Constitution Study Chatbot

This project is part of the Digital Pioneer of Egypt initiative. It is a chatbot designed specifically for law students who want to study the Egyptian Constitution.

# Team Members

- Abdelrahman Mohamed Mahmoud  
- Abdelrahman Mohamed Ali  
- Kenzy Mohamed  
- Mohamed Tarek  
- Marwan Tamer Mahmoud  
- Bassant Tarek

# Group: ALX1\_AIS2\_S1e

# Track: Generative AI Professional

# Under the Supervision of

Mohamed Agoor

# Project Description

The Constitution Study Chatbot is a generative AI-powered system designed to facilitate law students' interaction with the Egyptian Constitution. This innovative chatbot allows users to upload a PDF version of the Constitution and ask it questions related to the document's contents. Using advanced technologies such as ChromaDB for data storage and LlamaIndex for querying, the chatbot provides accurate and context-sensitive responses in Arabic, tailored to the needs of students and legal practitioners.  
  
The project stems from the need to modernize the study of legal texts by offering interactive, on-demand legal assistance. The chatbot operates through a user-friendly interface powered by Streamlit and offers seamless interaction via endpoints established with FastAPI. The key strength of the system is its ability to dynamically retrieve and generate legal information, ensuring that users receive accurate, context-aware responses to their constitutional queries.

# Importance of the Project

In the legal field, access to accurate and timely information is paramount. However, traditional methods of studying the Constitution—such as reading and interpreting lengthy texts—can be time-consuming and inefficient, especially when students or legal professionals are searching for specific clauses or laws.  
  
This chatbot offers a revolutionary solution by providing:  
1. Instant Legal Assistance: Instead of manually searching through legal texts, users can now get quick answers to specific constitutional queries.  
2. Enhanced Learning Experience: Law students benefit from a more interactive, engaging, and accessible way to study complex legal materials.  
3. Real-Time Interactions: By providing dynamic, context-aware responses, the chatbot mimics a real-time conversation with a legal expert.  
  
Additionally, this project contributes to the ongoing modernization of legal education, aligning with global trends toward digital transformation and AI-driven solutions in various sectors.

# Problem the Project Solves

The primary problem addressed by the Constitution Study Chatbot is the difficulty in efficiently accessing and interpreting specific sections of the Egyptian Constitution. Legal texts, especially constitutions, are extensive, complicated, and written in formal language, which makes them hard to navigate for non-experts.  
  
Key issues the chatbot solves:  
- Time-Consuming Manual Search: Traditionally, finding a specific article or understanding its context can take hours of study. The chatbot allows for instant retrieval of relevant sections.  
- Difficulty in Interpretation: Legal language is often complex and requires proper contextual understanding. The chatbot generates simplified, yet contextually accurate answers to users’ questions.  
- Lack of Personalization: Traditional study methods do not adapt to the user’s specific needs. This chatbot personalizes responses based on the user’s input, helping with targeted learning and comprehension.

# Learning Objectives

Through this project, both developers and users will achieve the following learning objectives:  
1. Understanding Legal Text Processing: Developers will learn how to preprocess and tokenize large legal texts like the Egyptian Constitution for efficient querying.  
2. AI-Powered Query Systems: Students will learn to integrate AI models like LlamaIndex to retrieve legal information and generate context-aware responses.  
3. Application of RAG Models: Develop an understanding of Retrieval-Augmented Generation models and their application in solving complex queries.  
4. Natural Language Processing (NLP): Gain proficiency in using NLP techniques for analyzing and generating text responses that are contextually aware.  
5. Real-World AI Integration: Learn to integrate various tools (ChromaDB, FastAPI, Streamlit) into a cohesive, user-friendly AI system for practical applications.  
6. Project Lifecycle Development: Understand the steps involved in developing, testing, and deploying a chatbot system.

# Tools and Algorithms

The following tools and algorithms are essential to the development of the Constitution Study Chatbot:  
  
- ChromaDB: A powerful vector database used for storing processed textual data from the Constitution. This allows for fast and efficient retrieval of information.  
- LlamaIndex: A retrieval-based language model that is designed to handle specific and context-rich queries. It is responsible for generating accurate and context-aware responses based on the Constitution.  
- RAG (Retrieval-Augmented Generation) Model: Combines both retrieval and generation models to improve the accuracy and relevance of the chatbot’s responses.  
- FastAPI: A high-performance API framework used to create the endpoints for uploading PDFs and querying the chatbot.  
- Streamlit: A Python-based library for building custom web applications, used to develop the user interface for the chatbot, ensuring a smooth interaction flow.  
- NLP (Natural Language Processing): Used extensively in tokenization, data cleaning, and in ensuring the chatbot understands and processes user input in Arabic.

# Steps to Apply the Project

**Step 1: Prepare the Environment**  
- Ensure you have a Python environment set up with all the necessary dependencies. Use the provided requirements.txt file to install them:  
```bash  
pip install -r requirements.txt  
```

**Step 2: Data Preprocessing**  
- Upload the Egyptian Constitution in PDF format.  
- Tokenize, clean, and process the data using NLP techniques to convert it into a structured form that can be queried.  
- Store the processed data in ChromaDB to allow for fast and accurate retrieval.

**Step 3: Implement Chatbot Development**  
- Develop the core chatbot functionality by integrating the RAG Model and LlamaIndex.  
- Train the model on the Egyptian Constitution to fine-tune it for answering legal questions.  
- Test the chatbot’s ability to provide accurate, context-aware answers to legal queries.

**Step 4: Create FastAPI Endpoints**  
- Develop two primary endpoints using FastAPI:  
 - A POST endpoint (/upload\_pdf) to allow users to upload their own PDFs for processing.  
 - A GET endpoint (/chat) to interact with the uploaded PDF and provide answers to users’ queries in real-time.  
  
**Step 5: Build the User Interface (UI)**  
- Use Streamlit to create a simple and intuitive user interface that allows users to interact with the chatbot.  
- Implement a chatbox where users can input their questions and receive responses from the chatbot in real-time.  
  
**Step 6: Test and Refine**  
- Test the chatbot with various legal queries, including complex questions that require context from multiple sections of the Constitution.  
- Implement any necessary refinements to the query model, such as improving prompt engineering for more accurate responses.  
**Step 7: Deploy the Chatbot**  
- Once tested, deploy the chatbot to a cloud service or local server, making it available for students and legal professionals to use.  
- Continuously monitor performance and gather feedback to enhance the system.

# How to Run the Chatbot

1. Install the Required Dependencies:  
 Make sure all required libraries are installed by running the following command in your terminal:  
 ```bash  
 pip install -r requirements.txt  
 ```  
  
2. Start the FastAPI Server:  
 Launch the FastAPI backend to handle the PDF upload and interaction requests:  
 ```bash  
 uvicorn endpoint:app --reload  
 ```  
  
3. Run the Streamlit UI:  
 Start the Streamlit front-end interface, which allows users to interact with the chatbot:  
 ```bash  
 streamlit run ui.py  
 ```