### **Assignment: AI-powered Information Processing Application**

In this assignment, you will act as a software engineer designing, implementing, and assessing an AI-powered system that helps users process and acquire knowledge efficiently.

You are expected to **independently** complete the assignment within **10-15 hours**, using external tools, frameworks, and APIs as needed. If the complete system is too large, you may focus on a meaningful subset or key capability.

#### **Choose one of the following assignments:**

### **1. AI-Powered Question-Answering System using RAG (Retrieval-Augmented Generation) and LLM**

**Design and develop a system that allows users to ask questions and receive accurate answers from a predefined knowledge base, using Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs).**

**Key Features:**

* Allow users to upload documents (e.g., PDFs, text files) or use a predefined dataset.
* Process and index the documents using **vector embeddings** (e.g., FAISS, ChromaDB).
* Retrieve relevant context from the indexed data and generate answers using an **LLM** (e.g., GPT-4, Mistral, LLaMA).
* Provide an interactive web-based UI (e.g., **Streamlit** or **Gradio**) or an API (e.g., **FastAPI**).
* (Optional) Fine-tune search and response accuracy based on user feedback.

**Possible Integrations:**

* Use **OpenAI API**, **Hugging Face models**, or **LangChain** for LLM-based processing.
* Integrate with **cloud storage** (e.g., AWS S3, Firebase) for document handling.

### **2. AI-Powered Legal Document Assistant using RAG and NLP**

**Design and develop an AI-powered system that helps users quickly extract insights, summarize, and answer legal-related queries from contracts, policies, and other legal documents using Retrieval-Augmented Generation (RAG) and Natural Language Processing (NLP).**

**Key Features:**

* Allow users to upload legal documents (e.g., contracts, policies, terms of service) in PDF or text format.
* Extract and process key clauses using **Named Entity Recognition (NER)** and **chunking** techniques.
* Implement **semantic search** to retrieve relevant sections from the documents.
* Use **RAG + LLM** to provide concise answers based on user queries.
* Provide an interactive web-based UI (e.g., **Streamlit, Gradio**) or an API (e.g., **FastAPI**).
* (Optional) Summarize complex legal texts into **plain language** for easier understanding.

**Possible Integrations:**

* Use **Hugging Face Transformers**, **OpenAI GPT-4**, or **Google NLP API** for document processing.
* Store processed data in **vector databases** (e.g., Pinecone, Weaviate, FAISS) for efficient retrieval.

### **Expectations & Deliverables:**

* **Codebase with clear documentation** on the chosen frameworks, libraries, and APIs.
* **A working prototype** that demonstrates core functionality.
* **Brief architecture explanation** (e.g., README or short documentation).
* **Usage instructions** for running and testing the application.

You are free to explore and learn new technologies while working on this task. The assignment is designed to assess your ability to build a functional AI-powered system efficiently within a limited timeframe.