**AI-Agent-PDF**

Purpose: using Ollama LLM model to run RAG (retrieval augmented generation) locally, add information to database, for question and answer based on PDF books

**RAG** stands for **Retrieval-Augmented Generation**.

It is an architecture used in natural language processing (especially with large language models like GPT, LLaMA, etc.) that **combines a retriever with a generator** to answer questions or generate text **based on external knowledge**.

**Frameworks**: Ollama, Langchain, ChromaDB

**Workflow**:

1. Load document (PDF) using PyPDFDirectoryLoader 🡪 Text splitter (breaking up text from the document in chunks) 🡪 Embeddings to store inside vector database.
2. Query question based on the prompt 🡪 Invoke Llama model to assess the context and question of the prompt 🡪 investigate vector database for appropriate info 🡪 output response.

**Steps setting**:

1. Install the dependencies:

|  |  |
| --- | --- |
| Library | Purpose |
| Pip install langchain  Pip install chromadb  Pip install pypdf  Pip install pytest | LLM Library  Vector storage  Load PDFs  Unit testing |

1. Create requirements.txt file
2. Load document (PDFs) using PyPDFDirectoryLoader
3. Break down text in chunks (easy for embedding)
4. Use Ollama Embedding model (nomic-embed-text) locally.
5. Create the database using ChromaDB by building vector database using the embedding function