Dollie uvettah ‘s STUDY BOT PROJECT REPORT

GitHub Repository

https://github.com/chibi1245/study_chatbot.git

Render

https://study-chatbot-1.onrender.com

Api link

http://127.0.0.1:8000/

The Project Overview

This project is a Study Bot built using the following FastAPI, LangChain, Groq LLM, and MongoDBATLAS and deployed using render. The chatbot answers study-related questions and maintains conversation memory for each user by storing chats in MongoDB. This will allows multiple users to interact with the chatbot while still storing their individual conversation history using a unique user_id without losing the memory.

• FastAPI – for backend API framework

• LangChain – LLM framework

• Groq which I used (openai/gpt-oss-20b) – for the Language model

• MongoDB Atlas – Database for storing chat memory

• Uvicorn – ASGI server

2. Description of Memory Implementation

The memory is implemented using MongoDB. Each message in a conversation is stored in a database with the following fields: user_id, role, message, and the timestamp.

Storing Messages

After the response has been generated, both the user question and assistant response are stored in the MongoDB database with a timestamp. This is done to create permanent memory.

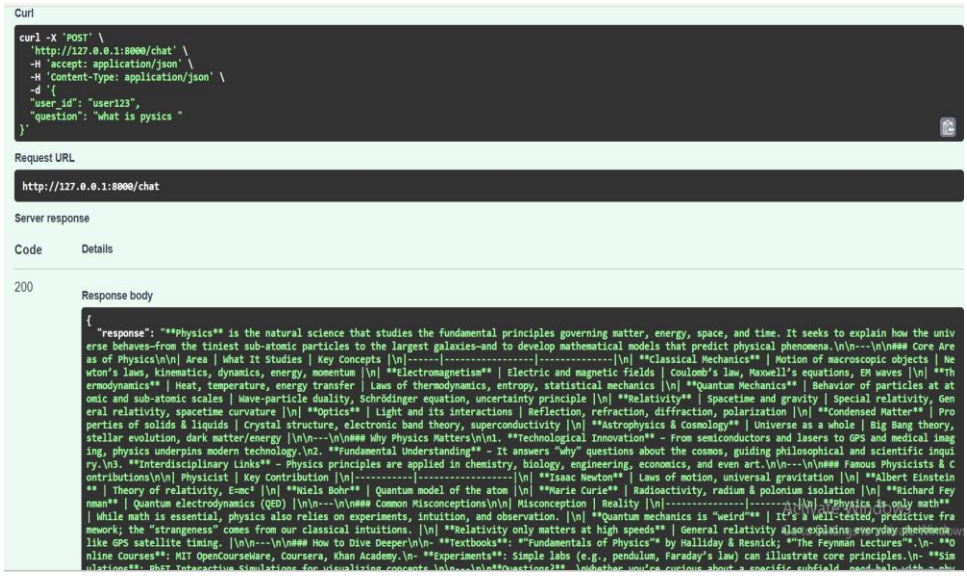
Retrieving Chat History

Before a new response is generated, the system retrieves all previous messages for that user in chronological order. The chat history is then passed to the language model.

Sending History to the Model

The chat history and current question are sent to the LLM using LangChain. This allows the chatbot to generate context-driven response

Sample pics for the api



Parameters

No parameters

Request body required

Edit Value | Schema

```
{
  "user_id": "user123",
  "question": "what is pysics "
}
```

Mongodb screenshot