



## 🔍 Ask the Camp: RAG Demo (Streamlit + FAISS + Open LLM)

Welcome to the **CodeBreakers RAG Demo**, a student-friendly starter project that teaches how to build an AI assistant using Retrieval-Augmented Generation (RAG). This project reflects the values in the [CodeBreakers Manifesto](#)—open, ethical, user-first technology.

### What Is This?

This is a **question-answering assistant** built with: - **Streamlit** – Easy UI for chatting with the assistant - 🔍 **FAISS** – Fast similarity search engine to retrieve relevant text chunks - **Hugging Face LLM** – Open-source language model for generating answers

### Folder Structure

```
rag-demo/  
├── app.py           # Main Streamlit app  
├── docs/           # Your data source (camp handouts, README files, etc.)  
├── faiss.index      # Auto-generated FAISS index file  
├── embeddings.json  # Embedding metadata for each chunk  
├── requirements.txt # Python dependencies  
└── README.md       # You're reading it!
```

### How to Run (Local)

#### 1. Clone the repo

```
git clone https://github.com/YOUR-ORG/rag-demo.git  
cd rag-demo
```

#### 2. Install dependencies (recommended: use a virtualenv)

```
pip install -r requirements.txt
```

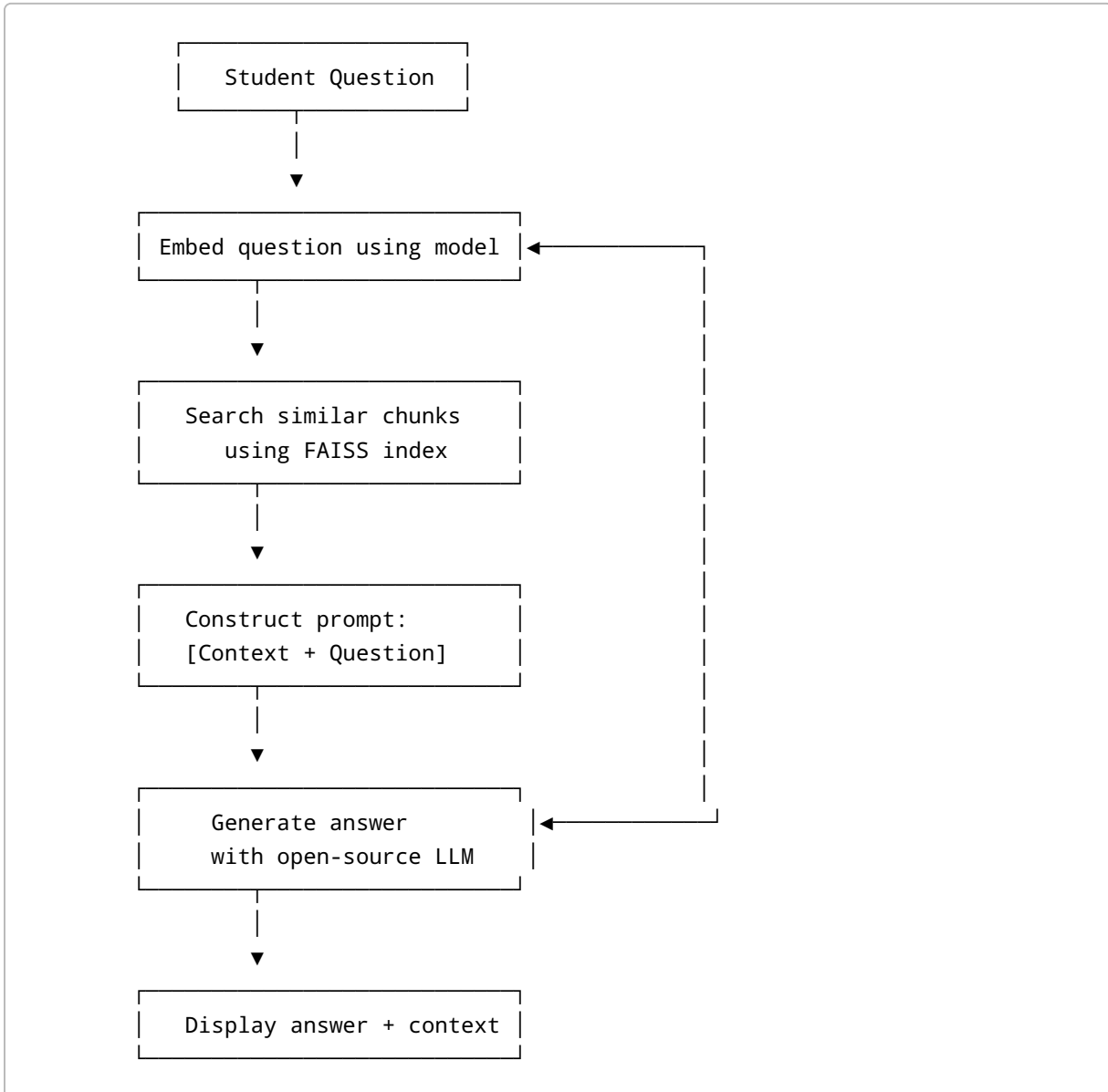
#### 3. Add documents Drop `.txt` or `.md` files into the `docs/` folder (e.g., camp schedule, tutorials).

#### 4. Launch the app

```
streamlit run app.py
```

5. Ask it anything! Example: *"What are students doing on Day 2?"*

### How It Works (Architecture)



### Student Challenges

- Add a new document (e.g., team instructions)

- Improve chunking logic (e.g., by section headings)
- 🔍 Let users see how the retrieved chunks impact the answer
- Replace model with local LLM (e.g., using [Ollama](#))
- Add a feedback button ("Was this answer helpful?")

## License

MIT License. You are free to remix, reuse, and deploy. Teach others!

---

**Built with ♥ by CodeBreakers / Next Shift Consulting** Join the movement → [CodeBreakers Manifesto](#)