Build an AI Chatbot Using LLMs

1. Abstract

This project focuses on designing and implementing a full-stack AI chatbot using Large Language Models (LLMs). The system integrates a Streamlit-based frontend with a FastAPI backend, which communicates with an LLM API to generate intelligent responses. The chatbot demonstrates real-world application of AI APIs, backend development, and frontend integration.

2. Objective

The objectives of this project are:

- To understand the working of Large Language Models
- To build a scalable backend using FastAPI.
- To design an interactive frontend using Streamlit.
- To integrate an LLM API for real-time chatbot responses.
- To follow modern software engineering practices.

3. System Architecture

The system follows a client-server architecture:

User

↓

Streamlit Frontend

↓ (HTTP POST Request)

FastAPI Backend

↓

LLM API (Groq – LLaMA 3.1)

↓

FastAPI Backend

↓

Streamlit Frontend

↓

User Response

Explanation:

- The user enters a message in the Streamlit UI.
- The frontend sends the message to the FastAPI backend.
- The backend forwards the request to the LLM API.
- The generated response is returned to the frontend and displayed to the user.

4. Technologies Used

Technologies	Purpose
Python	Programming Language
FastAPI	Backend RestAPI
Streamlit	Frontend UI
Groq LLM API	AI Response Generation
GitHub	Version Control and Docuentation

5. Design Decisions

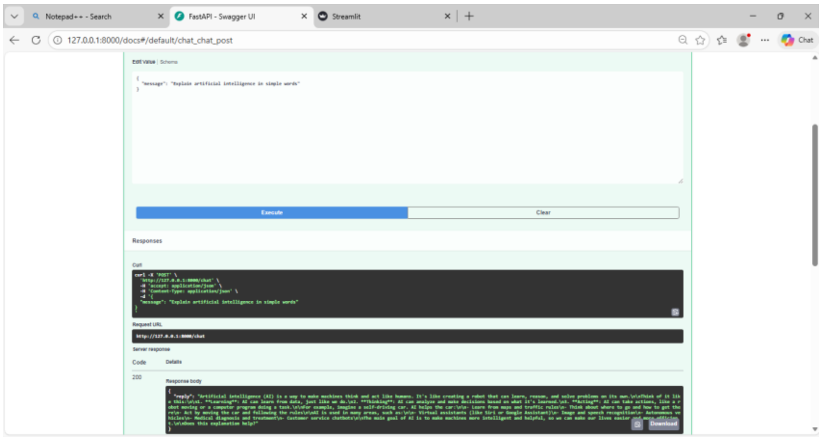
- **FastAPI** was chosen for its high performance and simplicity in building REST APIs.
- **Streamlit** was selected for rapid development of AI-based web interfaces.
- **Groq LLM API** was used instead of OpenAI to avoid paid API limitations.
- Modular folder structure was followed for better maintainability.
- REST API communication ensures separation of frontend and backend.

---

6. Implementation Details

- The backend exposes a /chat API endpoint using FastAPI.
- User messages are sent via HTTP POST requests.
- The LLM generates intelligent responses using LLaMA 3.1.
- Error handling is implemented to ensure system stability.

7. Screenshots


127.0.0.1:8000/docs#/default/chat_chat_post

Edit Value | Schema

```
{
  "message": "Explain artificial intelligence in simple words"
}
```

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/chat' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "message": "Explain artificial intelligence in simple words"
}'
```

Request URL

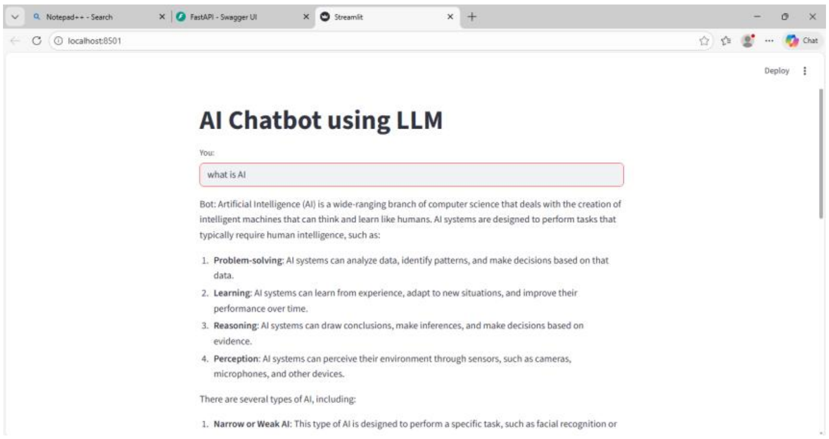
```
http://127.0.0.1:8000/chat
```

Server response

Code: 200

Response body

```
{
  "reply": "Artificial intelligence (AI) is a way to make machines think and act like humans. It's like creating a robot that can learn, reason, and solve problems on its own.\n\nThink of it lik
e this:\n\n1. **Learning**: AI can learn from data, just like we do.\n2. **Thinking**: AI can analyze and make decisions based on what it's learned.\n3. **Acting**: AI can take actions, like a r
obot moving or a computer program doing a task.\n\nFor example, imagine a self-driving car. AI helps the car:\n\n- Learn from maps and traffic rules\n- Think about where to go and how to get the
re\n- Act by moving the car and following the rules\n\nAI is used in many areas, such as:\n\n- Virtual assistants (like Siri or Google Assistant)\n- Image and speech recognition\n- Autonomous ve
hicles\n- Medical diagnosis and treatment\n- Customer service chatbots\n\nThe main goal of AI is to make machines more intelligent and helpful, so we can make our lives easier and more efficien
t.\n\nDoes this explanation help?"
}
```


localhost:8501

AI Chatbot using LLM

You:

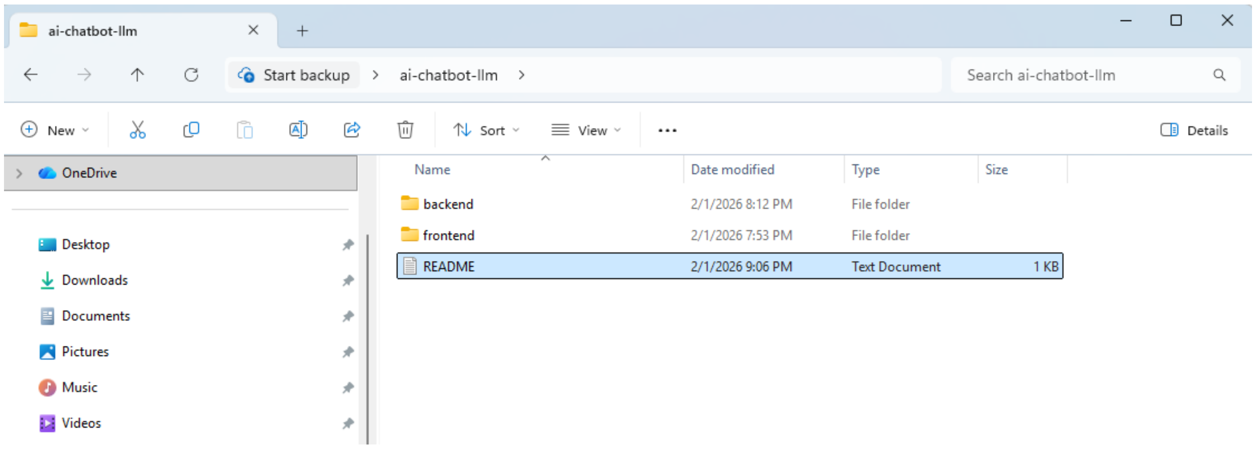
what is AI

Bot: Artificial Intelligence (AI) is a wide-ranging branch of computer science that deals with the creation of intelligent machines that can think and learn like humans. AI systems are designed to perform tasks that typically require human intelligence, such as:

1. **Problem-solving**: AI systems can analyze data, identify patterns, and make decisions based on that data.
2. **Learning**: AI systems can learn from experience, adapt to new situations, and improve their performance over time.
3. **Reasoning**: AI systems can draw conclusions, make inferences, and make decisions based on evidence.
4. **Perception**: AI systems can perceive their environment through sensors, such as cameras, microphones, and other devices.

There are several types of AI, including:

1. **Narrow or Weak AI**: This type of AI is designed to perform a specific task, such as facial recognition or



Name	Date modified	Type	Size
backend	2/1/2026 8:12 PM	File folder	
frontend	2/1/2026 7:53 PM	File folder	
README	2/1/2026 9:06 PM	Text Document	1 KB

8. Steps to Run the Project

Step 1: Install Dependencies

Code : pip install fastapi streamlit groq uvicorn

Step 2: Start Backend

Code: cd backend

python -m uvicorn main:app –reload

Step 3: Start Frontend

Code: cd frontend

python -m streamlit run app.py

Step 4: Use Chatbot

- Open browser
- Enter a message
- View AI response

9. Conclusion & Future Scope

The project successfully demonstrates a full-stack AI chatbot using LLMs. It highlights the integration of AI APIs with modern web frameworks. Future enhancements include user authentication, chat history storage, deployment to cloud platforms, and multi-language support.