LLM Apps

Full-stack starter with create-llama

Intro

- "create-llama" is a tool developed by LlamaIndex to streamline the creation of RAG full-stack application templates.

- Limitations:
  - Does not include a database to host private documents. Private documents are manually included in an app directory.
  - Therefore, it does not include CRUD functionality to manage private documents.

- Currently allows the creation of three types of full-stack applications:
  - Option 1: Frontend and Backend with Next.js serverless.
  - Option 2: Frontend with Next.js and Backend with FastAPI (Python).
  - Option 3: Frontend with Next.js and Backend with Express (Javascript).

Option 1: frontend and backend with Next.js serverless

- It is the simplest option.
- Only requires deployment on Vercel.
- Limitations: it will fall short when we try to scale the application.

Option 1: how does Llamandex use it?

- The RAG logic of LlamaIndex is in:
  - app/api/chat/engine/index.js

- Written in Typescript

Option 2: frontend with Next.js and backend with FastAPI

- The LlamaIndex tutorial suggests deploying both frontend and backend on Render.com

- I don't think it's a good idea, as it will limit us when scaling the frontend of the application. It seems more appropriate to deploy the frontend on Vercel and the backend on Render.com as we did with the ToDo application.

Option 2: how does LlamaIndex use it?

- The RAG logic of LlamaIndex is in the backend folder:
  - backend/app/utils/index.py

- Written in Python

Option 3: Frontend with Next.js and backend with Express

- Considering that LlamaIndex, as well as LangChain and the ChatGPT API, are natively developed in Python, I believe it is most advisable to specialize in Backend with FastAPI (Python) instead of Express (Javascript).

- For this reason, we will not cover this third option in the course, although it may be a good alternative for developers with a background in Javascript or working in teams specialized in Javascript.