NLP Challenge:

Personalized Learning Assistant

Objective:

Create a personalized learning assistant that helps you by answering questions, or by providing tailored summaries, or more detailed explanations (like the course's discussion topics!), or even quizzes you like your professor would, and all largely based on a corpus of personal content course documents, with a user-friendly interface, build in Streamlit.

Build the tool yourself (probably with the technologies listed below), so don't use all-in-one solutions like AnythingLLM and the likes. When in doubt if you can use a certain technology (because the field of NLP is changing at an explosive pace), ask your instructor.

Must have Components:

Streamlit, crewAI multi-agent framework, RAG, vector database, LLM (Groq API)

Application MVP:

- Streamlit user interface
    - Home page, upload section to add extra materials, dropdown to select which LLM you want to use (default should be the Groq API), chatbot interface that provides the requested answers (with links to the referenced materials)
- Content ingestion agent
    - Collect and preprocess added reference materials in the form of multiple pdf documents
    - Store processed content in the vector database
- Question answering agent
    - Answering questions by retrieving relevant information from the vector database, using RAG to generate accurate and contextually relevant information
- Other agents are also possible (these are considered extensions).
    - Maybe like a CheatSheet agent (= one page summary agent), or a Quiz (Exam) agent (multiple choice questions, or even more open questions)?

Possible Extensions:

- Implement multiple possible LLMs to select from (like **OpenAI** and others), but this also entails you have multiple types of embeddings you should store in your vector database (depending on the LLM used)
    - Extra extension: why not use a suitable model from **HuggingFace**? At first you can use the HuggingFace API to do the inferencing, but as extra credit extension, why not run the HuggingFace model on your own hardware (if you have suitable hardware), using **ollama**?
- Instead of using the default build-in local ChromaDB, use an external different vector database, like **FAISS, or Pinecone, or Qdrant**
- Extend the agent framework, to enrich the content ingestion with different types of content, like YouTube movies or other types of content, like for instance a web search
    - Or even go further, and really go for extracting information from powerpoints, including the graphs and images in there (hint: maybe via docling?)
    - Or even go for a full data orchestration framework, like **LlamaIndex**
- Adding (short-term) memory to your chat conversation, so you can reference earlier topics from the same chat-session. You can further extend this with other types of memory, like long term (to really personalize the response based on previous interactions), entity, or even contextual memory

Extra Credit Extension:

- Extending the chat interface to handle voice
    - Speech to Text model (STT) so you can turn voice into text, and it to the RAG system to give you an answer
        - Look for models that can handle automatic endpointing of the conversation
    - Text to Speech model, so the textual answer of the LLM can be turned back into voice
        - Look for models that can handle streaming data, so when you already have a part of the text it can already turn it into audio, so it doesn't have to wait until the entire textual answer is given (it might be a long text)