1. Goal

Build a simple retrieval-augmented-generation (RAG) chatbot exposed through one REST endpoint.

2. Core Behaviour

Query domain	Vector store	Custom prompt
Healthcare	Healthcare document embeddings	Healthcare-specific prompt
Fashion	Fashion document embeddings	Fashion-specific prompt

- No conversation history required—each request is stateless.
- The chatbot chooses the correct store & prompt solely from the incoming query.

3. Minimal Requirements

1. **API**
    a. `POST /chat` → body: `{ "query": "<user text>" }`
    b. returns: `{ "answer": "<chatbot reply>", “source”: <which source is used to generate the answer if any> }`

2. **Agent Logic**
    a. Detect domain *(healthcare vs. fashion)*.
    b. Retrieve top-k relevant passages from the appropriate vector DB.
    c. Generate an answer using the paired prompt.

3. **Data**
    a. Provide a small set of documents for each domain (blog posts, PDFs, or AI-generated text are fine).
    b. You may keep the vector stores in-memory.

4. Recommended Stack (feel free to swap)

- **FastAPI** – REST service
- **LangGraph / LangChain** – agent & tool calling
- **Any LangChain-compatible vector DB** – e.g. FAISS (in-memory)
- **OpenAI / Llama-compatible model** – for embeddings & generation
- AI-based IDE like Cursor, it’s free for students :)

5. What to Submit

Source code (Git repo or zip) with a README :))