**MediChat AI: Comprehensive Documentation**

**1. Project Overview**

**1.1 Project Goals**

The primary goal of this project is to develop an AI-powered medical chatbot that enables front-desk healthcare professionals to efficiently retrieve patient records, identify similar cases, and provide relevant medical insights. This chatbot leverages **Retrieval-Augmented Generation (RAG)** using **ChromaDB** for vector storage and **OpenAI's GPT model** for response generation.

**1.2 Key Features**

* **Conversational AI:** Multi-turn chat support (up to 3 interactions per session)
* **Patient Record Retrieval:** Fetch similar cases from ChromaDB
* **Moderation System:** Filters unsafe queries using OpenAI's moderation API
* **Efficient Search:** Uses OpenAI Embeddings to find relevant records
* **Production-Ready API:** Future scalability with LangChain integration

**2. Data Sources**

The chatbot is powered by structured **medical history data** stored in ChromaDB. The data includes:

* **Patient Demographics:** Name, age, gender
* **Symptoms & Diagnoses:** Key symptoms and diagnosed conditions
* **Medical History:** Past ailments, chronic conditions
* **Medications & Treatments:** Prescribed drugs and treatment plans

Data is converted into **JSON format**, embedded using **OpenAI Embeddings**, and stored in **ChromaDB** for vector-based retrieval.

**3. System Design**

**3.1 Architectural Components**

* **Frontend:** Gradio-based UI for user interaction
* **Backend:** Python-based API using OpenAI & ChromaDB
* **Database:** ChromaDB for efficient vector storage
* **Embedding Model:** OpenAI Embeddings for document similarity search
* **LLM (Language Model):** GPT-3.5-Turbo for chatbot responses
* **Moderation API:** OpenAI Moderation API for query filtering

**3.2 Workflow**

1. **User Query Input:** Front-desk personnel enter a patient query.
2. **Moderation Check:** Query passes through OpenAI’s moderation API.
3. **ChromaDB Retrieval:** If safe, search for similar cases using vector embeddings.
4. **GPT Processing:** Retrieved patient records are passed to GPT for response generation.
5. **AI Response:** The chatbot provides a medically relevant answer.
6. **Multi-turn Conversation Support:** Users can refine queries (up to 3 turns).

**4. Chatbot Architecture & Workflow**

**4.1 Chatbot Pipeline**

User Query → Moderation API → ChromaDB Search → GPT Response → Multi-turn Chat

**4.2 Workflow Breakdown**

**Step 1: User Query & Moderation**

* User inputs a medical-related query.
* OpenAI Moderation API flags inappropriate or unsafe queries.
* If flagged, the chatbot requests a rephrased input.

**Step 2: Vector Database Search (ChromaDB)**

* The safe query is converted into an embedding using OpenAI’s Embedding API.
* ChromaDB is queried for similar patient cases.
* The retrieved cases (if any) are formatted and passed to the chatbot.

**Step 3: GPT Response Generation**

* The chatbot formulates a response based on the retrieved data.
* The response is generated using **GPT-3.5-Turbo** with a token limit.

**Step 4: Multi-turn Conversation**

* Users can ask follow-up questions (conversation reset if typed exit).
* The chatbot retains context across turns.

**5. Challenges Faced**

**5.1 Handling Inconsistent JSON Responses**

* Issue: GPT-generated JSON responses were sometimes inconsistent.
* Solution: Used **strict JSON formatting prompts** & error handling.

**5.2 Ensuring Vector Similarity Matches**

* Issue: Inconsistent embeddings led to irrelevant patient records.
* Solution: Used **OpenAI Embeddings** for better vector retrieval.

**5.3 Multi-Turn Chatbot Logic**

* Issue: Responses needed context retention over multiple turns.
* Solution: Implemented **chat history tracking** within sessions.

**6. Future Enhancements**

* **LangChain Integration:** Modular framework for better LLM orchestration.
* **Hybrid Search:** Combine keyword-based & vector search for better retrieval.
* **HIPAA Compliance:** Ensure patient data security & compliance.
* **Fine-Tuning GPT:** Train a domain-specific model for improved accuracy.

**7. Conclusion**

This project successfully builds an intelligent **medical AI chatbot** capable of assisting front-desk professionals in retrieving patient records, identifying similar cases, and improving operational efficiency in healthcare settings. The integration of **ChromaDB, OpenAI GPT, and multi-turn chat logic** makes it a powerful tool with room for further scalability.

🚀 **Next Steps:** Deploy as an API for seamless integration into hospital systems!