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

I also containerized the project using Docker so it can be run easily anywhere. The full code and setup instructions are available in the GitHub repo.

Project

Backend (FastAPI)

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

- Accepts a POST request at /chat with a message.
- Sends that message to the Groq LLM API.
- Returns the chatbot's response back to the frontend.

I also added logging to help trace any errors or issues with API requests. This was helpful while debugging and is also part of the bonus requirements.

Frontend (Streamlit)

For the UI, I used Streamlit because it's lightweight and lets you build simple interfaces quickly.

The user can:

- Enter a question.
- See the chatbot's response.
- View previous messages in the same session (conversation history).

I also tried to track execution time and token usage, but Groq doesn't return token usage directly, so that part is marked as partially done.

Containerization (Bonus)

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 run the project in a container:
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