


Technical Report – RAG Chatbot with Streaming Responses

 **Candidate Name: Shivam Bhatia**

 **Company: Amlgo Labs**

 **Role: Junior AI Engineer (Assignment)**

 **Date: 11th July 2025**

Shivambhatia800@gmail.com

1. Project Overview

The goal of this project was to build a Retrieval-Augmented Generation (RAG) based chatbot that could answer user questions using content from a provided legal document (Terms & Conditions). The chatbot is designed to provide **factual, grounded answers** and include **streamed output responses** through a web-based interface using **Streamlit**.

The core idea was to demonstrate:

- Document preprocessing and chunking
- Semantic retrieval using a vector database (FAISS)
- Context-aware generation using a fine-tuned open-source LLM
- Real-time, interactive chatbot UI with source citations

This system is **completely local** and **does not rely on OpenAI or cloud services**, making it suitable for private legal, enterprise, and offline use cases.

2. Document Structure & Chunking Logic

The source document was a **legal PDF (~10,000+ words, 20 pages)** containing:

- Sectioned clauses (e.g., "User Obligations", "Termination", "Privacy", etc.)
- Long sentences with legal terms
- No consistent headings or formatting structure

Preprocessing Flow:

Step	Action
1	Loaded PDF using PyMuPDF (fitz)
2	Removed excess whitespace and newlines using regex
3	Used spaCy tokenizer (en_core_web_sm) to split into sentences
4	Grouped sentences into chunks of 150–300 words
5	Saved each chunk as chunk_0.txt, chunk_1.txt, ... in /chunks/

This ensured:

- Chunks respected sentence boundaries (for semantic integrity)
- Each chunk was compact enough for token limits (LLMs)
- The retrieval system could map answers back to chunks accurately

 **Total Chunks Generated: 64**

3. 🔍 **Embedding Model & Vector Database**

☒ **Embedding Model:**

- all-MiniLM-L6-v2 from sentence-transformers
- Lightweight (384-dim), fast, and great for short text embeddings
- Captures sentence-level semantic meaning

☒ **Vector Database:**

- **FAISS (Facebook AI Similarity Search)** – FlatL2 index
- Added all embeddings as dense vectors
- Used `.search()` for nearest neighbor matching (Top-k)

All embeddings were stored in:

- `vectordb/index.faiss` (vector index)
- `vectordb/chunk_text.pkl` (original text metadata)

This enabled the retriever to:

- Take user queries
- Convert them to embeddings
- Return the top-3 most relevant chunks (based on cosine similarity)

4. Generation & Prompt Logic

The answer generator used a **locally hosted LLM**:

☒ **mistral-7b-instruct-v0.1.Q4_K_M** via **LM Studio** (GGUF format)

Why Mistral-7B?

- Instruction-tuned (v0.1)
- Small enough to run on local hardware (GTX 1650)
- Good factual accuracy and adherence to prompt
- Works with LM Studio's OpenAI-compatible API

Prompt Template Used:

System Prompt:

"You are a helpful assistant. Use only the provided context to answer the user's question.

If the answer is not found, say 'I don't know based on the provided content.'"

User Prompt:

Context:

[1] <chunk 1>

[2] <chunk 2>

[3] <chunk 3>

Question: <user_query>

Answer:

Streaming:

- Set stream=True in the API call
- Used requests.stream + st.write_stream() in Streamlit
- Response appears **word-by-word**, just like ChatGPT

This greatly improves UX and makes the bot feel faster.

5. Sample Queries (Success & Failure Cases)

☒ Successful & Accurate Responses

# Question	Bot Answer	Source Chunks
1 What is the main purpose of the User Agreement?	Bot: The main purpose of the User Agreement is to set out the terms on which eBay offers you access to and use of their Services. It includes an overview of their policies, the Mobile Application Terms of Use, and additional terms posted on and in their sites, applications, tools, and services. The User Agreement outlines the terms that users must comply with when accessing or using eBay's Services.	0, 58, 15
2 What happens if a user violates eBay's policies?	Bot: If a user violates eBay's policies, they may face various consequences such as having their account suspended or terminated, being charged additional fees, and/or having legal action taken against them. It is important for users to carefully read and understand eBay's policies before using the platform to avoid any potential issues.	34, 56, 57
3 What happens if a seller tries to sell outside of eBay?	Bot: If we believe you are violating our policies prohibiting Offers to buy or sell outside of eBay, you may be subject to a range of actions, including limits on your buying and selling privileges, restrictions on listings and account features, suspension of your account, application of fees, and recovery of expenses for policy monitoring and enforcement. Additionally, as provided below in the Fees and Taxes section, you may be charged final value fees.	34, 61, 56

Properly Failed or Controlled Cases

Query	Response	Result
What is the Wi-Fi password?	"I don't know based on the provided content."	<input checked="" type="checkbox"/>
How do I apply for a refund?	"This agreement does not mention any refund policy."	<input checked="" type="checkbox"/>
Who signed this document?	"The context does not include signer names."	<input checked="" type="checkbox"/>

→ Note aprox all the Responses are accurate and Successful

6. ⚠ Limitations, Challenges, & Improvements

⚠ Observed Limitations

- **No reranking or MMR:** Top-3 FAISS chunks may overlap or contain redundant info
 - **Token limit:** Only 3–4 chunks passed to LLM per query to avoid prompt overflow
 - **Model cold-start delay:** 8–10 seconds if LLM not warmed up
 - **No PDF upload UI yet:** PDF path is fixed to /data/document.pdf
 -
-

☑ What Works Well

- Real-time responses even on GTX 1650 (Q4_K_M)
 - Minimal hallucination (due to strict prompt + grounding)
 - Reliable streaming via LM Studio
 - Fast retrieval from FAISS (<100ms)
 - Clean, usable UI with citations
 -
-

🔧 Future Enhancements

Feature	Description
☑ PDF Upload	Let user upload new documents from UI
☑ Reranking	Use LLM or scoring to refine chunk quality
☑ Chunk Highlighting	Highlight sources inside answers
☑ Export/Save Chat	Save history to PDF or markdown
☑ Deploy on Cloud	Shareable via Streamlit Cloud or Hugging Face Spaces

7. 📦 Tools & Libraries Used

Component	Tool/Library
PDF Parsing	PyMuPDF (fitz)
Sentence Splitting	spaCy (en_core_web_sm)
Embeddings	sentence-transformers
Vector DB	FAISS
LLM Inference	Mistral-7B-Instruct (GGUF)
LLM Hosting	LM Studio
UI Interface	Streamlit
HTTP API Client	requests

☑ Conclusion

This project showcases a complete **RAG pipeline** using:

- Smart document preprocessing
- Lightweight embeddings
- Efficient semantic retrieval
- Local LLM-based response generation
- A fast, interactive streaming chatbot UI

All components were built to be **modular, local, and privacy-respecting**, making the solution ideal for legal, policy, or compliance-oriented Q&A systems.