# **MultiModal Chat Assistant Documentation**

## **Overview**

The **MultiModal Chat Assistant** is an intelligent AI-powered assistant capable of understanding and responding to **both text and image inputs**.  
 It leverages **LLaVA (Large Language and Vision Assistant)** and **FastAPI** to deliver conversational and contextual responses in real time.

This project is designed to demonstrate multimodal AI capabilities — combining **language understanding**, **image comprehension**, and **contextual reasoning**.

## **Features**

* 🧠 **Text-based conversation** using a large language model
* 🖼️ **Image understanding** and caption-based question answering via LLaVA
* ⚙️ **FastAPI backend** for API integration
* 💬 **Frontend interface** for user interaction
* 🪶 **Lightweight & modular codebase** for easy customization and expansion
* 💻 Runs on both **CPU** and **GPU** environments

## **⚙️ Setup Instructions**

### **1️⃣ Clone the Repository**

git clone https://github.com/Rincysamuel/MultimodalChatAssistant.git

cd MultiModalChatAssistant

### **2️⃣ Create a Virtual Environment**

python -m venv venv

venv\Scripts\activate # On Windows

# source venv/bin/activate # On Mac/Linux

### **3️⃣ Install Dependencies**

pip install -r requirements.txt

🧩 If you’re using LLaVA, ensure you also install its dependencies.

## **🧠 Model Configuration (LLaVA)**

The backend/llava\_model.py file wraps the **LLaVA model** for multimodal processing.

llava = LLaVAWrapper(device="cpu") # or 'cuda' for GPU

response = llava.predict(["Describe this image"], [image])

The model uses a configuration file:

configs/model\_config.yaml

This defines model weights, tokenizer, and image encoder setup.

## **🧩 API Endpoints**

### **POST /process-audio/**

Processes an uploaded audio file and returns:

* Transcript
* Summary
* Action items

**Example Request:**

curl -X POST "http://127.0.0.1:8000/process-audio/" \

-F "file=@meeting.mp3"

**Response:**

{

"transcript": "Today's meeting covered project deadlines...",

"summary": "Discussed tasks and delivery timeline.",

"action\_items": ["Update the API docs", "Review new dataset"]

}

## **🧩 Run the Backend Server**

uvicorn backend.main:app --reload

The app will run at:

http://127.0.0.1:8000

You can check API docs at:

http://127.0.0.1:8000/docs

## **💬 Example Flow**

1. User uploads an image or sends a text query.
2. Backend routes request to LLaVA model for multimodal understanding.
3. Model generates context-aware response.
4. Response is returned to frontend for display.

## **Tech Stack**

| **Component** | **Technology** |
| --- | --- |
| **Backend** | FastAPI |
| **Frontend** | Streamlit / React (optional) |
| **Model** | LLaVA (Large Language and Vision Assistant) |
| **Language** | Python 3.10+ |
| **Environment** | Virtualenv |
| **Server** | Uvicorn |