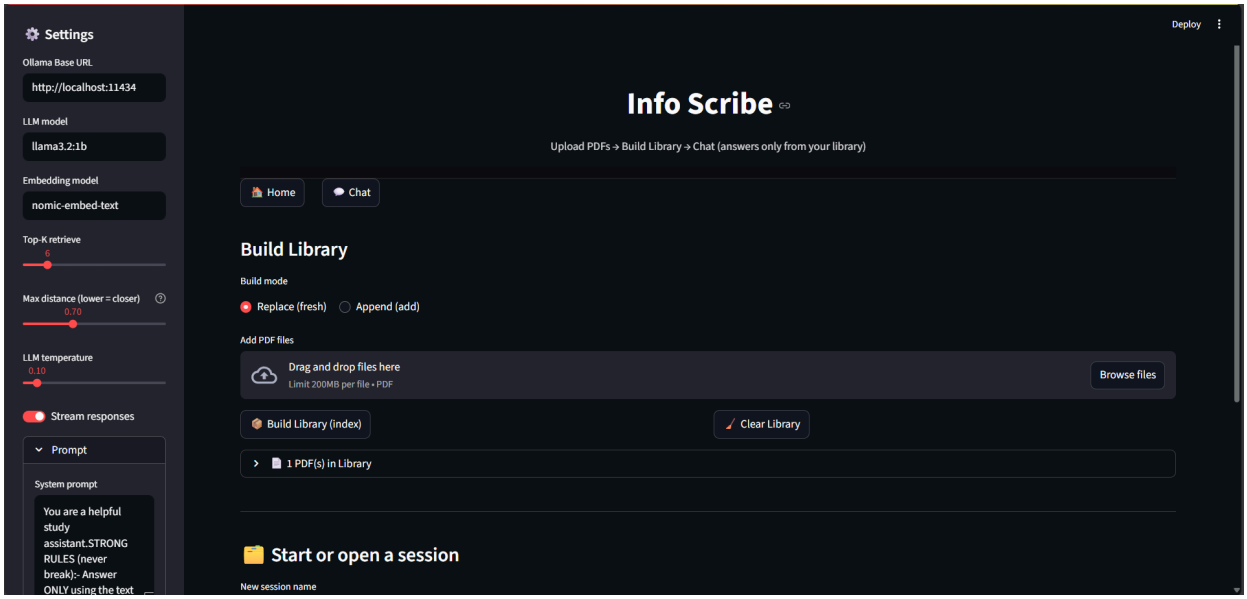
INFOSCRIBE

Here are the screenshots of my application **InfoScribe**, where I have poured in all my creativity and effort to design a **user-friendly PDF-based context reading chatbot**.
I have made the UI with 2 Tabs , One is home tab where user will be given options to browse for pdf , Build a library
I made a **Build Library** area where I can upload PDFs and turn them into a searchable knowledge base for the chatbot. I gave myself two modes:
Replace (fresh): I wipe the old PDFs and the old index, then rebuild everything from the newly uploaded PDFs. I use this when I want a clean start.
Append (add): I keep the existing PDFs, add new ones, and then rebuild the index over **all** PDFs currently in the folder. (Right now it's a full reindex, not incremental. That keeps things simple and consistent.)
When I click **Build Library (index)**, the app:
Reads each PDF page and extracts text ->Splits pages into chunks (so the search works well on smaller pieces) ->Creates vector embeddings for each chunk -> Saves a **FAISS** index so I can do fast semantic search at question time.

Other option in Home Tab is Session .



Settings Sidebar — What I Have Done and Why

In the sidebar of **InfoScribe**, I have added all the main controls that make the chatbot flexible and useful.

User can experiment with the parameters to get desired output

Here's what each one means:

Ollama Base URL
I have added this so the app knows where to connect to the Ollama server.
If Ollama is running locally, I can use the default `http://localhost:11434`.

LLM Model

I have given an option to choose which Large Language Model to use for answering questions.
For example, I can set it to `llama3.2:3b` for better quality answers.

Embedding Model
This is the model I use to turn PDF text into numerical vectors for searching.
I have made it editable so I can switch models depending on my needs.

Top-K Retrieve
This decides how many best-matching chunks from the PDFs will be given to the AI for answering.
I have kept it adjustable so I can choose between fewer, more accurate results or more, broader coverage.

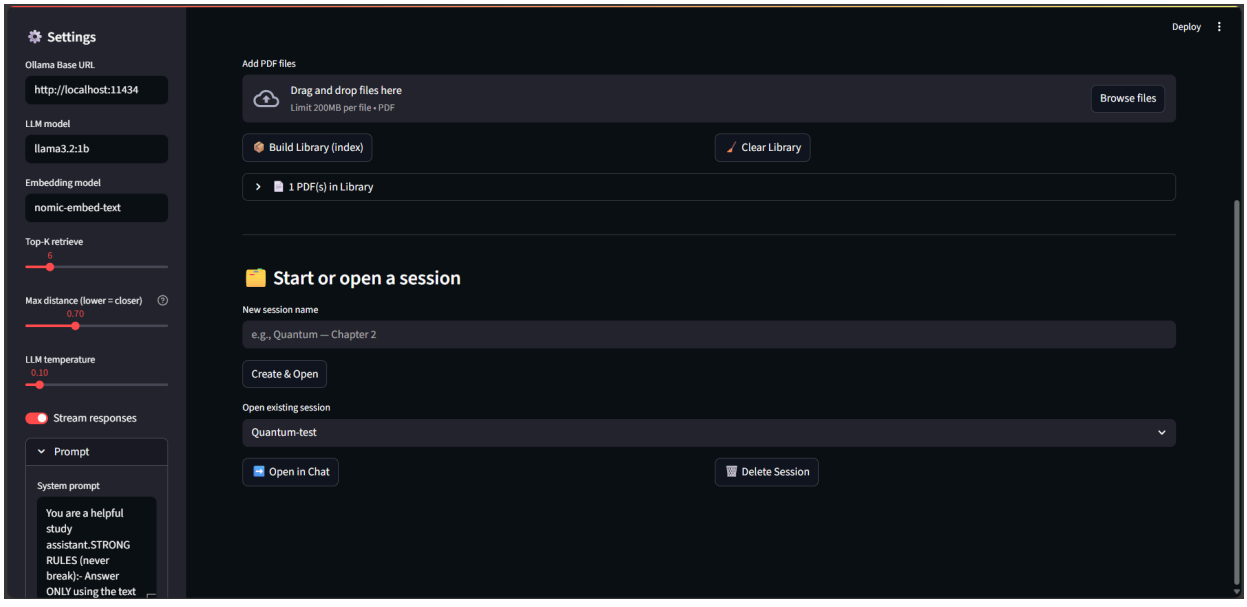
Max Distance
This sets how close in meaning the search results must be to my question.
Lower values mean only very relevant text is used; higher values allow looser matches.

Temperature
This controls how creative the AI is.
I keep it low for factual answers from PDFs, but I can raise it if I want more variety in wording.

Stream Responses
I have added this so I can choose if answers appear word-by-word in real time, or all at once after processing.

System Prompt

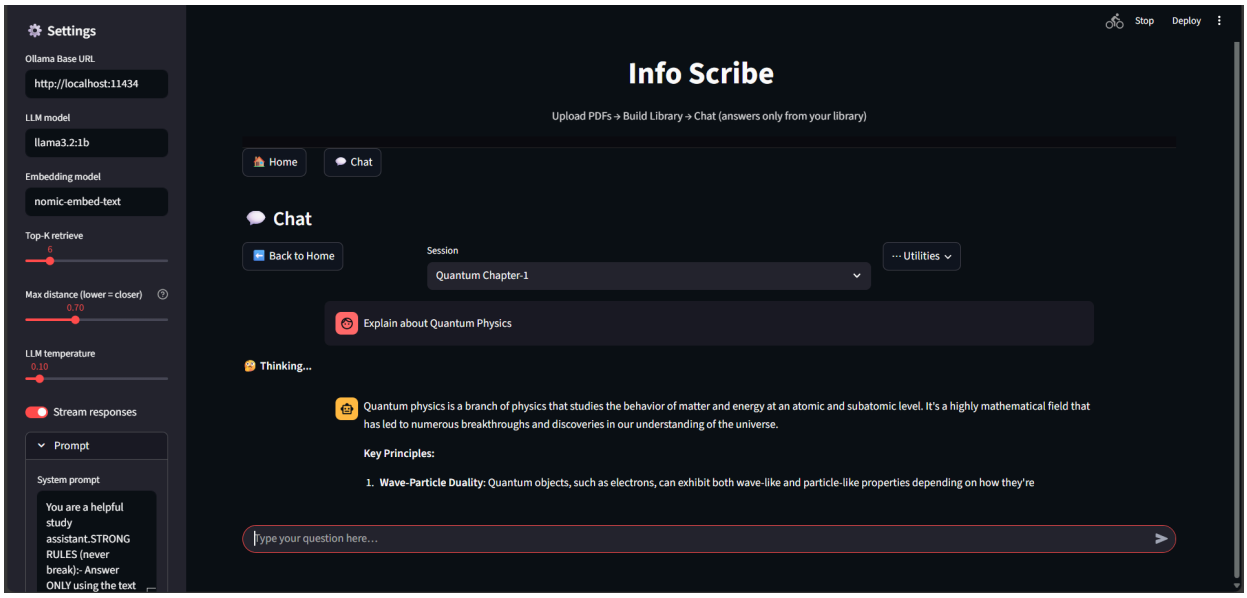
One can make it more strict or less



Sessions

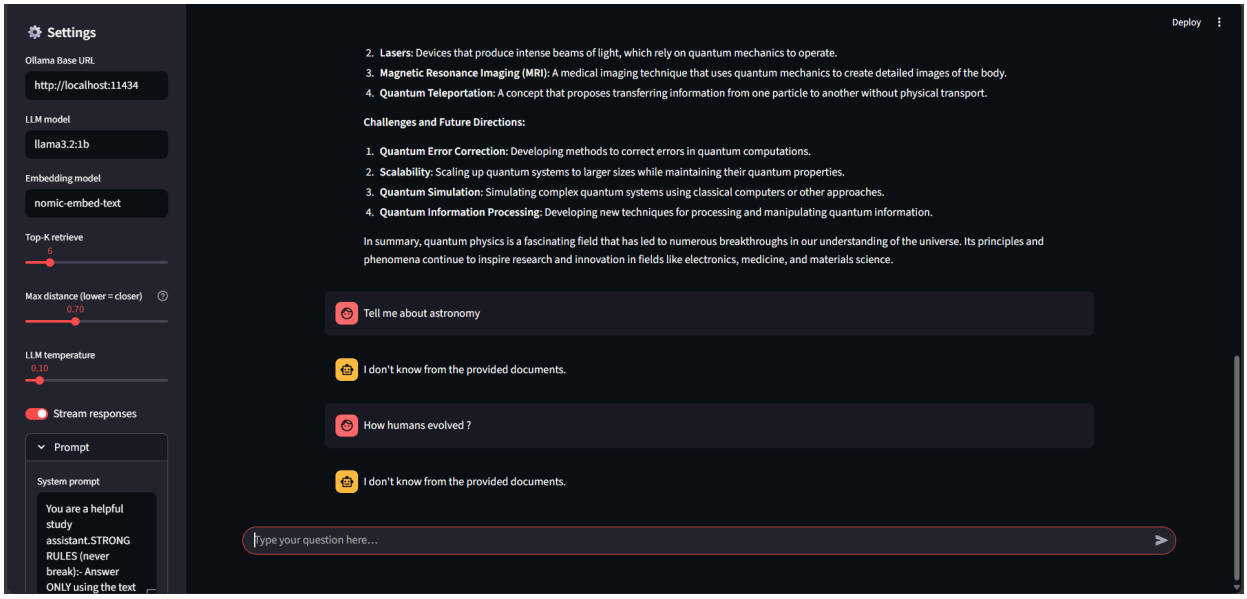
I have added a session system so I can keep different chats separate and organized. This allows me to have one chat for one topic, like *Quantum Physics*, and another for a completely different topic, like *Machine Learning*, without mixing the conversations. When I click “Create & Open,” the app generates a unique ID for that session, and I can give it a custom name (for example, *Chapter 2 Notes*) so I know what it’s about. The app saves this name, ID, and the creation date in a sessions list file so I can load it later. Every message I send and every answer from the bot is stored in a JSON file named after that session’s ID, which means when I open the same session again, I can see my past conversation exactly as I left it. I can easily pick an existing session from a dropdown and continue chatting without losing any messages, or delete it if I no longer need it — in which case the app removes both the chat file and its name from the session list.

User can switch between the sessions in the chat tab as well.

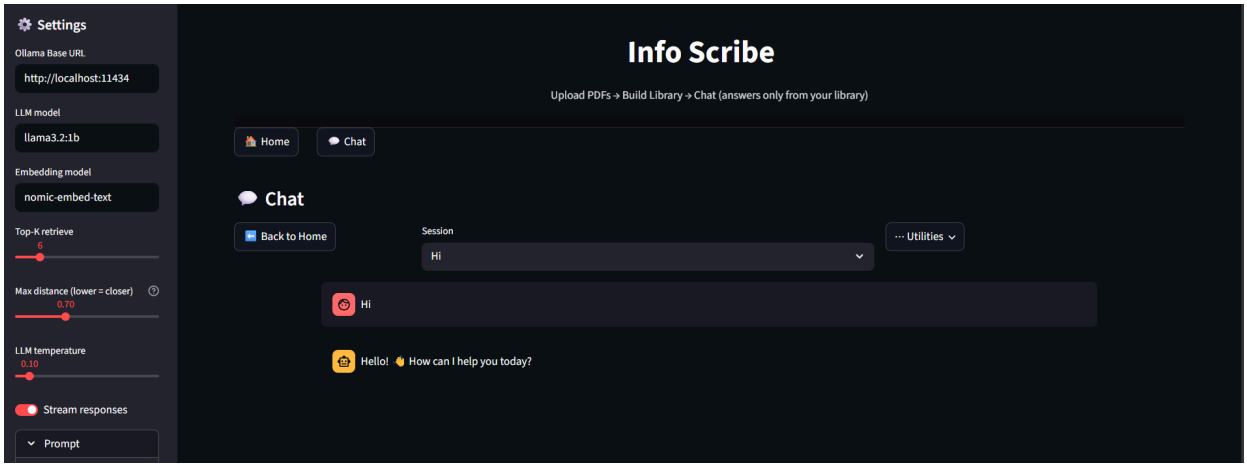


Here is the chat tab. I created a session named Quantum Chapter-1 and started asking it questions based on the PDF I uploaded for that topic. Because the chatbot is strictly linked to the content of the uploaded document, every answer it gives in this session comes only from the PDF context.

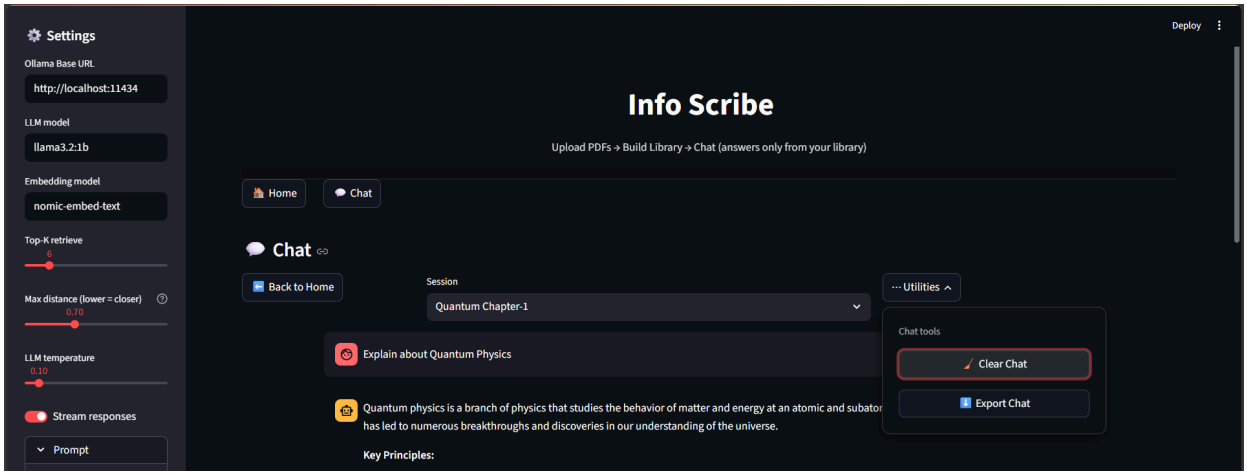
.



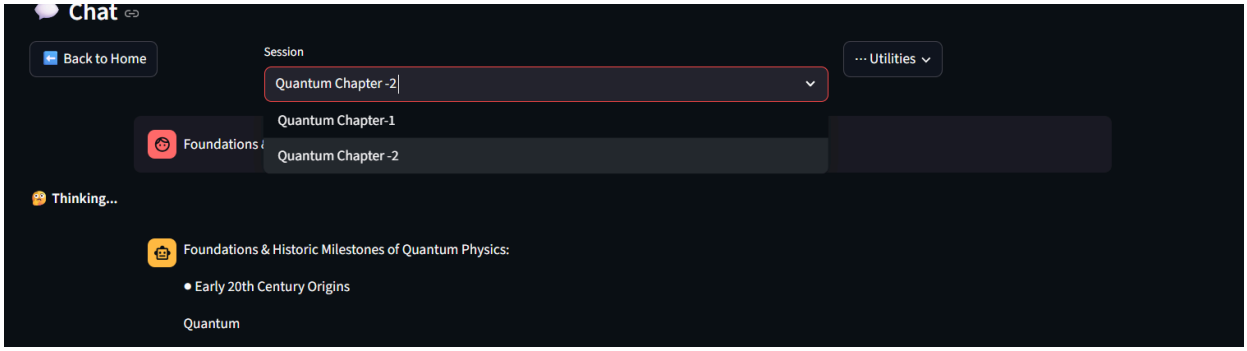
If I ask something unrelated to the uploaded PDF — for example, a completely different topic that is not covered in the document the chatbot will not answer it. Instead, it follows the strict rule I have set in the system prompt and replies with **“I don’t know from the provided documents.”**



However, if I send a greeting like “Hi” or “Hello,” the chatbot will still respond politely and greet me back, even if it’s not in the PDF. This keeps the interaction friendly.



In the **Utilities** dropdown inside the Chat tab, I have added two useful options for managing a session's conversation without deleting the entire session. First, I can **save the chat as a `chat.md` file**, which downloads the full conversation in a readable Markdown format for record-keeping or sharing. Second, I can **clear the chat** of the current session if I want to start fresh while still keeping the session itself. This way, I can reset the conversation without losing the session name or its place in my session list.



As I mentioned earlier, one can switch between the sessions in the chat tab also .