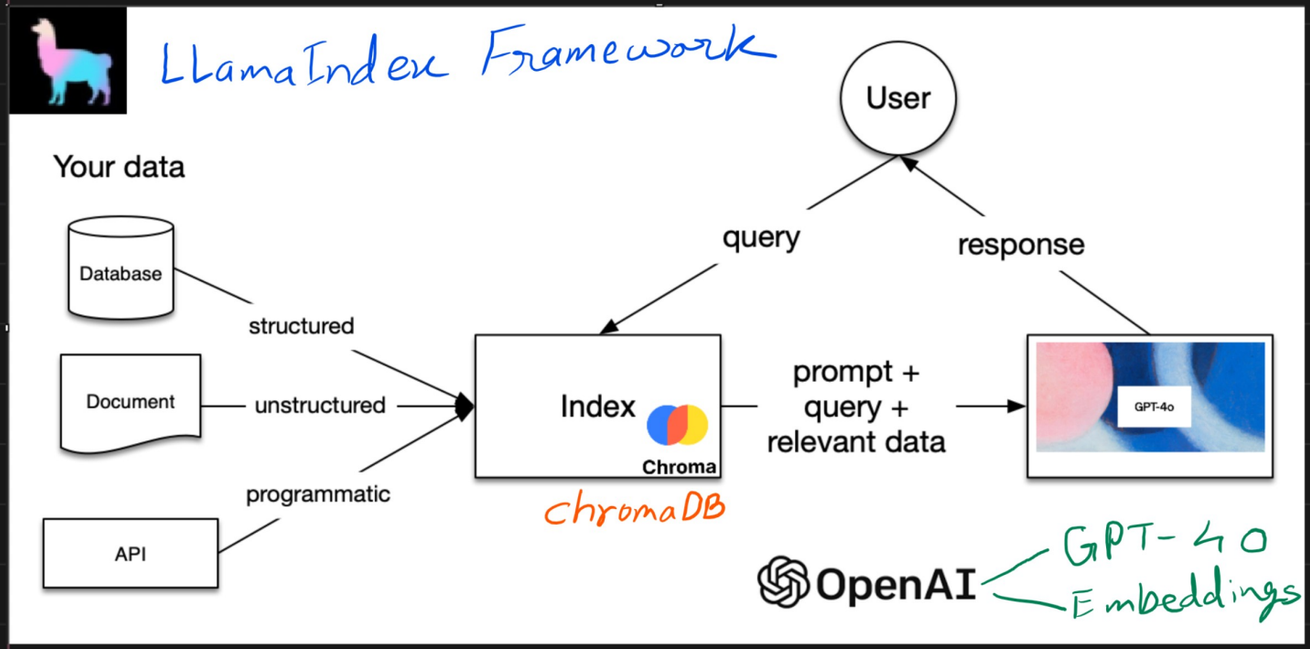
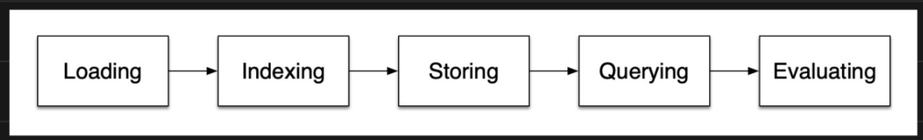
AI Nutritionist - Demo



Stages Of RAG:-



RAG- Retrival Augmented Generation

How does RAG work?

1. **Retrieval:** When a user asks a question or provides a prompt, the RAG system first retrieves relevant information from external knowledge sources. These sources can include documents, databases, or any structured or unstructured data.

2. **Augmentation:** The retrieved information is then used to augment the LLM's knowledge base. This gives the LLM the context it needs to understand the query more thoroughly.

3. **Generation:** Finally, the augmented LLM generates a response that is based on both its internal knowledge and the retrieved information. This results in a more informed and accurate answer than the LLM could generate on its own.

LOADING STAGE

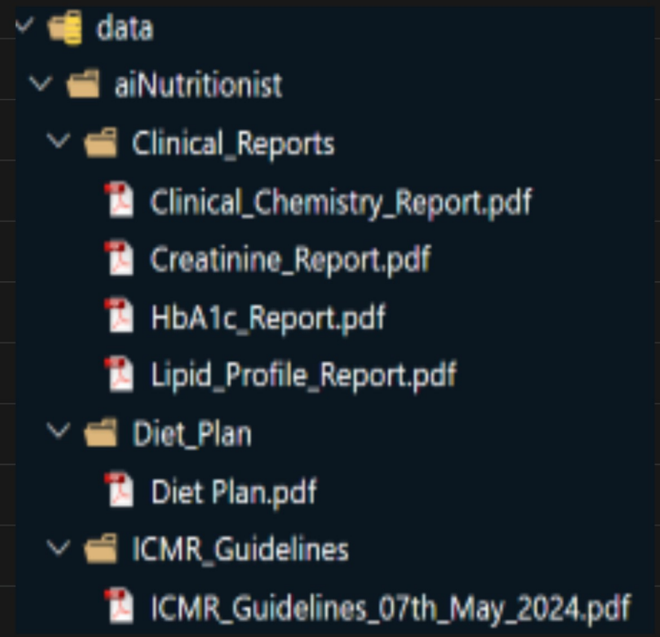
Loading: this refers to getting your data from where it lives -- whether it's text files, PDFs, another website, a database, or an API -- into your pipeline. LlamaHub provides hundreds of connectors to choose from.

1) Clinical Reports Of the user:

2) ICMR Guidelines (for Indians)

3) Existing Diet Plan followed by user

- We will be using SimpleDirectoryLoader to load these pdf documents.



INDEXING STAGE

Indexes: Once you've ingested your data, LlamaIndex will help you index the data into a structure that's easy to retrieve. This usually involves generating vector embeddings which are stored in a specialized database called a vector store. Indexes can also store a variety of metadata about your data.

OpenAI Embedding = Text-embedding-3-Large Model

Embeddings LLMs generate numerical representations of data called embeddings. When filtering your data for relevance, LlamaIndex will convert queries into embeddings, and your vector store will find data that is numerically similar to the embedding of your query.

An embedding is a sequence of numbers that represents the concepts within content such as natural language or code. Embeddings make it easy for machine learning models and other algorithms to understand the relationships between content and to perform tasks like clustering or retrieval.

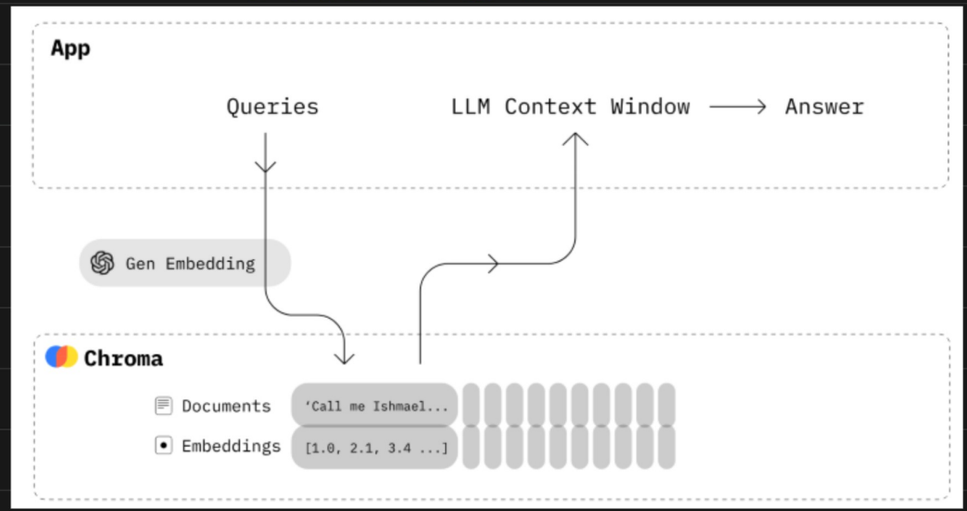
STORING STAGE

Storing: once your data is indexed you will almost always want to store your index, as well as other metadata, to avoid having to re-index it.

ChromaDB

Chroma gives you the tools to:

- store embeddings and their metadata
- embed documents and queries
- search embeddings



QUERING STAGE

Querying: for any given indexing strategy there are many ways you can utilize LLMs and LlamaIndex data structures to query, including sub-queries, multi-step queries and hybrid strategies.

1) Create Custom Prompt Template

2) Give the Context

3) Configure retriever

4) Configure response synthesizer

5) Assemble our query engine

6) Finally - we will ask questions over our data.

Putting it all together

There are endless use cases for data-backed LLM applications but they can be roughly grouped into three categories:

Query Engines: A query engine is an end-to-end pipeline that allows you to ask questions over your data. It takes in a natural language query, and returns a response, along with reference context retrieved and passed to the LLM.

Chat Engines: A chat engine is an end-to-end pipeline for having a conversation with your data (multiple back-and-forth instead of a single question-and-answer).

Agents: An agent is an automated decision-maker powered by an LLM that interacts with the world via a set of tools. Agents can take an arbitrary number of steps to complete a given task, dynamically deciding on the best course of action rather than following pre-determined steps. This gives it additional flexibility to tackle more complex tasks.