#### ****Retrieval-Augmented Generation (RAG)**** System

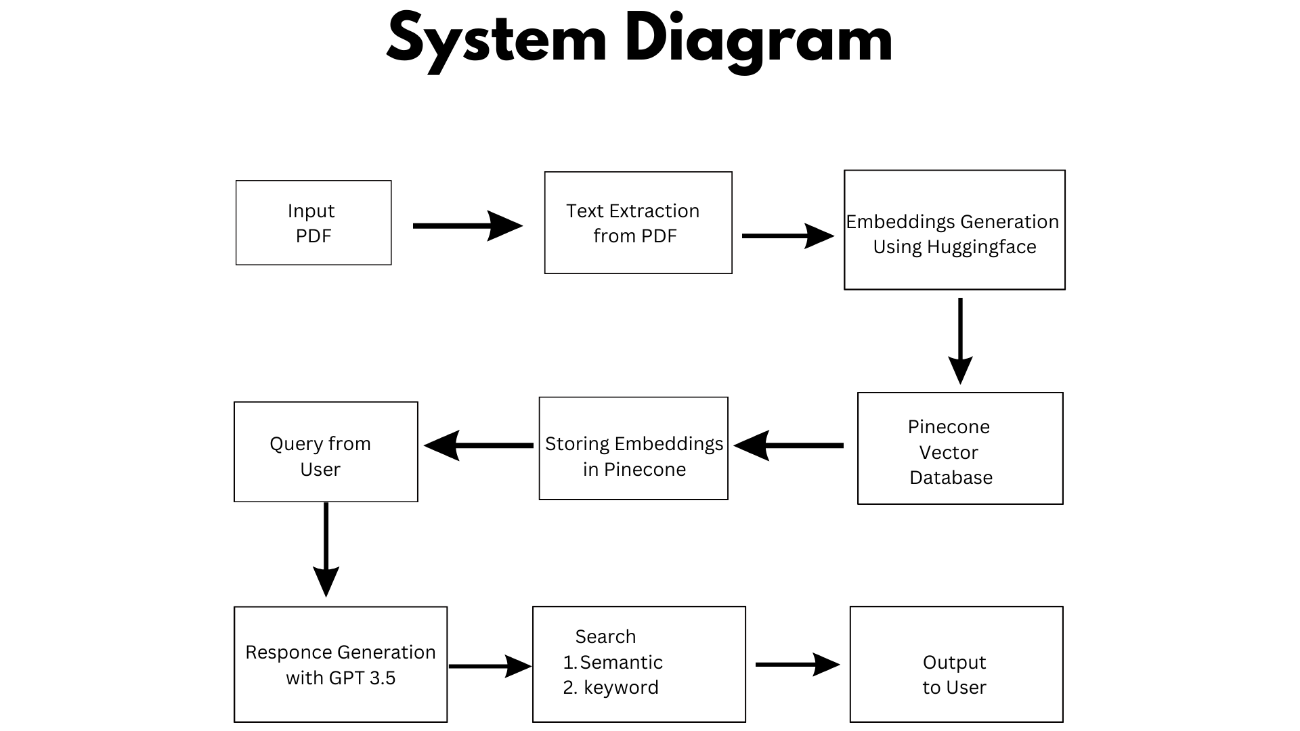
#### ****GitHub Repository:**** [*https://github.com/Mayurgohane/CerebralZip-Data-Scientist-Assignment*](https://github.com/Mayurgohane/CerebralZip-Data-Scientist-Assignment)

#### ****System Overview****

The system is a **Retrieval-Augmented Generation (RAG)** pipeline developed to process digital PDFs, extract relevant information, and provide answers and summaries based on the content. It is designed for English-language documents and ensures scalability for large datasets while delivering efficient and accurate query responses.

### **System Architecture**

1. **Input**: Digital PDFs in English.
2. **Preprocessing**: Extract text from digital PDFs.
3. **Embedding**: Generate vector representations of text data.
4. **Indexing**: Store embedding in a scalable vector database.
5. **Query Processing**:
   * Decompose user queries into sub-queries.
   * Retrieve relevant chunks
6. **Answer Generation**: Utilize LLM to generate accurate responses.
7. **Chat Memory**: Maintain session context for continuous interactions.

**System Diagram**

**Performance Evaluation**

* Query Relevance: Evaluated by accuracy of returned chunks.
* Latency: Average response time per query.
* Scalability: Successfully tested with embedding in Pinecone.

**Challenges faced**

* Managing token limitations for longer PDFs.
* Optimizing costs for API calls with large datasets.

**Future improvements**

* Integrate OCR capabilities for processing scanned PDFs.
* Enhance chunking strategies for longer-context models (e.g., GPT-4).
* Introduce parallel query processing for reduced latency.
* Explore smaller, domain-specific LLMs for cost-efficiency.