Problem Statement - AI/ML Olympiad: Education Assistant

Objective:

Develop an AI-powered, multimodal chatbot to assist medical students in learning anatomy. Using the "Anatomy and Physiology Vol 1 to 3" textbooks, the chatbot should answer questions and retrieve relevant text, images, and tables, enhancing the overall learning experience.

Textbooks Link: Click Here

Checkpoints for Phase - I:

1. **Data Extraction and Cleaning**: Extract relevant text, images, and tables from the provided books and ensure the data is cleaned before storing it in the database.

2. **Text Retrieval Accuracy**: Enable the chatbot to retrieve and deliver relevant textual content in response to user queries, ensuring accurate and contextually appropriate information.

3. **Support for Probing Questions**: Implement a system where the chatbot can ask clarifying questions when uncertain about a query, improving its understanding before providing an answer.

4. **Retrieval of Relevant Image and Table Content**: Ensure that the chatbot can identify and fetch relevant images and tables related to user queries, enhancing the depth and clarity of responses.

5. **Latency Optimization:** Focus on minimising response time, ensuring that the chatbot delivers answers quickly and efficiently to maintain a seamless user experience.

Final Deliverable for Phase - I:

A fully functional educational assistant chatbot with a simple yet effective user interface, capable of retrieving and presenting relevant information from a mix of text and visual data. The chatbot should handle complex queries and provide accurate and helpful responses to support students in their studies.

Checkpoints for Phase - II:

1. **Image-Based Prompts:** Enable the chatbot to understand and respond to queries based on images provided by the user, enhancing its ability to handle visual data inputs.

2. **Flow Chart Generation for Relevant Queries:** Implement a feature that allows the chatbot to generate flowcharts for specific types of queries, providing visual aids that help users better understand complex information.

3. **Table Reconstruction:** Develop the ability for the chatbot to reconstruct and present tables based on the information provided or retrieved, ensuring clear and structured data representation.

4. **Answering Reference Questions at the End of Each Chapter (PDF File):** Create a system where the chatbot can accurately answer reference questions found at the end of textbook chapters, delivering responses in a PDF format for easy access and review.

5. **Lesser Latency:** Further reduce response times to ensure even quicker and more efficient interactions, enhancing the overall user experience.

Final Deliverable for Phase - II:

An enhanced educational assistant chatbot with advanced capabilities, including image-based prompt recognition and flowchart generation. The chatbot will deliver faster responses with reduced latency, accurately reconstruct tables, and answer reference questions from textbook chapters, providing comprehensive support in PDF format. These features will further empower students by offering detailed, visually enriched, and contextually appropriate assistance for their anatomy studies.

Tech Stack

- Programming Language: Python (version 3.7 or above)
- Language Models: Proprietary models like GPT or open-source LLMs
- Environment: Local or Cloud (e.g., Google Colab)
- Technology: Retrieval Augmented Generation (RAG)
- Database: Open-source VectorDB
- Modeling: Fine-tuning (optional)
- User Interface: Streamlit or Gradio (UI) or any basic UI
- Operating System: Windows (preferred), Linux

Guidelines

- Code Documentation: All code must be documented in `.py` files and accompanied by a README file that includes steps for configuration and execution.
- Dependencies: A `requirements.txt` file should be included with all dependencies, and the entire repository should be provided in a zip format.
- File Sharing: Do not share large database files or model files. Only share scripts needed to recreate them on any local system.

Note:

1. **Book Integration:** Using at least one book is mandatory. While it's not necessary to use all three volumes, integrating multiple books could be advantageous for your team during the final evaluation.
2. **User Interface:** A simple UI is sufficient as the frontend will not be a factor in the evaluation.
3. **Deployment**: Deploying the app will be considered an advantage.