For use, to a post request to https://fastapi-production-7244.up.railway.app/ with your openai access token

We uses langchain, chatgpt, and frontend deployment to create a RAG chatbot which answers questions based on 171 documents. The prompt template is set up to provide an answer to users question along with the subsection the answer was found in. The prompt also includes saying I don't know if it's not sure. Langchain memory was also used to have continuous conversation with the user.

In addition we tested chatgpt4, cohere RAG, bingchat to test out how other solutions performed. With the help of our legal expert we had these observations:

* All tested language model aren't good enough to perform high accuracy RAG

* Sometimes a model hallucinated a quote and sometimes it completely misunderstood the question or the legislation. GPT4 mentioned a certain document had a quote when in fact it didn't. When confronted, gpt4 made up a reason as to why it said something existed when it fact it didn't.

* Cohere solution was very neat in that it provided hyperlinks to documents it was quoting on. Yet it made up stuff or did not understand the content of documents properly

* bingchat overall performed better than gpt4 and cohere RAG but still hallucinated or misunderstand the legislation.

Hence we created custom chatbot. Feel free to use it and provide feedback. We did check legal LLMs on huggingface although there were not chat legal LLMs and most of the legal models were fill-mask. We do believe the best solution would be a fine-tuned legal LLM based on a large legal dataset.

Data is title 15A of New Jersey law downloaded from justia.com with selenium. 171 individual documents are the sub-sections of title 15A. (https://law.justia.com/codes/new-jersey/2022/title-15a/)