**Assignment – 3: Retrieval-Augmented Generation (RAG) using LangChain**

**Part-I: Conceptual Understanding of RAG Objective:**

To understand the concept and need for Retrieval-Augmented Generation, its architecture, components, and relevance in modern LLM applications.

**Task 1: Short Answer Questions:**

1. **What is the motivation behind Retrieval-Augmented Generation (RAG)?**

**Ans:**

* Overcoming LLM Memory Limits
* Improving Accuracy and Relevance
* Dynamic Knowledge Integration
* Reducing Hallucinations

1. **Compare “stuff”, “map\_reduce”, and “refine” document chain types in LangChain.**

Ans:

| Stuff | Map Reduce | Refine |
| --- | --- | --- |
| Combines all retrieved documents into a single prompt and sends it to the LLM. | Maps each document to a response using the LLM, then reduces those responses into a final answer. | Starts with an initial document and response, then iteratively refines the answer using additional documents. |
| Best for small document sets or short texts. | deal for large document sets needing summarization. | Useful when incremental improvement is needed. |
|  |  |  |

1. **What is the role of a vector store in a RAG pipeline?**

Ans:The **vector store** plays a central role in the RAG pipeline by acting as the memory bank of embedded knowledge.

* Stores Embeddings:
* Feeds Context to the LLM:

4**Explain the difference between RAG and standard LLM-based QA.**

Ans:

| Feature | Standard LLM-Based QA | RAG-Based QA |
| --- | --- | --- |
| Knowledge Source | Relies solely on pre-trained model knowledge | Retrieves external documents at query time |
| Accuracy | Prone to hallucinations or outdated facts | Anchored to real, retrieved data |

**5)What are the main components of a basic LangChain RAG pipeline?**

Ans:

* Document Loader & Chunker
* Embedding Model
* Vector Store
* Retriever

**Task 2: RAG Pipeline Diagram**

Draw or describe the flow of a RAG system showing:

● User Query ● Retriever ● Vector Store ● LLM ● Final Answer Generation

