Ahsan AI Courses (Chatbot)

FastAPI + Streamlit + BOT + RAG

Overview

The **Ahsan AI Courses Chatbot** is an intelligent conversational assistant designed to provide detailed information about various AI-related courses such as *AI Automation*, *Data Science*, *Agentic AI*, and *Generative AI*. The chatbot engages with users to answer course-related questions and captures leads by collecting user details such as email, phone number, and course interest. The system includes an integrated knowledge base containing structured course information. Using this knowledge base, the chatbot leverages a Large Language Model (LLM) to generate contextual and meaningful responses. When the chatbot initiates a conversation, it asks whether the user wishes to enroll. If the user agrees, a registration form appears where the user submits their details. This data is then securely stored in a database, marking the user as enrolled. Additionally, the chatbot maintains conversation memory, ensuring continuity in user interactions.

Tools Stack

Backend: FastAPI — defines the API endpoints and handles server-side operations.
Frontend: Streamlit — provides an interactive user interface for the chat system and enrollment form.
Database: SQLite (via SQLAlchemy) — stores chat history and user lead information.
Knowledge Base Ingestion: Course details are stored in Markdown or text format. These documents are embedded into vector representations for semantic retrieval (RAG — Retrieval Augmented Generation).
LLM Integration: Uses the GEMINI API for generating context-aware responses.
Session Management: Each user is assigned a unique `session_id` for maintaining chat history and state management across sessions.

System Flow

1. **Frontend Interaction:** The user opens the Streamlit application, and a unique `session_id` is generated for tracking their session.
2. **Chat or Lead Form:** The user can either chat with the bot or fill out a course interest form.
3. **Enrollment Process:** If the user wishes to enroll, a form is displayed for personal details (name, email, phone, course interest), which are then stored in the database.
4. **Chat Handling:** The frontend sends chat messages to the `/chat` endpoint with

payloads containing `session_id` and message data.
5. **Backend Processing:**
- Retrieves relevant information from the knowledge base using the `get_relevant_docs()` function.
- Generates a meaningful response using the LLM and contextual information.
- Saves both the user query and chatbot response in the `chat_history` table linked to the `session_id`.
6. **Frontend Display:** The chatbot's response is displayed in the chat interface while preserving the chat history.
7. **Lead Management:** Form submissions are handled through the `/lead` endpoint and stored in the `leads` table.
8. **Admin Endpoints:**
- `/admin/leads` — View all collected leads.
- `/kb/status` — Check the knowledge base ingestion and loading status.
9. **Knowledge Base Ingestion:** Markdown or text course files are processed into embeddings for efficient retrieval during chat responses.
10. **Database Management:** SQLite manages persistent storage for chat sessions and user leads. The `session_id` ensures that each user's chat remains isolated and traceable.

Summary

In essence, the **Ahsan AI Courses Chatbot** acts as a virtual AI course counselor, designed to intelligently interact with users seeking AI-related education. It provides detailed, context-aware responses to user inquiries, guides users through enrollment, stores user leads and chat history, and ensures a personalized experience through session memory management.