**Report on "Local Retrieval-Augmented Generation (RAG) with PDFs"**

# Problem Statement

In many domains, the efficient processing and retrieval of information from large/complex documents such as PDFs is crucial. Manual searching through information is usually very time consuming and quite often fails to catch the complex semantics deeply embedded into lengthy documents. This project aims to solve a challenging task of ingesting, processing and retrieving meaningful information from PDF documents using state-of-the-art machine learning methods.

# Unique Idea Brief (Solution)

The proposed solution is to apply advanced NLP techniques using the existing state of art methods on PDF documents so that they can be easily searched and processed. The system uses vector embeddings and a retrieval-augmented generation (RAG) method in order to improve the search for relevant information based on large text corpuses. The solution incorporates the following unique elements:

· **Ingesting PDFs:** Using specialized loaders to convert PDF content into text.

· **Vector Embeddings:** Transforming text into vector embeddings for efficient retrieval.

· **Retrieval-Augmented Generation:** Employing a language model to generate contextually relevant responses based on retrieved information.

# Features Offered

· **PDF Ingestion:** Ability to load and process PDF documents, converting them into structured text data.

· **Text Splitting and Chunking:** Breaking down large texts into manageable chunks for better processing and embedding.

· **Vector Embeddings:** Using models like nomic-embed-text to create vector representations of text chunks.

· **Vector Database:** Storing vector embeddings in a vector database (Chroma) for efficient retrieval.

· **Multi-Query Retrieval:** Enhancing retrieval by generating multiple perspectives on user queries to fetch the most relevant documents.

· **Language Model Integration:** Using models like ChatOllama for generating answers based on retrieved information.

# Process Flow

· **PDF Ingestion:** The PDF document is loaded using UnstructuredPDFLoader.

· **Text Processing:** The document is split into chunks using RecursiveCharacterTextSplitter.

· **Vector Embeddings:** Each chunk is embedded into a vector space using OllamaEmbeddings.

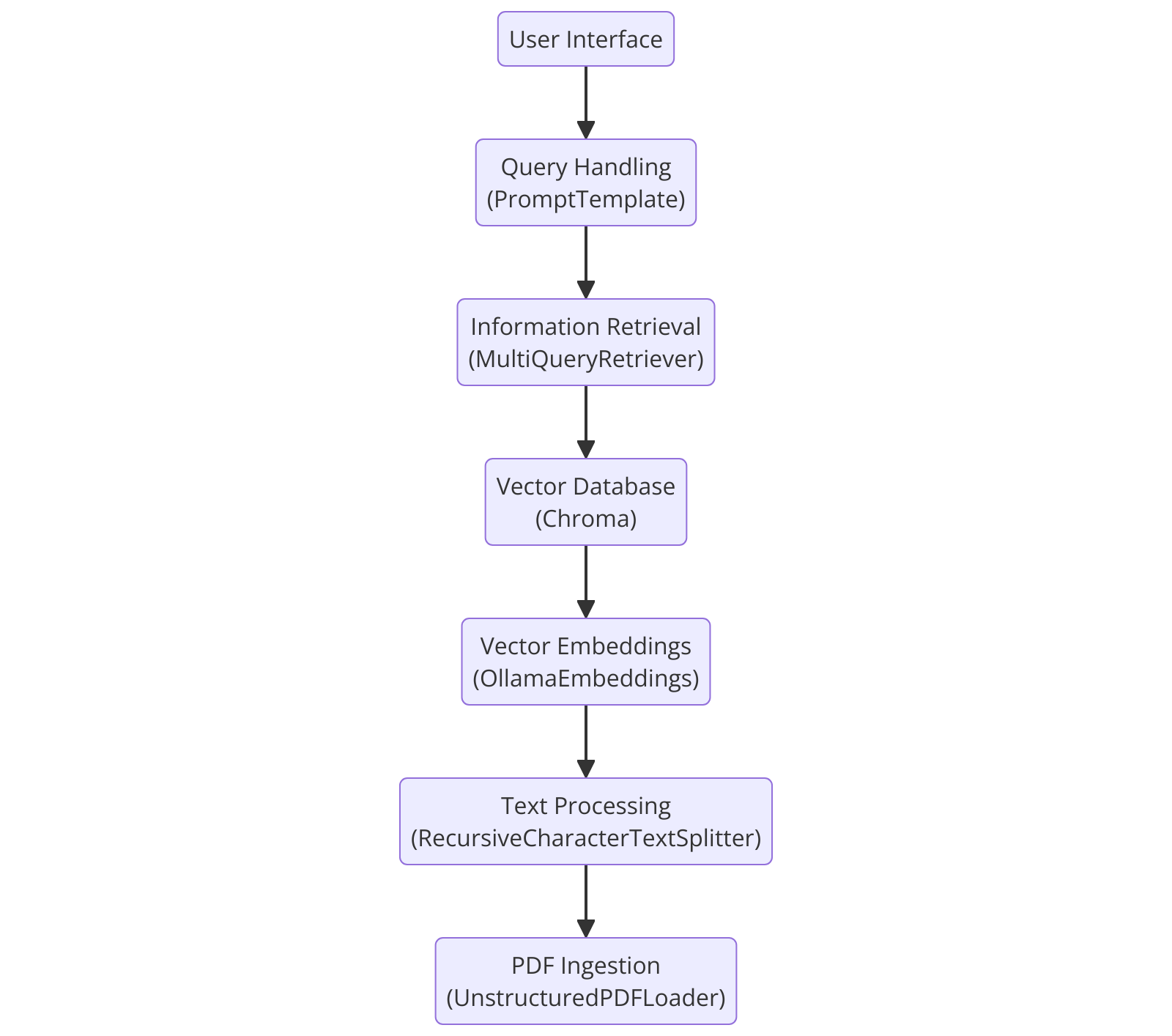
· **Database Storage:** The vectors are stored in a Chroma vector database.

· **Query Handling:** User queries are processed, and multiple versions of the query are generated using PromptTemplate.

· **Information Retrieval:** Relevant document chunks are retrieved from the vector database.

· **Response Generation:** The retrieved information is used by the ChatOllama model to generate a response.

# Architectural Diagram



# Technologies Used

· **Langchain:** For document processing and text splitting.

· **Ollama:** For generating text embeddings and handling language models.

· **Chroma:** Vector database for storing and retrieving vector embeddings.

· **Langchain-Community:** For integrating various components like PDF loaders and text splitters.

· **Python Libraries:** Including unstructured, langchain, and chromadb for implementing the solution.

# Team Members and Contribution

The diligent work of **Diptesh Bal** and **Harsh pd Singh**, two members of our team, enabled the Local Retrieval-Augmented Generation (RAG) system with PDFs to be developed and implemented successfully.

**Harsh pd Singh** oversaw and provided strategic direction to make sure the project stayed on course and achieved all of its goals. He played a key role in the architecture's design, the integration of different parts, and the resolution of challenging problems that came up during development

.

An important part of the system's practical implementation was handled by **Diptesh Bal (team lead).** Coding the ingestion and processing modules, establishing the vector database, and incorporating the language model for response generation were all under his purview. His meticulous attention to detail and innovative problem-solving skills were key to optimizing the system's performance.

Together, their collaborative efforts resulted in a robust and efficient solution that significantly enhances information retrieval from large PDF documents.

# Conclusion

The problem of processing and obtaining data from big PDF documents is effectively addressed by the suggested method. Through the use of vector embeddings, sophisticated natural language processing (NLP) techniques, and a resilient retrieval-augmented generation framework, the system greatly improves the capacity to extract and apply information from complicated documents. This solution can be especially helpful in fields like business, law, and research where it's critical to retrieve information quickly and accurately.