# 🤖 LangGraph AI Chatbot Agent

An intelligent, friendly AI chatbot framework built with **LangGraph**, **FastAPI**, and **Streamlit**, supporting powerful LLMs from **OpenAI** and **Groq** with optional real-time web search via **Tavily**.

## 🌟 Features

- 🔁 Multi-model support: OpenAI (`gpt-4o-mini`), Groq (`llama-3.3-70b-versatile`, `mixtral-8x7b-32768`)

- 🌐 Real-time Web Search with Tavily

- ▯ Customizable system prompt to shape chatbot personality

- ▯ Natural Language Agent reasoning using LangGraph ReAct agents

- ⚡ FastAPI backend with clean API structure

- 📱 Streamlit-based frontend UI

---## 🚀 Quickstart

### 1. Clone the repository

```bash
git clone https://github.com/yourusername/langgraph-ai-agent.git

cd langgraph-ai-agent
```

2. Install dependencies

It's recommended to use pipenv, but you can use pip as well:

bash

CopyEdit

```
pip install -r requirements.txt
```

3. Set up environment variables

Create a .env file in the root directory and add your API keys:

env

CopyEdit

```
OPENAI_API_KEY=your_openai_key

GROQ_API_KEY=your_groq_key

TAVILY_API_KEY=your_tavily_key
```

Or you can export them manually in your terminal session:

bash

CopyEdit

```
export OPENAI_API_KEY=your_openai_key

export GROQ_API_KEY=your_groq_key

export TAVILY_API_KEY=your_tavily_key
```

4. Run the backend

bash

CopyEdit

```
python backend.py
```

- FastAPI server will start at http://127.0.0.1:9999
- Swagger docs available at http://127.0.0.1:9999/docs

5. Run the frontend

bash

CopyEdit

```
streamlit run frontend.py
```

- Open browser to http://localhost:8501 to access the UI

□ Project Structure

bash

CopyEdit

```
.

 ├── ai_agent.py       # Core LangGraph agent logic

 ├── backend.py        # FastAPI backend to serve agent API

 ├── frontend.py       # Streamlit UI for user interaction
 ├── models.py         # Pydantic schemas for request validation
 ├── .env              # API keys (not tracked by Git)
 ├── requirements.txt   # Python dependencies
└── README.md          # Project documentation
```

📫 API Endpoint

POST /chat

Request Body:

json

CopyEdit

```
{
  "model_name": "llama-3.3-70b-versatile",
  "model_provider": "Groq",
  "system_prompt": "You are a helpful assistant.",
  "messages": ["What's the capital of Canada?"],
  "allow_search": true
}
```

Response:

json

CopyEdit

```
{
  "response": "The capital of Canada is Ottawa."
}
```

□ Models Supported

Provider	Model
OpenAI	gpt-4o-mini
Groq	llama-3.3-70b-versatile
Groq	mixtral-8x7b-32768

🛠️ Requirements

- Python 3.9+
- langchain, langgraph, streamlit, fastapi, requests, pydantic, uvicorn, etc.

Install all via:

bash

CopyEdit

```
pip install -r requirements.txt
```

📌 TODO / Coming Soon

- 🗨️ Multi-turn chat history with memory
- 💬 Streaming responses
- 🌍 Deployment on cloud (Render, Vercel, or Hugging Face Spaces)
- 🔒 Authentication and user history logs