# **Little Agent Chatbot**

A simple, local-first AI agent and RAG chatbot powered by Ollama, LangChain, and Gradio. This project is designed to be a lightweight, accessible starting point for anyone interested in exploring local Large Language Models (LLMs) and building their own conversational AI agents.

## **✨ Features**

* **💻 Run Locally:** Built to run entirely on your own machine. Your data and conversations stay private and secure.
* **🤖 Powered by Ollama:** Easily integrates with [Ollama](https://ollama.com/) to run powerful open-source LLMs locally, even on consumer-grade hardware.
* **🧠 Qwen2 Model:** Utilizes the efficient and capable qwen2 model (e.g., qwen2:1.7b or qwen2:4b), which offers great performance on modest hardware setups.
* **🔗 LangChain Integration:** Leverages the [LangChain](https://www.langchain.com/) framework to create intelligent agents that can do more than just chat.
* **📄 Retrieval-Augmented Generation (RAG):** Enhance your chatbot's knowledge by allowing it to retrieve information directly from your own PDF documents using nomic-embed-text.
* **🌐 Simple Web Interface:** A clean and straightforward user interface built with [Gradio](https://www.gradio.app/), making it easy to interact with your agent.
* **🛠️ Extensible Agent Capabilities:** The agent architecture allows for easy extension. For example, you can give it tools to:
  + Fetch real-time data from the web (e.g., live weather updates).
  + Perform accurate mathematical calculations.
* **🎓 Educational & Developmental:** This project serves as a perfect foundation for learning about local LLMs or as a launchpad for developing more complex agent-based applications.

## **🚀 Getting Started**

### **Prerequisites**

Before you begin, ensure you have [Python](https://www.python.org/downloads/) (3.8 or higher) and [Ollama](https://ollama.com/) installed and running on your system.

### **Installation**

1. **Clone the Repository**  
   git clone https://github.com/your-username/Little\_Agent\_Chatbot.git  
   cd Little\_Agent\_Chatbot
2. **Set Up a Virtual Environment (Recommended)**  
   python3 -m venv .venv  
   source .venv/bin/activate  
     
   *On Windows, use .venv\Scripts\activate*
3. **Install Dependencies**  
   pip install langchain langchain-community langchain-core langchain-ollama chromadb sentence-transformers pypdf python-dotenv unstructured[pdf] tiktoken gradio
4. Download the LLM and Embedding Models  
   Use the Ollama CLI to pull the necessary models. qwen2 is the main language model, and nomic-embed-text is used for document embeddings (RAG).  
   # Pull a language model (choose one)  
   ollama pull qwen2:4b  
   # OR for lower-spec hardware  
   ollama pull qwen2:1.7b  
     
   # Pull the model for embeddings  
   ollama pull nomic-embed-text
5. Add Your Data  
   Place any PDF documents you want the chatbot to use for RAG into a designated data folder (you may need to create this folder).

### **Running the Chatbot**

1. **Ensure Ollama is running.**
2. **Launch the application:**  
   python app.py
3. Open your web browser and navigate to the local URL provided by Gradio (usually http://127.0.0.1:7860).

## **🔧 How It Works**

This project combines several key technologies to create a cohesive chatbot experience:

1. **Ollama:** Serves the open-source LLM (qwen2) and the embedding model (nomic-embed-text) locally, making them accessible as API endpoints.
2. **LangChain:** Provides the core framework for building the application. It orchestrates the flow between the user, the retrieval system, the agent, and the LLM.
3. **RAG Pipeline:** When you ask a question, the system first uses nomic-embed-text to create embeddings of your query and search your local PDF documents for relevant information. This context is then "augmented" or added to your prompt.
4. **Agent Executor:** The agent is equipped with tools (like a calculator or a web search tool). It uses the LLM to reason about which tool to use based on your query.
5. **Gradio:** Creates the user-friendly web interface that you interact with, managing inputs and displaying the chatbot's responses.

## **📜 License**

This project is released under the **MIT License**. See the [LICENSE](https://www.google.com/search?q=LICENSE) file for more details. Feel free to use, modify, and distribute it as you see fit.