Medical RAG ChatBot Technical Documentation

Emre Büyükyılmaz

September 18, 2025

Contents

Overview

This project provides:

- **API** (FastAPI on Uvicorn): retrieval-augmented generation and chat/log management.
- **UI** (Gradio): a web interface to interact with the API.

Core components:

- **LLM**: `meta-llama/Meta-Llama-3-8B-Instruct`
- **Retriever**: FAISS index + BioBERT embeddings
- **SQLite**: API logs and chat sessions/messages
- **Docker Compose**: separate API and UI services

Architecture

High-Level

```
+-----------------+        Docker network       +-----------------+
|     UI (7860)   |  --->  http://app:8000  ---> |   API (8000)    |
|   Gradio app    |                              |  FastAPI+LLM    |
+-----------------+                              +-----------------+
            ^                                             |
            |                                             v
            |---------------------------------------------+
```

Services and Ports

- **API**: 8000:8000
- **UI**: 7860:7860

Setup

Prerequisites

- Docker Desktop (Windows) with WSL2 backend
- NVIDIA GPU recommended (CUDA-enabled base image)
- Hugging Face account with a `read` token

Environment Variables

Create `.env` next to `compose.yaml`:

```
HF_TOKEN=hf_xxxxxxxxxxxxxxxxxxxxxxxxx
```

Build & Run

```
# From the project root:
docker compose up --build
# Subsequent runs:
docker compose up
```

UI: http://localhost:7860   API: http://localhost:8000

Database Schema

SQLite at `data/logs/chat_logs.db`:

- **api_logs**: id, timestamp, question, answer, retrieved_sources (JSON), retrieval_time, generation_time, total_time
- **chat_sessions**: session_id (PK), created_at
- **chat_messages**: session_id (FK), timestamp, role (user—assistant), content

REST API

Use Swagger UI for full details and interactive testing:

- Swagger UI: http://localhost:8000/docs
- ReDoc: http://localhost:8000/redoc

Endpoints:

- `GET /health`
- `GET /meta`
- `POST /query`
- `GET /logs`
- `GET /chats`

- GET /chats/{session_id}
- DELETE /chats/{session_id}

UI (Gradio)

- Chat interface with example question boxes
- “New Chat” creates a fresh session and clears local history
- Auto-refresh of chat list after sends/deletes

Data & Evaluation Assets

Combined Evaluation Metrics (CSV)

- **Path**: data/evaluate/combined_evaluation_metrics.csv
- **Purpose**: main evaluation artifact combining human ratings and automated metrics.
- **Key columns** (header examples): query, expected_answer, generated_answer, Relevance, Accuracy, Source_Citation, Fluency, Query_Length, Retrieval_Time_MS, Generation_Time_MS, Total_Time_MS, BLEU, ROUGE-1, ROUGE-2, ROUGE-L, METEOR, BERTScore-P, BERTScore-R, BERTScore-F1, Perplexity.

QA Files (Source for Evaluation)

- A set of 100 questions (QA files) in ./data/evaluation/test_questionsV2 directory was used to produce the rows in combined_evaluation_metrics.csv.

Performance & Sizing

- **Cold start**: initial model download and load can be long on first run; subsequent runs use cache.
- **GPU memory**: 8B class model; ensure sufficient VRAM. Monitor with nvidia-smi.