qwen)      |
                    +------------------------+
                          | Updated Summary
                          V (Store in DB, Chroma)

[ RAG System (ChromaDB + Embedding Model) feeds into "LLM SELECTION & PROMPT CONSTRUCTION" ]
[ Summaries (Qwen) feed into RAG System and "CALL GEMINI FOR EVAL" ]
```

Process flow diagram or Use-case diagram

6. Main AI Response Generation:
+ Gemini Path:
 ++gemini_processor.generate_response_for_chat() is called with the system prompt, user query, RAG context, and direct image bytes (if any). It uses Google Search tool if needed.
+ Gemma Path:
 ++gemma_processor.generate_streaming() is called with the user query, (Gemma-generated) image description, and implicitly uses RAG context and its system prompt.
 + The chosen LLM generates the main AI response (e.g., analysis, clarification questions, preliminary diagnosis).

7. Post-Main Response Processing & Sending:
+ The generated AI response (ai_final_response_text_for_client) is sanitized.
+ The response is sent to the client via WebSocket.
+ The AI response is saved to the SQLite database (messages table), tagged with the active_llm_service.
+ If Gemma was used, its internal short-term history is updated.
+ TTS Generation (Optional): If requested (generate_tts is true), the AI's text response is sent to tts_processor.synthesize_initial_speech() (Kokoro), and the resulting audio is streamed back to the client.



```
+---------------------------+      +---------------------------+      +-----------------------------+
|     USER INTERFACE        |----->|     BACKEND WEBSOCKET     |<---->|     STATE & SESSION MGR     |
| (Frontend: React/Vite)    |      |         HANDLER           |      | (Toggles, API Key, Sess ID) |
+---------------------------+      +---------------------------+      +-----------------------------+
           ^      |                                |   |
           |      | User Input (Text, Image, Voice)  |  | Get/Set Toggle State
           |      V                                V   | Get API Key
           |   +---------------------------+      +---------------------------+
           |   |  INPUT PROCESSING & REQ   |      |   AUTHENTICATION &        |
           |   | (process_user_input...)   |<-----|   CONFIGURATION           |
           |   +---------------------------+      +---------------------------+
           |              |                                       |
           |              | If Image:                             | If Gemini Active & No Key:
           |              |   +------------------------+          | +-----------------------+
           |              |   | IMAGE DESC. (Gemma)    |          | | Notify Client: Key Req |
           |              |   | (if Gemma is primary)  |          | +-----------------------+
           |              |   +------------------------+          |          ^
           |              |              |                          |---------|
           |              |              | Image Desc. Text         |
           |              V              V                          |
           |   +-------------------------------------------+        |
           |   |         LLM SELECTION & PROMPT            |--------| (Select LLM based on Toggle)
           |   |             CONSTRUCTION                  |
           |   |  (System Prompt, RAG, User Q, Img/Desc)   |
           |   +-------------------------------------------+
           |              |                |
           | (Use Gemma)  |                | (Use Gemini)
           |      +-------+                +-------+
           |      V                                V
           |   +------------------------+      +------------------------+
           |   | GemmaMultimodalProcessor |       |   GeminiAPIProcessor   |
           |   |  - generate_streaming  |      | - generate_response_..|
           |   +------------------------+      +------------------------+
           |              |                                |
           |  (Local LLM) |                                | (Cloud LLM API)
           |      +-------+                +-------+
           |      V                                V
           |   +------------------------------------------+
           |   |          MAIN AI RESPONSE                |
           |   | (Analysis, Clarification Qs, Diagnosis)  |
           |   +------------------------------------------+
           |              |
           |              | If Eval Toggle ON:
           |              |   +--------------------------+
           |              |   |   CALL GEMINI FOR EVAL   |
           |              |   | (Summary, Last Q, Image) |
           |              |   +--------------------------+
           |              |              | Eval Response
           |              V              V
           |   +------------------------------------------+
           |   |   FORMAT & SEND RESPONSE(S) TO CLIENT  |
           |   |   (Main Response, Optional Eval Resp.)   |
           |   +------------------------------------------+
           |              |
           |              | (Store messages, image info)
           |              V
           |   +------------------------------------------+
           |   |          DATA PERSISTENCE              |
           |   | (SQLite: Messages, Sessions, Summaries)  |
           |   | (ChromaDB: Embeddings for RAG)         |
           |   +------------------------------------------+
           |              |
           |              | (Trigger TTS if requested)
           |              V
           |   +------------------------+
           |   |  TTS (Kokoro)          |
           |   +------------------------+
           |              | TTS Audio
           |--------------| (Back to User Interface)
                          |
                          | (Trigger Summarization if threshold met)
                          V
                    +------------------------+
                    | SUMMARIZER (Qwen)      |
                    +------------------------+
                          | Updated Summary
                          V (Store in DB, Chroma)

[ RAG System (ChromaDB + Embedding Model) feeds into "LLM SELECTION & PROMPT CONSTRUCTION" ]
[ Summaries (Qwen) feed into RAG System and "CALL GEMINI FOR EVAL" ]
```

Process flow diagram or Use-case diagram

8. Evaluation Mode (If Eval Toggle ON):

+ If GLOBAL_EVAL_MODE_ACTIVE is true and gemini_processor is available:

++The latest conversation summary (from Qwen, via gemma_processor's session state) is retrieved.

++The user's current query (user_text_for_db) and the current turn's image (current_image_bytes_for_gemini) are prepared. gemini_processor.generate_eval_response() is called with the Eval-specific system prompt, summary, query, and image.

++The Eval response (prefixed with <EVAL>) is sent to the client as a distinct message (type: "eval_response", sender: "AI_Evaluator").

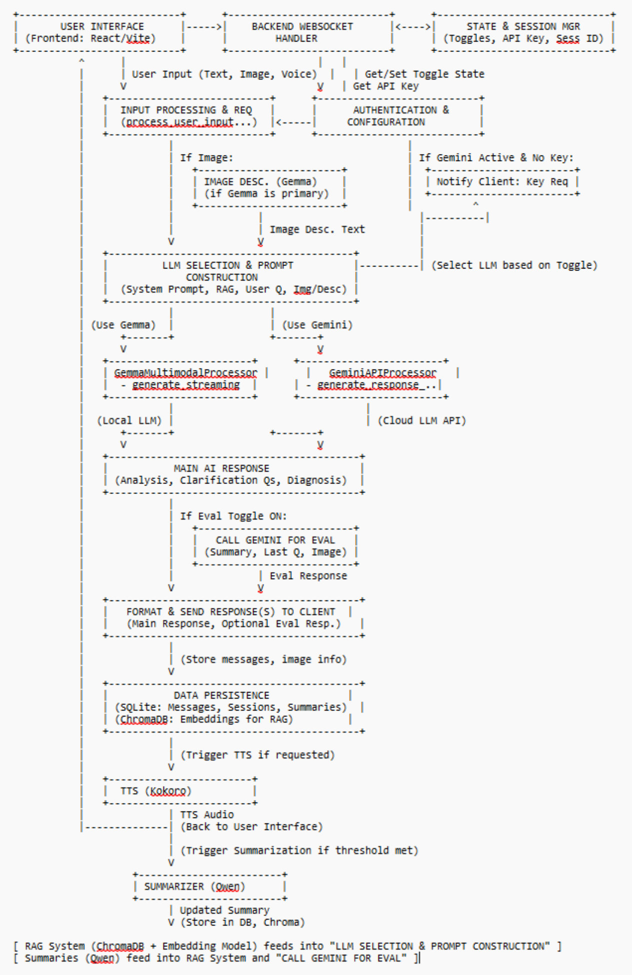
++ The Eval response is saved to the database.

9. Conversation Summarization (Background Trigger):

+ After a certain number of message exchanges (SUMMARY_TRIGGER_THRESHOLD), update_session_summary is called as a background task.

+ Recent conversation turns are fetched from the database. GLOBAL_SUMMARIZER.summarize_text() (Qwen-Instruct) is called with the new turns and the existing running summary.

+ The new, updated, structured summary is stored in the session_summaries SQLite table and gemma_processor's session state, and its embedding is upserted into ChromaDB for future RAG.

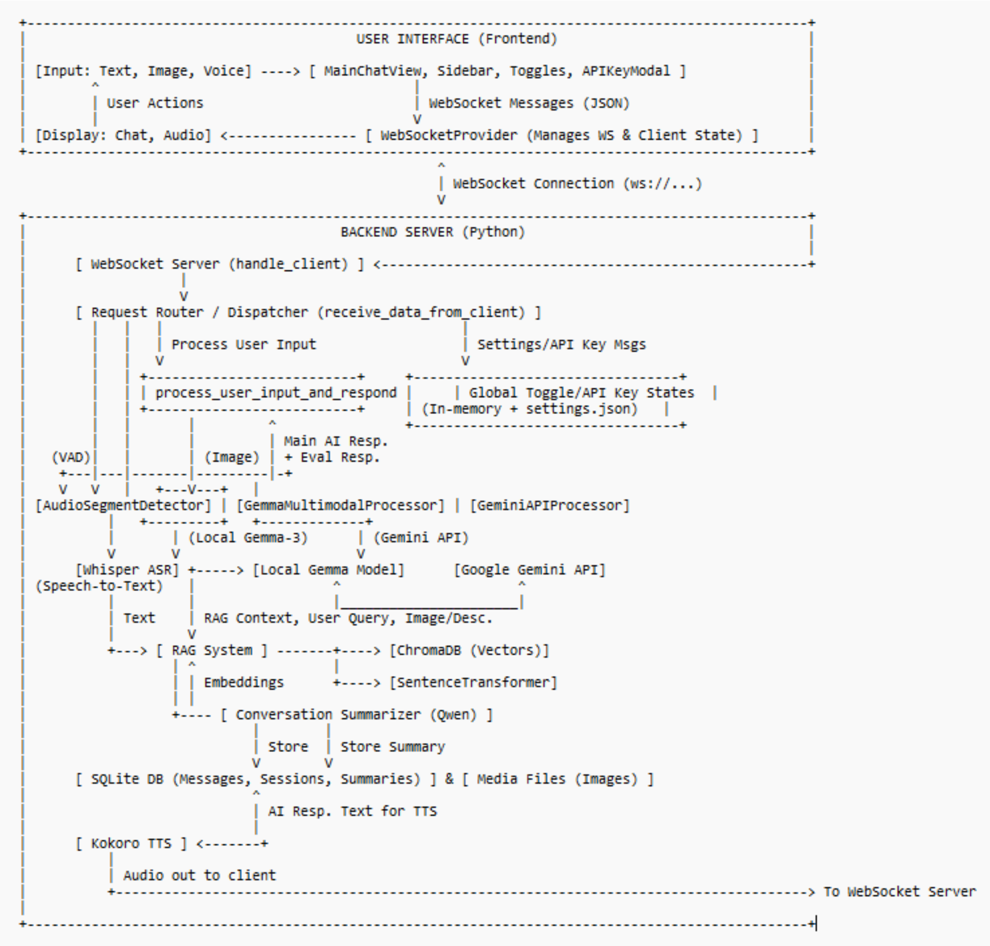


```
+--------------------------+      +--------------------------+      +-----------------------------+
|      USER INTERFACE      |----->|    BACKEND WEBSOCKET     |<---->|     STATE & SESSION MGR     |
| (Frontend: React/Vite)   |      |         HANDLER          |      | (Toggles, API Key, Sess ID) |
+--------------------------+      +--------------------------+      +-----------------------------+
          ^      |                                 |  |
          |      | User Input (Text, Image, Voice) |  | Get/Set Toggle State
          |      V                                 V  | Get API Key
          |   +--------------------------+      +--------------------------+
          |   | INPUT PROCESSING & REQ   |      |     AUTHENTICATION &     |
          |   | (process_user_input...)  |<-----|    CONFIGURATION         |
          |   +--------------------------+      +--------------------------+
          |              |                                     |
          |              | If Image:                           | If Gemini Active & No Key:
          |              |  +------------------------+         |  +------------------------+
          |              |  | IMAGE DESC. (Gemma)    |         |  | Notify Client: Key Req |
          |              |  | (if Gemma is primary)  |         |  +------------------------+
          |              |  +------------------------+         |         ^
          |              |             |                         |----------|
          |              |             | Image Desc. Text        |
          |              V             V                         |
          |   +----------------------------------------+         |
          |   |          LLM SELECTION & PROMPT        |---------| (Select LLM based on Toggle)
          |   |               CONSTRUCTION             |
          |   | (System Prompt, RAG, User Q, Img/Desc) |
          |   +----------------------------------------+
          |              |               |
          | (Use Gemma)  |               | (Use Gemini)
          |      +-------+               +-------+
          |      V                               V
          |   +-----------------------+      +-----------------------+
          |   | GemmaMultimodalProcessor |    |   GeminiAPIProcessor    |
          |   |  - generate_streaming |      | - generate_response_..|
          |   +-----------------------+      +-----------------------+
          |              |                              |
          |  (Local LLM) |                              | (Cloud LLM API)
          |      +-------+               +-------+
          |      V                               V
          |   +-----------------------------------------+
          |   |             MAIN AI RESPONSE            |
          |   | (Analysis, Clarification Qs, Diagnosis) |
          |   +-----------------------------------------+
          |              |
          |              | If Eval Toggle ON:
          |              |  +--------------------------+
          |              |  |   CALL GEMINI FOR EVAL   |
          |              |  | (Summary, Last Q, Image) |
          |              |  +--------------------------+
          |              |             | Eval Response
          |              V             V
          |   +-----------------------------------------+
          |   |   FORMAT & SEND RESPONSE(S) TO CLIENT  |
          |   |   (Main Response, Optional Eval Resp.)  |
          |   +-----------------------------------------+
          |              |
          |              | (Store messages, image info)
          |              V
          |   +-----------------------------------------+
          |   |             DATA PERSISTENCE           |
          |   | (SQLite: Messages, Sessions, Summaries) |
          |   | (ChromaDB: Embeddings for RAG)         |
          |   +-----------------------------------------+
          |              |
          |              | (Trigger TTS if requested)
          |              V
          |   +-----------------------+
          |   |  TTS (Kokoro)         |
          |   +-----------------------+
          |              | TTS Audio
          |--------------| (Back to User Interface)
                         |
                         | (Trigger Summarization if threshold met)
                         V
                   +-----------------------+
                   | SUMMARIZER (Qwen)     |
                   +-----------------------+
                         | Updated Summary
                         V (Store in DB, Chroma)

[ RAG System (ChromaDB + Embedding Model) feeds into "LLM SELECTION & PROMPT CONSTRUCTION" ]
[ Summaries (Qwen) feed into RAG System and "CALL GEMINI FOR EVAL" ]
```

Architecture diagram of the proposed solution - a link to view https://drive.google.com/file/d/1VHYnZHmTXHtP1CIgHvhSr2KucsZ1IWgq/view?usp=sharing

Architecture diagram of the proposed solution

1. User Interface (Top Box - Frontend):
+ The user interacts here, providing input (text, image uploads, camera captures, voice via microphone).
+ UI components like MainChatView, Sidebar, Toggles, and ApiKeyModal manage these inputs.
+ User actions are translated into WebSocket messages (JSON payloads) by the WebSocketProvider.
+ This layer also receives responses (chat messages, audio) from the backend to display to the user.
2. WebSocket Connection (Vertical Line):
+ Represents the persistent, real-time communication channel between the frontend and backend.



```
+--------------------------------------------------------------------------------------------------+
|                                         USER INTERFACE (Frontend)                                |
|                                                                                                  |
| [Input: Text, Image, Voice] ----> [ MainChatView, Sidebar, Toggles, APIKeyModal ]                |
|        ^                                       |                                                 |
|        | User Actions                          | WebSocket Messages (JSON)                       |
|        |                                       V                                                 |
| [Display: Chat, Audio] <---------------- [ WebSocketProvider (Manages WS & Client State) ]       |
+--------------------------------------------------------------------------------------------------+
                                                    ^
                                                    | WebSocket Connection (ws://...)
                                                    V
+--------------------------------------------------------------------------------------------------+
|                                       BACKEND SERVER (Python)                                    |
|                                                                                                  |
|      [ WebSocket Server (handle_client) ] <------------------------------------------------------+
|                   |
|                   V
|      [ Request Router / Dispatcher (receive_data_from_client) ]
|        |   |   |                                     |
|        |   |   | Process User Input                  | Settings/API Key Msgs
|        |   |   V                                     V
|        |   | +--------------------------+     +----------------------------------+
|        |   | | process_user_input_and_respond |     | Global Toggle/API Key States   |
|        |   | +--------------------------+     | (In-memory + settings.json)    |
|        |   |       |         ^                +----------------------------------+
|        |   |       |         | Main AI Resp.
|   (VAD)|   |       | (Image) | + Eval Resp.
|    +---|---|-------|---------|-+
|    V   V   |   +---V---+   |
| [AudioSegmentDetector] | [GemmaMultimodalProcessor] | [GeminiAPIProcessor]
|          |   +---------+   +-------------+
|          |       | (Local Gemma-3)      | (Gemini API)
|          V       V                      V
|      [Whisper ASR] +-----> [Local Gemma Model]      [Google Gemini API]
| (Speech-to-Text)   |                 ^                      ^
|          |         |                 |______________________|
|          | Text    | RAG Context, User Query, Image/Desc.
|          |         V
|          +---> [ RAG System ] -------+----> [ChromaDB (Vectors)]
|                  | ^                 |
|                  | | Embeddings      +----> [SentenceTransformer]
|                  | |
|                  +---- [ Conversation Summarizer (Qwen) ]
|                            |        |
|                            | Store  | Store Summary
|                            V        V
|      [ SQLite DB (Messages, Sessions, Summaries) ] & [ Media Files (Images) ]
|                            ^
|                            | AI Resp. Text for TTS
|                            |
|      [ Kokoro TTS ] <-------+
|          |
|          | Audio out to client
|          +---------------------------------------------------------------------------------------> To WebSocket Server
|
+--------------------------------------------------------------------------------------------------+|
```

Architecture diagram of the proposed solution

3. Backend Server (Bottom Box - Python):

+ WebSocket Server (handle_client): The entry point for incoming WebSocket connections and messages.

+ Request Router (receive_data_from_client): Parses messages.

  ++ If it's a settings message (API key, toggle change), it updates the Global Toggle/API Key States. The API key is also persisted to settings.json.

  ++ If it's user input (text, image, or transcribed voice), it's passed to process_user_input_and_respond.

  ++ Voice data (from realtime_input) goes to AudioSegmentDetector (VAD).

+ VAD & ASR:

  ++ AudioSegmentDetector identifies speech.

  ++ WhisperTranscriber converts speech to text. This text then flows into process_user_input_and_respond.

+ process_user_input_and_respond (Main Logic):

  ++ This is the central orchestrator.

  ++ If an image is provided and Gemma is the primary LLM, it first calls GemmaMultimodalProcessor for an Image Description.

  ++ It then constructs a full prompt including:

    +++A System Prompt (either for Gemma or Gemini).

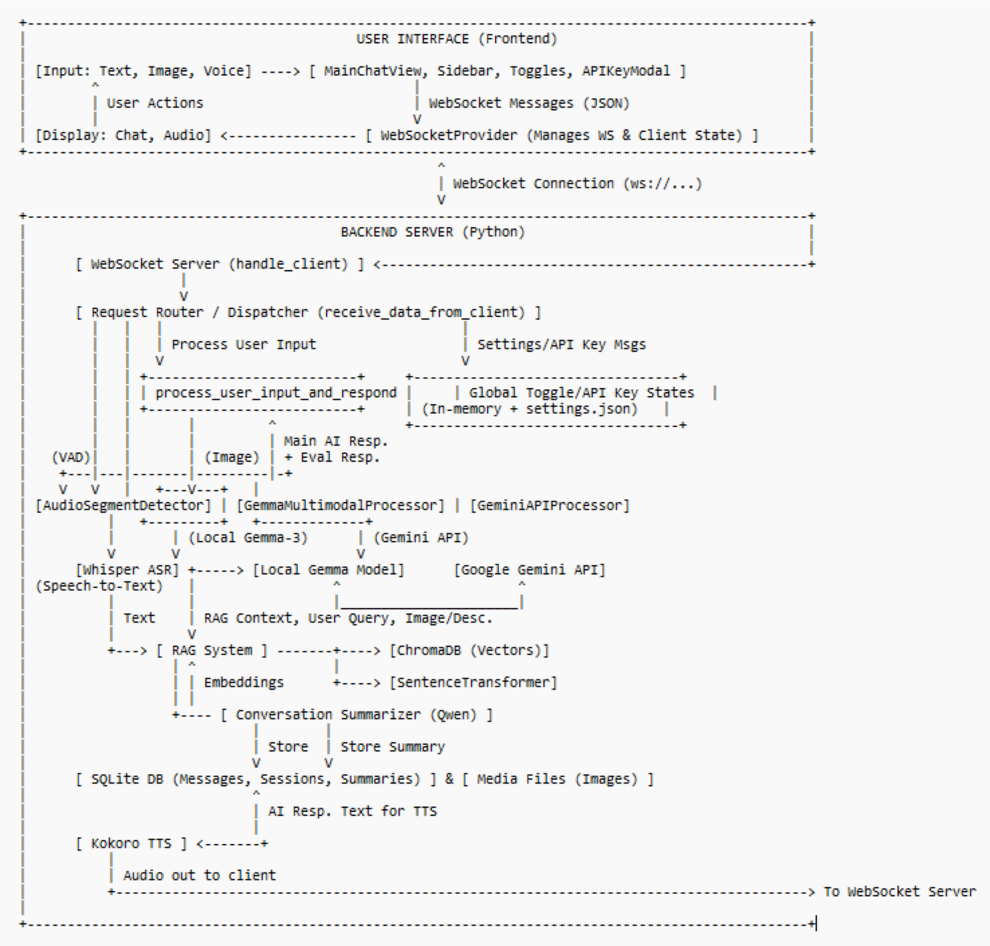
    +++Context from the RAG System.

    +++The User's Query.

    +++The Image itself (for Gemini) or the Image Description text (for Gemma).

  ++ It selects either GemmaMultimodalProcessor or GeminiAPIProcessor based on the active toggle and API key availability.

(cont.)

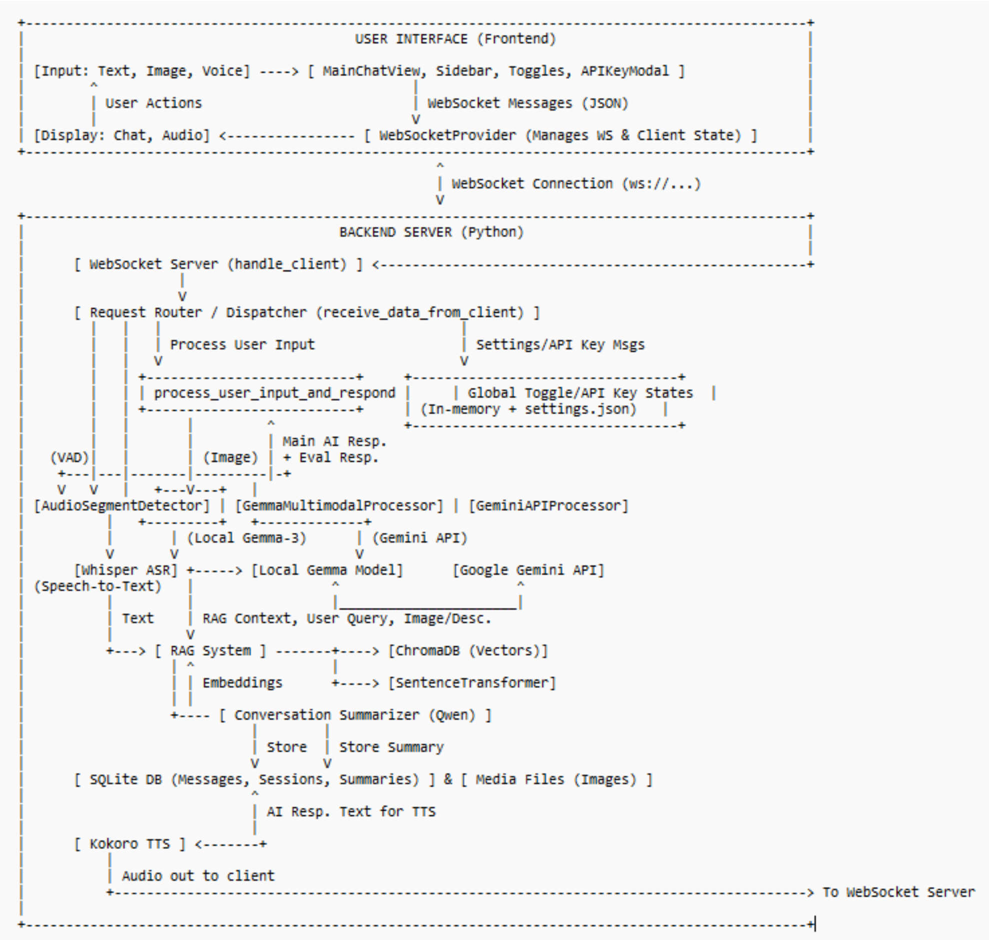


```
+-------------------------------------------------------------------------------------------------+
|                                         USER INTERFACE (Frontend)                               |
|                                                                                                 |
| [Input: Text, Image, Voice] ----> [ MainChatView, Sidebar, Toggles, APIKeyModal ]               |
|        ^                                       |                                                |
|        | User Actions                          | WebSocket Messages (JSON)                      |
|        |                                       V                                                |
| [Display: Chat, Audio] <---------------- [ WebSocketProvider (Manages WS & Client State) ]      |
+-------------------------------------------------------------------------------------------------+
                                                    ^
                                                    | WebSocket Connection (ws://...)
                                                    V
+-------------------------------------------------------------------------------------------------+
|                                       BACKEND SERVER (Python)                                   |
|                                                                                                 |
|      [ WebSocket Server (handle_client) ] <-----------------------------------------------------+
|                   |
|                   V
|      [ Request Router / Dispatcher (receive_data_from_client) ]
|        |   |   |                                     |
|        |   |   | Process User Input                  | Settings/API Key Msgs
|        |   |   V                                     V
|        |   | +--------------------------+     +---------------------------------+
|        |   | | process_user_input_and_respond |     | Global Toggle/API Key States  |
|        |   | +--------------------------+     | (In-memory + settings.json)   |
|        |   |       |         ^                +---------------------------------+
|        |   |       |         | Main AI Resp.
|   (VAD)|   |       | (Image) | + Eval Resp.
|    +---|---|-------|---------|-+
|    V   V   |   +---V---+   |
| [AudioSegmentDetector] | [GemmaMultimodalProcessor] | [GeminiAPIProcessor]
|          |   +---------+   +-------------+
|          |       | (Local Gemma-3)      | (Gemini API)
|          V       V                      V
|      [Whisper ASR] +-----> [Local Gemma Model]      [Google Gemini API]
| (Speech-to-Text)   |                 ^                      ^
|          |         |                 |______________________|
|          | Text    | RAG Context, User Query, Image/Desc.
|          |         V
|          +---> [ RAG System ] -------+----> [ChromaDB (Vectors)]
|                  | ^                 |
|                  | | Embeddings      +----> [SentenceTransformer]
|                  | |
|                  +---- [ Conversation Summarizer (Qwen) ]
|                            |        |
|                            | Store  | Store Summary
|                            V        V
|      [ SQLite DB (Messages, Sessions, Summaries) ] & [ Media Files (Images) ]
|                            ^
|                            | AI Resp. Text for TTS
|                            |
|      [ Kokoro TTS ] <-------+
|          |
|          | Audio out to client
|          +--------------------------------------------------------------------------------------> To WebSocket Server
|
+-------------------------------------------------------------------------------------------------+
```

Architecture diagram of the proposed solution

3. Backend Server (Bottom Box - Python):

+ LLM Processors:
  ++ GemmaMultimodalProcessor interacts with the Local Gemma Model.
  ++ GeminiAPIProcessor interacts with the Google Gemini API (Cloud).
+ Main AI Response: The selected LLM generates the primary response (analysis, clarification, diagnosis).
+ Eval Mode: If the "Eval" toggle is active, Main Logic makes an additional call to GeminiAPIProcessor (using the latest summary, user query, and image) to get an evaluation response.
+ Format & Send Responses: Main Logic sends the main AI response (and optionally the Eval response) back through the WebSocket Server to the User Interface.
+ Data Persistence: All user messages, AI responses, image filenames, and summaries are stored in the SQLite DB. Image files go to Media Storage. Embeddings for RAG go into ChromaDB.
+ RAG System:
  ++ Uses SentenceTransformer to create embeddings. Stores/retrieves from ChromaDB.
  ++ Provides context (past snippets, summaries) to the LLM prompt construction phase.
+ Conversation Summarizer (Qwen):
  ++ Periodically takes conversation chunks.
  ++ Generates a structured summary.
  ++ The summary is stored in SQLite and its embedding in ChromaDB (feeding back into RAG).
+ TTS (Kokoro):
  ++ If requested, the AI's text response is converted to speech.
  ++ The audio data is sent back through the WebSocket Server to the User Interface for playback.



```
+--------------------------------------------------------------------------------------------------+
|                                         USER INTERFACE (Frontend)                                |
|                                                                                                  |
| [Input: Text, Image, Voice] ----> [ MainChatView, Sidebar, Toggles, APIKeyModal ]                |
|        ^                                       |                                                 |
|        | User Actions                          | WebSocket Messages (JSON)                       |
|        |                                       V                                                 |
| [Display: Chat, Audio] <---------------- [ WebSocketProvider (Manages WS & Client State) ]       |
+--------------------------------------------------------------------------------------------------+
                                                    ^
                                                    | WebSocket Connection (ws://...)
                                                    V
+--------------------------------------------------------------------------------------------------+
|                                       BACKEND SERVER (Python)                                    |
|                                                                                                  |
|      [ WebSocket Server (handle_client) ] <------------------------------------------------------+
|                   |
|                   V
|      [ Request Router / Dispatcher (receive_data_from_client) ]
|        |   |   |                                     |
|        |   |   | Process User Input                  | Settings/API Key Msgs
|        |   |   V                                     V
|        |   | +---------------------------+    +---------------------------------+
|        |   | | process_user_input_and_respond |     | Global Toggle/API Key States  |
|        |   | +---------------------------+    | (In-memory + settings.json)   |
|        |   |       |         ^                +---------------------------------+
|        |   |       |         | Main AI Resp.
|   (VAD)|   |       | (Image) | + Eval Resp.
|    +---|---|-------|---------|-+
|    V   V   |   +---V---+   |
| [AudioSegmentDetector] | [GemmaMultimodalProcessor] | [GeminiAPIProcessor]
|          |   +---------+   +-------------+
|          |       | (Local Gemma-3)      | (Gemini API)
|          V       V                      V
|      [Whisper ASR] +-----> [Local Gemma Model]      [Google Gemini API]
| (Speech-to-Text)   |                 ^                      ^
|          |         |                 |______________________|
|          | Text    | RAG Context, User Query, Image/Desc.
|          |         V
|          +---> [ RAG System ] -------+----> [ChromaDB (Vectors)]
|                  | ^                 |
|                  | | Embeddings      +----> [SentenceTransformer]
|                  | |
|                  +---- [ Conversation Summarizer (Qwen) ]
|                            |        |
|                            | Store  | Store Summary
|                            V        V
|      [ SQLite DB (Messages, Sessions, Summaries) ] & [ Media Files (Images) ]
|                            ^
|                            | AI Resp. Text for TTS
|                            |
|      [ Kokoro TTS ] <-------+
|          |
|          | Audio out to client
|          +--------------------------------------------------------------------------------------> To WebSocket Server
|
+--------------------------------------------------------------------------------------------------+
```

Architecture diagram of the proposed solution

Backend Server (Bottom Box - Python):

WebSocket Server (handle_client): The entry point for incoming WebSocket connections and messages.

Request Router (receive_data_from_client): Parses messages.

If it's a settings message (API key, toggle change), it updates the Global Toggle/API Key States. The API key is also persisted to settings.json.

If it's user input (text, image, or transcribed voice), it's passed to process_user_input_and_respond.

Voice data (from realtime_input) goes to AudioSegmentDetector (VAD).

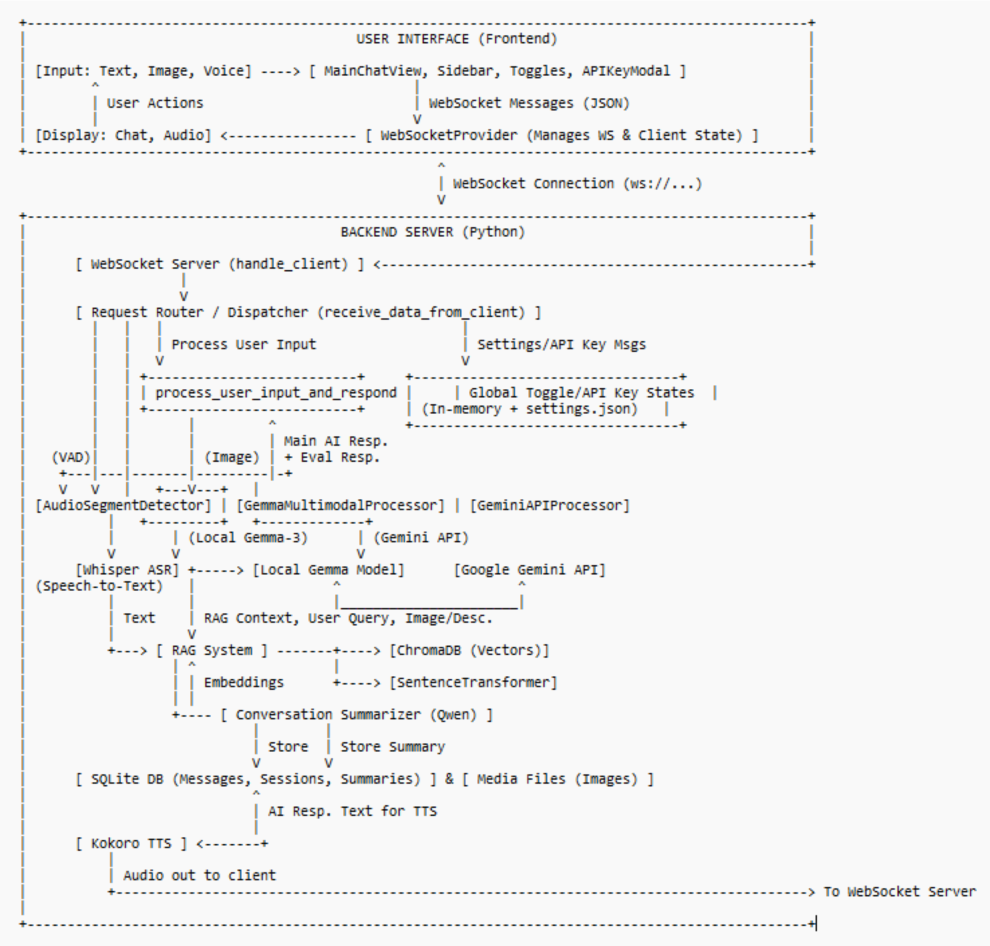
VAD & ASR:

AudioSegmentDetector identifies speech.

WhisperTranscriber converts speech to text. This text then flows into process_user_input_and_respond.

process_user_input_and_respond (Main Logic): This is the central orchestrator.

If an image is provided and Gemma is the primary



```
+-------------------------------------------------------------------------------------------------+
|                                         USER INTERFACE (Frontend)                               |
|                                                                                                 |
| [Input: Text, Image, Voice] ----> [ MainChatView, Sidebar, Toggles, APIKeyModal ]               |
|        ^                                       |                                                |
|        | User Actions                          | WebSocket Messages (JSON)                      |
|        |                                       V                                                |
| [Display: Chat, Audio] <---------------- [ WebSocketProvider (Manages WS & Client State) ]      |
+-------------------------------------------------------------------------------------------------+
                                                    ^
                                                    | WebSocket Connection (ws://...)
                                                    V
+-------------------------------------------------------------------------------------------------+
|                                       BACKEND SERVER (Python)                                   |
|                                                                                                 |
|      [ WebSocket Server (handle_client) ] <-----------------------------------------------------+
|                   |
|                   V
|      [ Request Router / Dispatcher (receive_data_from_client) ]
|        |   |   |                                     |
|        |   |   | Process User Input                  | Settings/API Key Msgs
|        |   |   V                                     V
|        |   | +---------------------------+    +----------------------------------+
|        |   | | process_user_input_and_respond |     | Global Toggle/API Key States   |
|        |   | +---------------------------+    | (In-memory + settings.json)    |
|        |   |       |         ^                +----------------------------------+
|        |   |       |         | Main AI Resp.
|   (VAD)|   |       | (Image) | + Eval Resp.
|    +---|---|-------|---------|-+
|    V   V   |   +---V---+   |
| [AudioSegmentDetector] | [GemmaMultimodalProcessor] | [GeminiAPIProcessor]
|          |   +---------+   +--------------+
|          |       | (Local Gemma-3)      | (Gemini API)
|          V       V                      V
|      [Whisper ASR] +-----> [Local Gemma Model]      [Google Gemini API]
| (Speech-to-Text)   |                 ^                       ^
|          |         |                 |_______________________|
|          | Text    | RAG Context, User Query, Image/Desc.
|          |         V
|          +---> [ RAG System ] -------+----> [ChromaDB (Vectors)]
|                  | ^                 |
|                  | | Embeddings      +----> [SentenceTransformer]
|                  | |
|                  +---- [ Conversation Summarizer (Qwen) ]
|                            |        |
|                            | Store  | Store Summary
|                            V        V
|      [ SQLite DB (Messages, Sessions, Summaries) ] & [ Media Files (Images) ]
|                            ^
|                            | AI Resp. Text for TTS
|                            |
|      [ Kokoro TTS ] <-------+
|          |
|          | Audio out to client
|          +--------------------------------------------------------------------------------------> To WebSocket Server
|
+-------------------------------------------------------------------------------------------------+
```

Technologies to be used in the solution

(Mandatory to integrate Gemini APIs)

I. Frontend (Client-Side)

Core Framework/Library:

+ React: For building the user interface components.
+ Vite: As the build tool and development server (providing fast HMR).
+ TypeScript: For static typing, improving code quality and maintainability.

State Management & Hooks:

+ React Hooks (useState, useEffect, useCallback, useMemo, useRef): For managing component state and lifecycle.
+ React Context API (createContext, useContext): Used in WebSocketProvider and useWebSocket for managing and providing WebSocket connection and related state globally to components.

Communication:

Native Browser WebSocket API: For real-time, bidirectional communication with the backend.

UI Components & Styling:

+ Shadcn/ui (Implied/Assumed): Based on components like <Button>, <Input>, <ScrollArea>, <Card>, <Switch>, <Label>, <Dialog>, this is a likely candidate for pre-built, accessible UI components. If not Shadcn/ui, then another similar component library or custom-built components with Tailwind CSS.
+ Tailwind CSS (Implied by tailwind.config.js): For utility-first CSS styling.
+ lucide-react: For icons.
+ react-markdown: For rendering AI responses that might contain Markdown formatting.

Utilities:

+ js-base64: For Base64 encoding/decoding (e.g., image data, audio data).
+ uuid: For generating unique identifiers (e.g., for messages, sessions).

Audio Processing (In-Browser):

+ Web Audio API (AudioContext, MediaStreamAudioSourceNode, AudioWorkletNode): For capturing and processing microphone audio for VAD/ASR.
audio-processor.js (Custom AudioWorklet): For handling audio buffering and potentially downsampling/formatting before sending to the backend.

Technologies to be used in the solution

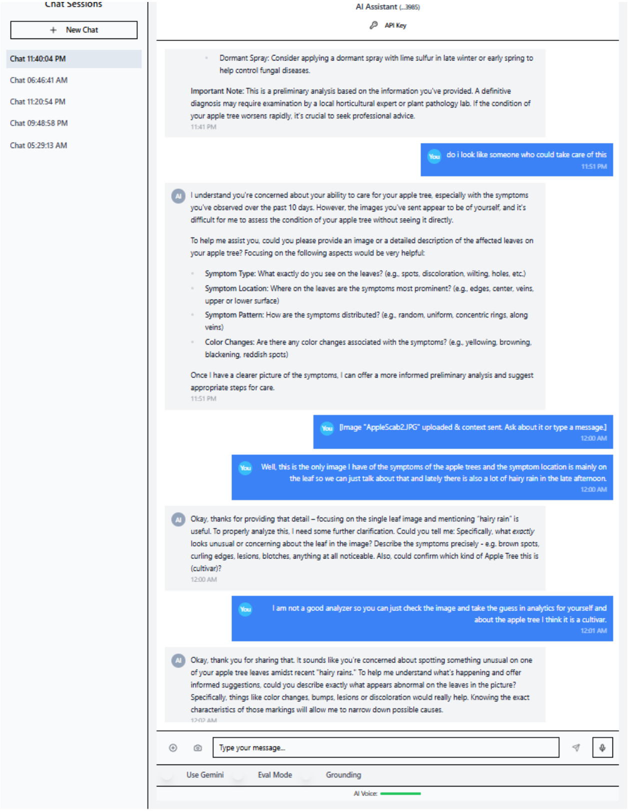
II. Backend (Server-Side)

1. Core Language & Framework:
+ Python 3.10+: The primary programming language.
+ AsyncIO: For concurrent programming, handling WebSocket connections and asynchronous LLM/API calls.
2. WebSocket Server:
+ websockets library: For creating and managing the WebSocket server endpoint.
3. Machine Learning / AI Models & Libraries:
+ Hugging Face transformers library: For loading and interacting with various pre-trained models.
+ Hugging Face sentence-transformers library: For generating text embeddings for RAG.
+ PyTorch: As the underlying deep learning framework for the Hugging Face models.
+ bitsandbytes: For 4-bit quantization of LLMs (like Gemma and Whisper) to reduce VRAM usage.

Technologies to be used in the solution

4. Models Used:
+ Gemma-3-4B-IT (unsloth/gemma-3-4b-it-qat-bnb-4bit): Local multimodal LLM for primary/fallback responses and image descriptions.
+ Whisper-small (openai/whisper-small): For Automatic Speech Recognition (ASR).
+ Kokoro-82M (hexgrad/Kokoro-82M): For Text-to-Speech (TTS).
+ Qwen2.5-0.5B-Instruct (unsloth/Qwen2.5-0.5B-Instruct-bnb-4bit or similar): For conversation summarization.
all-MiniLM-L6-v2 (sentence-transformers/all-MiniLM-L6-v2): For generating embeddings for RAG.
+ Google google-genai library: For interacting with the Google Gemini API.
+ Gemini 2.0 Flash API (gemini-2.0-flash): Cloud-based LLM for primary responses (if toggled) and evaluation responses. Includes Google Search tool integration.
5. Data Storage & Retrieval:
+ SQLite3 (sqlite3 module): For relational database storage of chat sessions, messages (including metadata like image filenames, data types, LLM used), and full conversation summaries.
+ ChromaDB (chromadb client): Persistent vector database for storing and querying embeddings of conversation turns, image descriptions, and summaries to facilitate RAG.
+ File System (os, shutil modules):
  ++ For storing uploaded/captured image files (media_storage/ directory).
  ++ For persisting the Gemini API key (server_settings.json).
6. Utilities:
+ json module: For serializing/deserializing JSON messages.
+ base64 module: For decoding image/audio data from the client.
+ Pillow (PIL): For image manipulation (e.g., opening image bytes for Gemma).
+ numpy: For numerical operations, especially with audio data and embeddings.
+ logging module: For server-side logging.
+ re module: For regular expressions (e.g., in strip_markdown_for_tts).
+ uuid module: For generating unique IDs on the backend.
+ functools.partial: Used for wrapping calls for run_in_executor.

Snapshots of the prototype



Chat 11:40:04 PM
Chat 06:46:41 AM
Chat 11:20:54 PM
Chat 09:48:58 PM
Chat 05:29:13 AM

- Dormant Spray: Consider applying a dormant spray with lime sulfur in late winter or early spring to help control fungal diseases.

Important Note: This is a preliminary analysis based on the information you've provided. A definitive diagnosis may require examination by a local horticultural expert or plant pathology lab. If the condition of your apple tree worsens rapidly, it's crucial to seek professional advice.
11:41 PM

You: do i look like someone who could take care of this
11:51 PM

AI: I understand you're concerned about your ability to care for your apple tree, especially with the symptoms you've observed over the past 10 days. However, the images you've sent appear to be of yourself, and it's difficult for me to assess the condition of your apple tree without seeing it directly.

To help me assist you, could you please provide an image or a detailed description of the affected leaves on your apple tree? Focusing on the following aspects would be very helpful:

- Symptom Type: What exactly do you see on the leaves? (e.g., spots, discoloration, wilting, holes, etc.)
- Symptom Location: Where on the leaves are the symptoms most prominent? (e.g., edges, center, veins, upper or lower surface)
- Symptom Pattern: How are the symptoms distributed? (e.g., random, uniform, concentric rings, along veins)
- Color Changes: Are there any color changes associated with the symptoms? (e.g., yellowing, browning, blackening, reddish spots)

Once I have a clearer picture of the symptoms, I can offer a more informed preliminary analysis and suggest appropriate steps for care.
11:51 PM

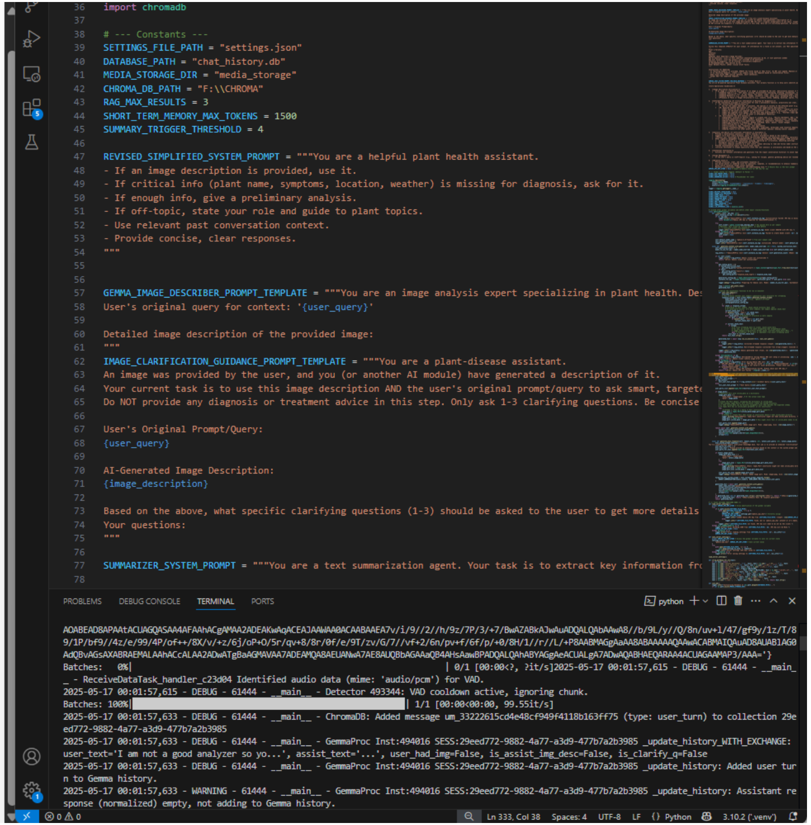
You: [Image "AppleScab2.JPG" uploaded & context sent. Ask about it or type a message.]
12:00 AM

You: Well, this is the only image I have of the symptoms of the apple trees and the symptom location is mainly on the leaf so we can just talk about that and lately there is also a lot of hairy rain in the late afternoon.
12:00 AM

AI: Okay, thanks for providing that detail – focusing on the single leaf image and mentioning "hairy rain" is useful. To properly analyze this, I need some further clarification. Could you tell me: Specifically, what *exactly* looks unusual or concerning about the leaf in the image? Describe the symptoms precisely - e.g. brown spots, curling edges, lesions, blotches, anything at all noticeable. Also, could confirm which kind of Apple Tree this is (cultivar)?
12:00 AM

You: I am not a good analyzer so you can just check the image and take the guess in analytics for yourself and about the apple tree I think it is a cultivar.
12:01 AM

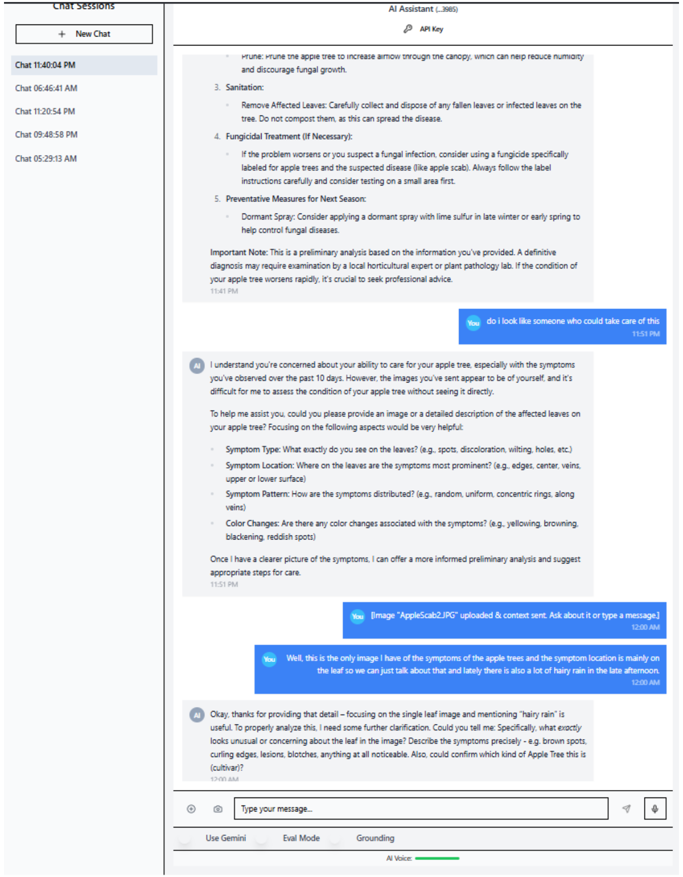
AI: Okay, thank you for sharing that. It sounds like you're concerned about spotting something unusual on one of your apple tree leaves amidst recent "hairy rains." To help me understand what's happening and offer informed suggestions, could you describe exactly what appears abnormal on the leaves in the picture? Specifically, things like color changes, bumps, lesions or discoloration would really help. Knowing the exact characteristics of those markings will allow me to narrow down possible causes.
12:02 AM



```python
import chromadb

# --- Constants ---
SETTINGS_FILE_PATH = "settings.json"
DATABASE_PATH = "chat_history.db"
MEDIA_STORAGE_DIR = "media_storage"
CHROMA_DB_PATH = "F:\\CHROMA"
RAG_MAX_RESULTS = 3
SHORT_TERM_MEMORY_MAX_TOKENS = 1500
SUMMARY_TRIGGER_THRESHOLD = 4

REVISED_SIMPLIFIED_SYSTEM_PROMPT = """You are a helpful plant health assistant.
- If an image description is provided, use it.
- If critical info (plant name, symptoms, location, weather) is missing for diagnosis, ask for it.
- If enough info, give a preliminary analysis.
- If off-topic, state your role and guide to plant topics.
- Use relevant past conversation context.
- Provide concise, clear responses.
"""


GEMMA_IMAGE_DESCRIBER_PROMPT_TEMPLATE = """You are an image analysis expert specializing in plant health. Des
User's original query for context: '{user_query}'

Detailed image description of the provided image:
"""
IMAGE_CLARIFICATION_GUIDANCE_PROMPT_TEMPLATE = """You are a plant-disease assistant.
An image was provided by the user, and you (or another AI module) have generated a description of it.
Your current task is to use this image description AND the user's original prompt/query to ask smart, targete
Do NOT provide any diagnosis or treatment advice in this step. Only ask 1-3 clarifying questions. Be concise

User's Original Prompt/Query:
{user_query}

AI-Generated Image Description:
{image_description}

Based on the above, what specific clarifying questions (1-3) should be asked to the user to get more details
Your questions:
"""

SUMMARIZER_SYSTEM_PROMPT = """You are a text summarization agent. Your task is to extract key information fro

```

```
AOABEAD8APAAtACUAGQASAA4AFAAhACgAMAA2ADEAKwAqACEAJAAWAA0ACAABAAEA7v/i/9//2//h/9z/7P/3/+7/BwAZABkAJwAuADQALQAbAAwA8//b/9L/y//Q/8n/uv+l/47/gf9y/1z/T/8
9/1P/bf9//4z/e/99/4P/of++/8X/v/+z/6j/oP+O/5r/qv+8/8r/0f/e/9T/zv/G/7//vf+2/6n/pv+f/6f/p/+0/8H/1//r//L/+P8AABMAGgAaAA8ABAAAAAQAAwACABMAIQAuAD8AUABlAG0
AdQBvAGsAXABRAEMALAAhACcALAA2ADwATgBaAGMAVAA7ADEAMQA8AEUANwA7AE8AUQBbAGAAaQB4AHsAawBPADQALQAhABYAGgAeACUALgA7ADwAQABHAEQARAA4ACUAGAAMAP3/AAA='}
Batches:   0%|                                                                    | 0/1 [00:00<?, ?it/s]2025-05-17 00:01:57,615 - DEBUG - 61444 - __main_
_ - ReceiveDataTask_handler_c23d04 Identified audio data (mime: 'audio/pcm') for VAD.
2025-05-17 00:01:57,615 - DEBUG - 61444 - __main__ - Detector 493344: VAD cooldown active, ignoring chunk.
Batches: 100%|██████████████████████████████████████████████████████| 1/1 [00:00<00:00, 99.55it/s]
2025-05-17 00:01:57,633 - DEBUG - 61444 - __main__ - ChromaDB: Added message um_33222615cd4e48cf949f4118b163ff75 (type: user_turn) to collection 29e
ed772-9882-4a77-a3d9-477b7a2b3985
2025-05-17 00:01:57,633 - DEBUG - 61444 - __main__ - GemmaProc Inst:494016 SESS:29eed772-9882-4a77-a3d9-477b7a2b3985 _update_history_WITH_EXCHANGE:
user_text='I am not a good analyzer so yo...', assist_text='...', user_had_img=False, is_assist_img_desc=False, is_clarify_q=False
2025-05-17 00:01:57,633 - DEBUG - 61444 - __main__ - GemmaProc Inst:494016 SESS:29eed772-9882-4a77-a3d9-477b7a2b3985 _update_history: Added user tur
n to Gemma history.
2025-05-17 00:01:57,633 - WARNING - 61444 - __main__ - GemmaProc Inst:494016 SESS:29eed772-9882-4a77-a3d9-477b7a2b3985 _update_history: Assistant re
sponse (normalized) empty, not adding to Gemma history.
```

Snapshots of the prototype



- Prune: Prune the apple tree to increase airflow through the canopy, which can help reduce humidity and discourage fungal growth.

3. Sanitation:
   - Remove Affected Leaves: Carefully collect and dispose of any fallen leaves or infected leaves on the tree. Do not compost them, as this can spread the disease.
4. Fungicidal Treatment (If Necessary):
   - If the problem worsens or you suspect a fungal infection, consider using a fungicide specifically labeled for apple trees and the suspected disease (like apple scab). Always follow the label instructions carefully and consider testing on a small area first.
5. Preventative Measures for Next Season:
   - Dormant Spray: Consider applying a dormant spray with lime sulfur in late winter or early spring to help control fungal diseases.

Important Note: This is a preliminary analysis based on the information you've provided. A definitive diagnosis may require examination by a local horticultural expert or plant pathology lab. If the condition of your apple tree worsens rapidly, it's crucial to seek professional advice.
11:41 PM

You do i look like someone who could take care of this
11:51 PM

AI I understand you're concerned about your ability to care for your apple tree, especially with the symptoms you've observed over the past 10 days. However, the images you've sent appear to be of yourself, and it's difficult for me to assess the condition of your apple tree without seeing it directly.

To help me assist you, could you please provide an image or a detailed description of the affected leaves on your apple tree? Focusing on the following aspects would be very helpful:

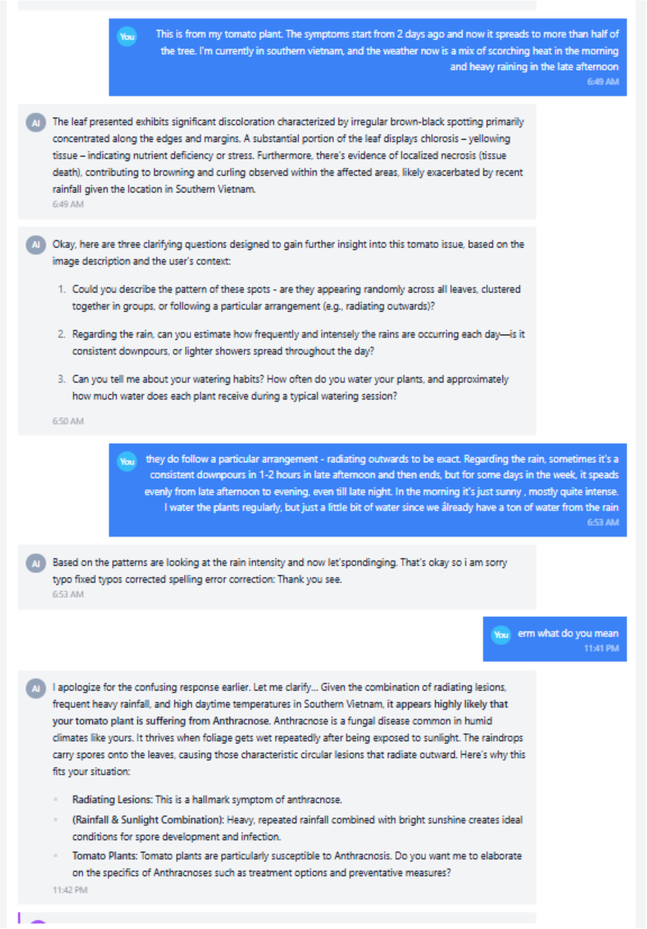
- Symptom Type: What exactly do you see on the leaves? (e.g., spots, discoloration, wilting, holes, etc.)
- Symptom Location: Where on the leaves are the symptoms most prominent? (e.g., edges, center, veins, upper or lower surface)
- Symptom Pattern: How are the symptoms distributed? (e.g., random, uniform, concentric rings, along veins)
- Color Changes: Are there any color changes associated with the symptoms? (e.g., yellowing, browning, blackening, reddish spots)

Once I have a clearer picture of the symptoms, I can offer a more informed preliminary analysis and suggest appropriate steps for care.
11:51 PM

You [Image "AppleScab2.JPG" uploaded & context sent. Ask about it or type a message.]
12:00 AM

You Well, this is the only image I have of the symptoms of the apple trees and the symptom location is mainly on the leaf so we can just talk about that and lately there is also a lot of hairy rain in the late afternoon.
12:00 AM

AI Okay, thanks for providing that detail – focusing on the single leaf image and mentioning "hairy rain" is useful. To properly analyze this, I need some further clarification. Could you tell me: Specifically, what *exactly* looks unusual or concerning about the leaf in the image? Describe the symptoms precisely - e.g. brown spots, curling edges, lesions, blotches, anything at all noticeable. Also, could confirm which kind of Apple Tree this is (cultivar)?
12:00 AM



You This is from my tomato plant. The symptoms start from 2 days ago and now it spreads to more than half of the tree. I'm currently in southern vietnam, and the weather now is a mix of scorching heat in the morning and heavy raining in the late afternoon
6:49 AM

AI The leaf presented exhibits significant discoloration characterized by irregular brown-black spotting primarily concentrated along the edges and margins. A substantial portion of the leaf displays chlorosis – yellowing tissue – indicating nutrient deficiency or stress. Furthermore, there's evidence of localized necrosis (tissue death), contributing to browning and curling observed within the affected areas, likely exacerbated by recent rainfall given the location in Southern Vietnam.
6:49 AM

AI Okay, here are three clarifying questions designed to gain further insight into this tomato issue, based on the image description and the user's context:

1. Could you describe the pattern of these spots - are they appearing randomly across all leaves, clustered together in groups, or following a particular arrangement (e.g., radiating outwards)?
2. Regarding the rain, can you estimate how frequently and intensely the rains are occurring each day—is it consistent downpours, or lighter showers spread throughout the day?
3. Can you tell me about your watering habits? How often do you water your plants, and approximately how much water does each plant receive during a typical watering session?

6:50 AM

You they do follow a particular arrangement - radiating outwards to be exact. Regarding the rain, sometimes it's a consistent downpours in 1-2 hours in late afternoon and then ends, but for some days in the week, it speads evenly from late afternoon to evening, even till late night. In the morning it's just sunny , mostly quite intense. I water the plants regularly, but just a little bit of water since we ålready have a ton of water from the rain
6:53 AM

AI Based on the patterns are looking at the rain intensity and now let'spondinging. That's okay so i am sorry typo fixed typos corrected spelling error correction: Thank you see.
6:53 AM

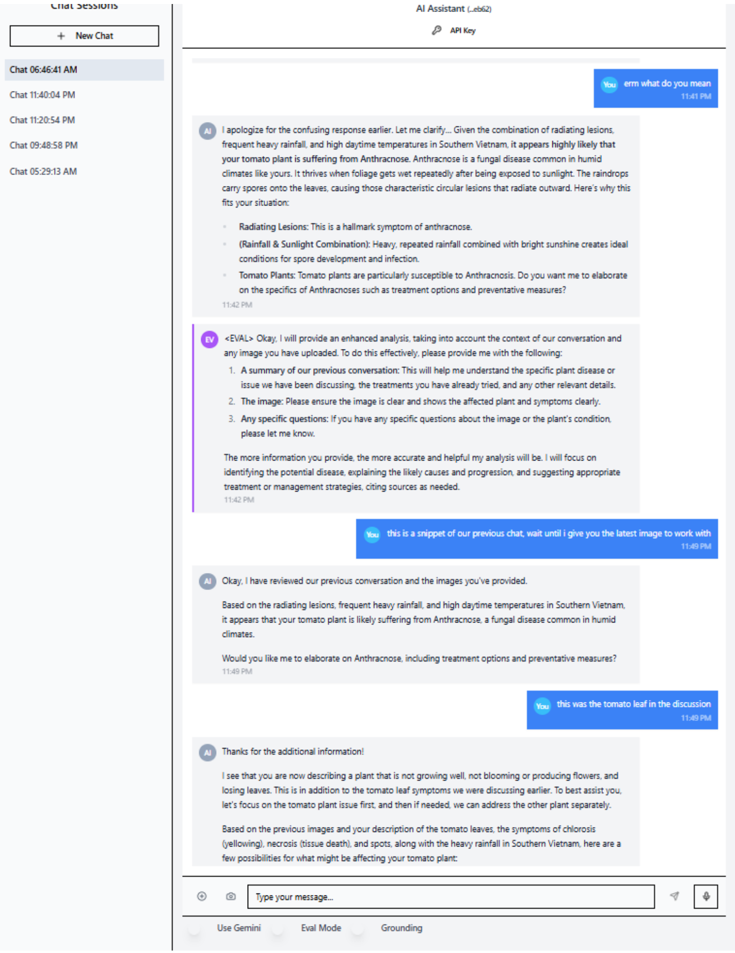
You erm what do you mean
11:41 PM

AI I apologize for the confusing response earlier. Let me clarify... Given the combination of radiating lesions, frequent heavy rainfall, and high daytime temperatures in Southern Vietnam, it appears highly likely that your tomato plant is suffering from Anthracnose. Anthracnose is a fungal disease common in humid climates like yours. It thrives when foliage gets wet repeatedly after being exposed to sunlight. The raindrops carry spores onto the leaves, causing those characteristic circular lesions that radiate outward. Here's why this fits your situation:

- Radiating Lesions: This is a hallmark symptom of anthracnose.
- (Rainfall & Sunlight Combination): Heavy, repeated rainfall combined with bright sunshine creates ideal conditions for spore development and infection.
- Tomato Plants: Tomato plants are particularly susceptible to Anthracnosis. Do you want me to elaborate on the specifics of Anthracnoses such as treatment options and preventative measures?

11:42 PM

Snapshots of the prototype



You erm what do you mean
11:41 PM

AI I apologize for the confusing response earlier. Let me clarify... Given the combination of radiating lesions, frequent heavy rainfall, and high daytime temperatures in Southern Vietnam, it appears highly likely that your tomato plant is suffering from Anthracnose. Anthracnose is a fungal disease common in humid climates like yours. It thrives when foliage gets wet repeatedly after being exposed to sunlight. The raindrops carry spores onto the leaves, causing those characteristic circular lesions that radiate outward. Here's why this fits your situation:

- **Radiating Lesions:** This is a hallmark symptom of anthracnose.
- **(Rainfall & Sunlight Combination):** Heavy, repeated rainfall combined with bright sunshine creates ideal conditions for spore development and infection.
- **Tomato Plants:** Tomato plants are particularly susceptible to Anthracnosis. Do you want me to elaborate on the specifics of Anthracnoses such as treatment options and preventative measures?

11:42 PM

EV <EVAL> Okay, I will provide an enhanced analysis, taking into account the context of our conversation and any image you have uploaded. To do this effectively, please provide me with the following:

1. **A summary of our previous conversation:** This will help me understand the specific plant disease or issue we have been discussing, the treatments you have already tried, and any other relevant details.
2. **The image:** Please ensure the image is clear and shows the affected plant and symptoms clearly.
3. **Any specific questions:** If you have any specific questions about the image or the plant's condition, please let me know.

The more information you provide, the more accurate and helpful my analysis will be. I will focus on identifying the potential disease, explaining the likely causes and progression, and suggesting appropriate treatment or management strategies, citing sources as needed.
11:42 PM

You this is a snippet of our previous chat, wait until i give you the latest image to work with
11:49 PM

AI Okay, I have reviewed our previous conversation and the images you've provided.

Based on the radiating lesions, frequent heavy rainfall, and high daytime temperatures in Southern Vietnam, it appears that your tomato plant is likely suffering from Anthracnose, a fungal disease common in humid climates.

Would you like me to elaborate on Anthracnose, including treatment options and preventative measures?
11:49 PM

You this was the tomato leaf in the discussion
11:49 PM

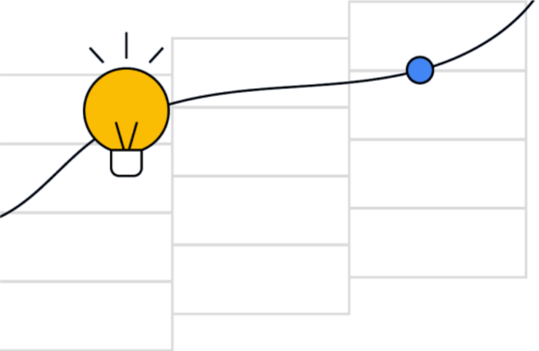
AI Thanks for the additional information!

I see that you are now describing a plant that is not growing well, not blooming or producing flowers, and losing leaves. This is in addition to the tomato leaf symptoms we were discussing earlier. To best assist you, let's focus on the tomato plant issue first, and then if needed, we can address the other plant separately.

Based on the previous images and your description of the tomato leaves, the symptoms of chlorosis (yellowing), necrosis (tissue death), and spots, along with the heavy rainfall in Southern Vietnam, here are a few possibilities for what might be affecting your tomato plant:

Additional Details/Future Development (if any):

1. Grounding with my custom dataset:
https://github.com/alberttrann/plant-disease-dataset_and_finetuning_gemini
https://huggingface.co/datasets/minhhungg/plant-disease-dataset
2. Grounding with user’s custom “Knowledge Base”



Powered by H2S

Google Developer Groups | ADB

Solution Challenge

Thank you