# **📘 Chat with Your Text**

**This project is a Streamlit application that allows users to paste any text, ask a question about it, and get an AI-generated answer. It combines several important tools:**

* **Streamlit – for the web interface**
* **Sentence Transformers – to create embeddings of text**
* **FAISS – to search and retrieve relevant parts of the text**
* **Flan-T5 – a transformer model for generating answers**

## **1️⃣ Libraries Used**

* **streamlit: Creates an interactive web app**
* **transformers: Provides pretrained AI models (Flan-T5 in this case)**
* **sentence\_transformers: Converts text into vector embeddings**
* **faiss: Fast similarity search to retrieve relevant text parts**
* **numpy: Numerical operations**
* **textwrap: Formats output neatly**

## **2️⃣ Model Loading & Caching**

* **SentenceTransformer("all-MiniLM-L6-v2") → Creates embeddings (vector representation) of text**
* **pipeline("text2text-generation", model="flan-t5-base") → Loads Flan-T5 for question answering**
* **@st.cache\_resource → Caches models so they load only once (saves time & memory)**

## **3️⃣ Chunking Text**

* **Large text is broken into smaller overlapping chunks (200 words, 50% overlap)**
* **Example: A 1000-word article becomes ~5 chunks**
* **Why overlapping? → Ensures context continuity between chunks, so answers are more accurate**

## **4️⃣ Embeddings & FAISS Index**

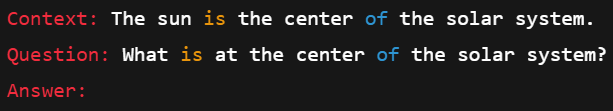
* **Each chunk is converted into normalized embeddings**
* **FAISS IndexFlatIP is used to store embeddings and perform cosine similarity search**
* **Why Inner Product with normalized embeddings? → More accurate semantic search than Euclidean distance and faster**

## **5️⃣ Question Handling**

* **Question is converted into normalized embeddings**
* **FAISS searches the index to find the top\_k=2 most relevant chunks**
* **Relevance threshold is applied to filter unrelated questions**

## **6️⃣ Answer Generation**

* **Merges relevant chunks into a single context**
* **Prepares prompt: Context + Question**
* **Flan-T5 generates the final answer based on context**

****

## **7️⃣ Streamlit UI**

* **App title and description**
* **Text area to paste input text**
* **Text box to type a question**
* **Button to trigger answer generation**

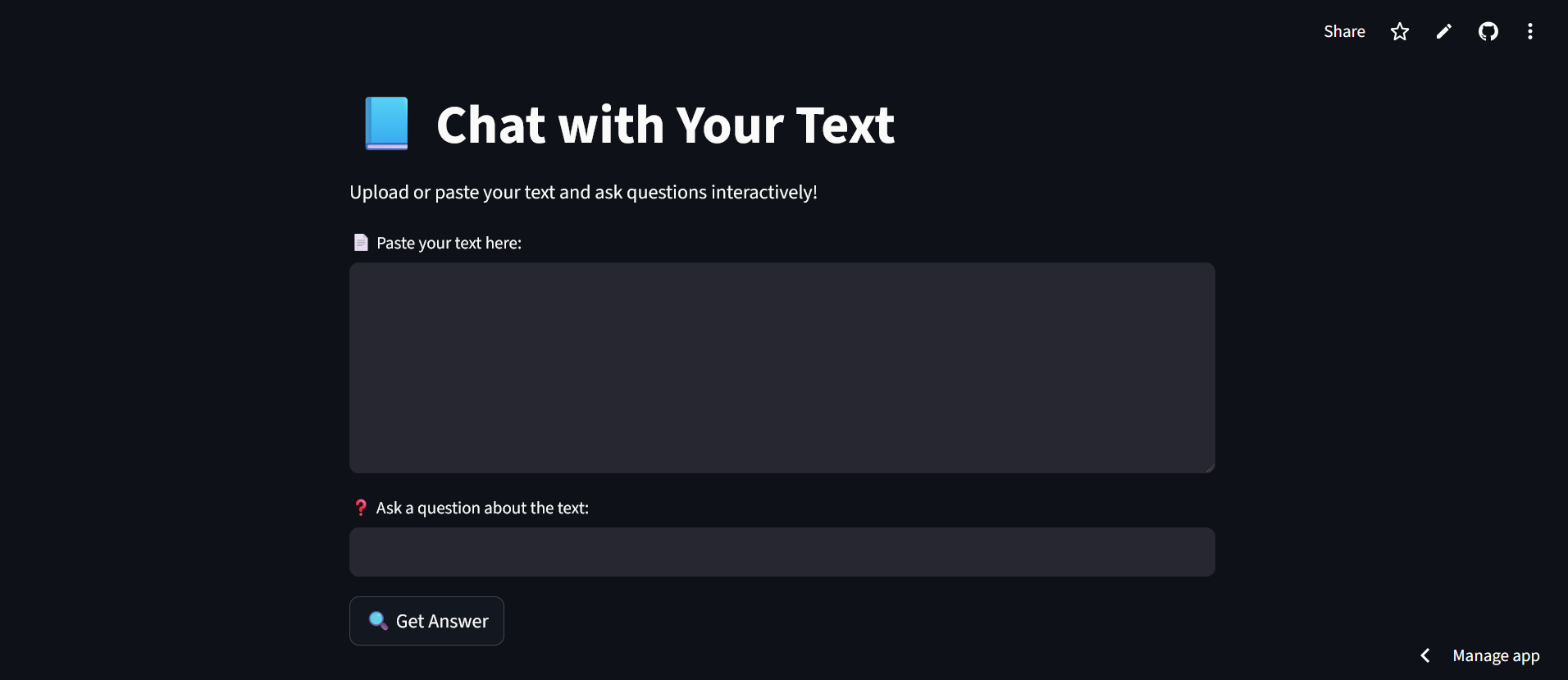
## **8️⃣ Workflow When “Get Answer” is Clicked**

1. **Check if text/question is empty → show warning**
2. **Split text into overlapping chunks**
3. **Build FAISS index with normalized embeddings**
4. **Retrieve the most relevant chunks**
5. **Merge them into context**
6. **Generate answer using Flan-T5**
7. **Display the result in a nicely formatted text box**

## **🔑 Key Takeaways**

* **Chunking with overlap → Handles large text efficiently**
* **Normalized embeddings + Inner Product → Accurate semantic search**
* **FAISS → Fast retrieval of relevant text**
* **Flan-T5 → Generates answers using context**
* **Streamlit → User-friendly, interactive interface**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

****