RAG Chatbot Project Report

1. Document Structure & Chunking

The chatbot uses a **Retrieval-Augmented Generation (RAG)** approach to answer user queries from a knowledge base (e.g., eBay User Agreement).

- **Document ingestion**: The source document (PDF, DOCX, or text) is read into memory.
- **Text preprocessing**: Headers, paragraphs, and bullet points are preserved while unnecessary formatting (extra spaces, page numbers) is removed.
- **Chunking**: The document is split into smaller chunks (200–500 tokens each) using a **sliding window** technique. This ensures that important contextual overlaps are not lost between chunks.
- **Metadata tagging**: Each chunk stores metadata such as *page number*, *section title*, and *timestamp of ingestion*. This allows better retrieval and filtering.

2. Embedding Model & Vector Database

- **Embedding Model**: The system uses sentence-transformers/all-MiniLM-L6-v2 (or similar) to convert text chunks into high-dimensional vectors. This model balances speed and accuracy for semantic similarity searches.
- **Vector Database**: **FAISS** is used for efficient similarity search. It supports cosine similarity and inner product distance metrics, enabling quick retrieval even with thousands of chunks.
- **Index Type**: FAISS IndexFlatIP is chosen for small to medium datasets. For larger datasets, HNSW or IVF indexing can be used.

3. Prompt Format & Generation Logic

Once relevant chunks are retrieved, they are injected into a **structured prompt** for the LLM.

Prompt Template Example:

You are an assistant that answers based on the provided context.

Only answer from the context and do not fabricate information.

Context:

{retrieved_chunks}

Question:

{user_query}

Answer:

- The retrieved chunks are concatenated and inserted into {retrieved_chunks}.
- The LLM (e.g., GPT-4, LLaMA 3, or Mistral) then generates an answer strictly using the provided context.
- If no relevant chunk is found, the chatbot responds with:
  "I could not find relevant information in the document."

4. Example Queries

✅ Successful Queries (Good Retrieval)

Query: What are the main terms of the eBay User Agreement?
Answer: The agreement outlines terms for accessing eBay's services, including compliance with policies, contracting entity information, and applicable laws.

Query: Explain the payment services section.
Answer: Sellers must follow the Payments Terms of Use, provide necessary business details, and payments are processed via eBay Payment Entities.

❌ Failure / Hallucination Cases

Query: Who created this chatbot and in which year?
Answer: The document does not mention the creator or year.

Query: List all products sold on eBay in 2025.
Answer: Not available in the document.

5. Hallucinations, Limitations & Speed Issues

Hallucinations:

- If no relevant context is retrieved, the LLM might try to guess the answer instead of returning "not found."
- Mitigation: Add strict instructions in the system prompt to avoid guessing.

Limitations:

- Answers depend entirely on the uploaded document.
- Cannot handle ambiguous queries well without multiple clarifying steps.
- Large documents require higher compute and memory.

Speed Issues:

- Slow response for large PDFs due to chunking + embedding computation.
- Mitigation: Precompute embeddings and cache results.
- Retrieval speed can be improved by switching from FAISS flat index to HNSW for large datasets.

Conclusion

This RAG chatbot effectively retrieves and answers queries based on uploaded documents, ensuring accuracy when relevant chunks are available. With optimizations for speed, better prompt engineering, and improved hallucination handling, it can be deployed as a robust document Q&A system for legal, academic, and business use cases.