Ahsan AI Courses Chatbot (Streamlit)

FastAPI + Streamlit + BOT + RAG

Overview

This project represents an intelligent chatbot designed to operate within the domain of Artificial Intelligence education. It covers various AI-related courses such as “AI Automation”, “Data Science”, “Agentic AI”, and “Generative AI”. Users can interact with the chatbot to learn more about these courses, request detailed information, and express interest in enrollment. The system also captures leads (user name, email, and course of interest) for future engagement. A structured knowledge base underpins the chatbot, storing comprehensive course information. The chatbot utilizes a Large Language Model (LLM) via the GEMINI API to generate context-aware and meaningful responses based on this knowledge base.

Technology Stack

- **Backend:** FastAPI — Used to define and manage API endpoints. - **Frontend:** Streamlit — Provides an interactive chat UI and lead capture form. - **Database:** SQLite (or SQLAlchemy with SQLite) — Stores chat history and captured leads. - **Knowledge Base Ingestion:** Markdown or text files containing course data are converted into vector embeddings for retrieval-augmented generation (RAG). - **LLM / Generative API:** GEMINI API — Powers the AI conversational model. - **Session Management:** Each user session is identified by a unique `session_id` generated via the frontend, enabling chat history tracking.

System Flow

1. **Session Initialization:** The user opens the Streamlit app, initiating a new session with a unique `session_id`. 2. **Interaction Modes:** - **Chat:** The user sends messages through the chatbot interface. - **Lead Capture:** The user fills out a form expressing interest in a course. 3. **Message Handling:** When a chat message is sent, the frontend sends a payload (`session_id` + message) to the `/chat` endpoint. 4. **Backend Chat Flow:** - Relevant documents are retrieved from the knowledge base via the `get_relevant_docs()` function. - The LLM generates an appropriate response based on retrieved context. - The user query and AI response are stored in the `chat_history` table linked with the `session_id`. 5. **Lead Handling:** Lead information (name, email, interest) is sent to the `/lead` endpoint and stored in the `leads` table. 6. **Admin and Monitoring:** - `/admin/leads` — View all captured leads. - `/kb/status` — Check knowledge base load status. 7. **Knowledge Base Ingestion:** Markdown files containing course descriptions are processed at startup, embeddings generated, and stored for retrieval. 8. **Data Storage:** All data (chat history, leads) is persisted in a local SQLite database.

FastAPI Backend Endpoints

- **GET /** → Returns a welcome message. - **GET /health** → Returns system health status. - **GET /kb/status** → Displays the knowledge base load state and number of documents. - **POST /chat** → Processes user messages and returns AI-generated replies. - **POST /kb/search** → (Optional) Directly searches the knowledge base. - **POST /lead** → Captures user lead information. - **GET /admin/leads** → Displays all stored leads for administrative review.

Summary

The Ahsan AI Courses Chatbot serves as a **virtual course advisor** specializing in AI-related educational programs. It intelligently answers user queries, retrieves course-specific insights from a knowledge base, and records potential leads for course enrollment. By integrating FastAPI, Streamlit, RAG-based retrieval, and GEMINI-powered language models, it provides an end-to-end intelligent counseling system for prospective AI learners.