Project Overview:

For this task, I built a simple chatbot that uses a free LLM API (Groq) to generate responses. The app is split into two parts:

- A FastAPI backend that handles requests and calls the LLM
- A Streamlit frontend that provides a basic chat interface for users

The full project is also containerized using Docker for easy deployment. All code and setup instructions are provided in the GitHub repository.

Project

Backend (FastAPI)

I chose FastAPI because it's simple, fast, and works well for JSON APIs. The backend does the following:

- Accepts POST /chat requests containing user messages.
- Forwards the message to the Groq LLM API (llama3-8b-8192 model).
- Returns the generated response, total tokens used, and detailed execution time.
- Adds structured logging to track requests, errors, and response time.

Frontend (Streamlit)

The frontend was built using Streamlit to quickly create a responsive UI. Features include:

- Input box for user questions.
- Chatbot responses streamed in real-time.
- Conversation history displayed during the session.
- Displays token usage (prompt, completion, total), Backend execution time and Model (Groq) response time

Containerization (Docker)

To make deployment easier, I used Docker to containerize the whole app. The Dockerfile installs the dependencies and starts both FastAPI and Streamlit servers.

To use docker:

docker build -t llm-chatbot .
docker run -p 8000:8000 -p 8501:8501 llm-chatbot

Then you can open:

"http://localhost:8501" to use the UI
"http://localhost:8000/docs" to test the API