Local LLM Chat App using Streamlit, FastAPI, and Ollama (Mistral)

Overview

This project is a real-time local chatbot built using:

- Streamlit for the frontend chat interface

- FastAPI as the backend API layer

- Ollama for running the Mistral language model locally

It allows users to interact with the Mistral model in real-time without using internet-based LLMs.

Architecture

The architecture consists of three layers:

1. Frontend (Streamlit): Provides a web-based user interface where users can input messages and view model responses.

2. Backend (FastAPI): Receives messages from the frontend and forwards them to the local LLM.

3. Local LLM (Ollama with Mistral): Processes the messages using the Mistral model and generates appropriate responses.

All interactions occur on the local machine without relying on external APIs.

Setup Instructions

Follow the steps below to set up the project:

1. Install required Python libraries:

   - streamlit

- fastapi
- uvicorn
- requests

2. Download and install Ollama, then pull the Mistral model:
   - ollama pull mistral

3. Start the FastAPI server to handle chat requests.

4. Launch the Streamlit app to start chatting.

All components should run on the same machine.

Workflow Summary

1. User enters a message via Streamlit UI.
2. Message is sent to FastAPI backend.
3. FastAPI forwards it to Ollama's local API using the Mistral model.
4. Mistral generates a response.
5. Response is returned to FastAPI and displayed in Streamlit.

This setup enables fast, private, and offline interaction with an LLM.