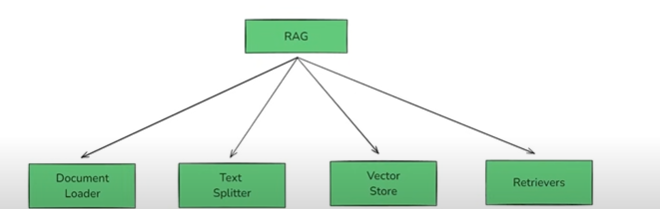
RAG ( **Retrieval-Augmented Generation**)

RAG is a method that improves how AI answers questions by retrieving relevant information from external sources (like documents or databases) and then generating a response using that information.

Think of it as:

🔍 Search first → 🧠 Think next → ✍️ Write the answer



It has got the following components

1. Document Loader – Kisi bhi data source se data load karsakte ho
2. Text Splitter – Bade sentence ko chote chunks mein divide karna
3. Vector Store – Store the embeddings of these chunks in a store
4. Retrieval – Retrieve the most closest context based embeddings
5. Generation - Combines the retrieved chunks with the user’s question and sends it to a language model (like GPT) to generate a final answer

**Document Loader**

* **What it does:** Loads data from various sources like PDFs, websites, databases, or text files.
* **Why it's important:** You need content to search from — this is how you bring it in.
* **Example:** Load a company’s internal wiki or product manuals.

**b) Text Splitter**

* **What it does:** Breaks large documents into smaller, manageable chunks.
* **Why it's important:** Smaller chunks improve retrieval accuracy and reduce memory load.
* **Example:** Split a long article into paragraphs or sections.

**c) Embedding & Vector Store**

* **What it does:** Converts each chunk into a vector (a list of numbers) using an **embedding model**, and stores them in a **vector database**.
* **Why it's important:** Vectors help the system understand meaning and similarity between texts.
* **Example tools:** FAISS, Pinecone, Weaviate

**d) Retrieval**

* **What it does:** When a user asks a question, the system finds the most relevant chunks by comparing vector similarity.
* **Why it's important:** This step ensures the model gets the right context before answering.
* **Example:** For “What is RAG?”, it retrieves chunks that explain the concept from stored documents.

**e) Generation**

* **What it does:** Combines the retrieved chunks with the user’s question and sends it to a language model (like GPT) to generate a final answer.
* **Why it's important:** This is where the actual response is created using both the question and the retrieved knowledge

**Working in detail**

**Why RAG?**

* Traditional language models (like GPT) rely only on what they were trained on. If you ask about something new or specific, they might guess or hallucinate.
* They have been trained on so much of data and they store their knowledge in **parameters ( Paremetric Knowledge )**  but will this knowledge be helpful if I ask something that is beyond the date till which it is trained? ( **knowledge cutoff date** )
* Also lets say I have to ask some very specific business requirement question will that be able to help? It might , it might not ( The accuracy for it would be less )
* Sometimes the LLM returns factually incorrect information with confidence

Mota Mota 3 situations

1. Asking questions related to private data
2. Asking questions after the knowledge cutoff date( most llms )
3. Hallucinations

Okk , so is there any way to solve these issues?

**Use Fine Tuning**

1. Ham pre-trained model ko fir se train karte hai us data se jo hame chahiye use knowledge dene ke liye

Analogy

Student -> LLM

Engineering Curriculum -> Pretraining

Company -> Fine Tuning

1. Ways of fine tuning

- Supervised Fine Tuning -> labelled data dete ho ( prompt -> desired output )

- Continued pre training ( Unsupervised) -> example yt video ko train karna hai

Transcript uthao aur use de do model ko

- RLHF , LORA , QLoRA

1. Normally we use supervised method

d) Cons of fine tuning

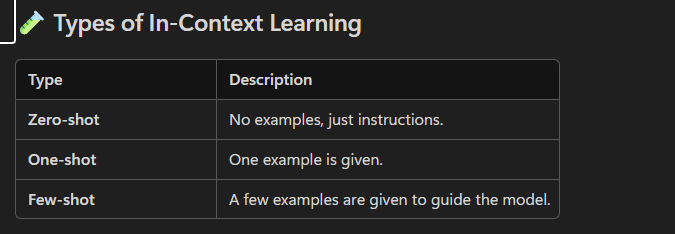
- Computationaly expensive

- Proper technical expertise needed

- Agar baar baar data add karna hai to fine tuning baar baar karna padega

**Use In Context Learning**

1. In context learning the LLMs like ( GPT3/4, Claude and Llama ) learn by using the examples given in the prompt ( Sabhi models ni karenga aisa obviously )



Now comes RAG?

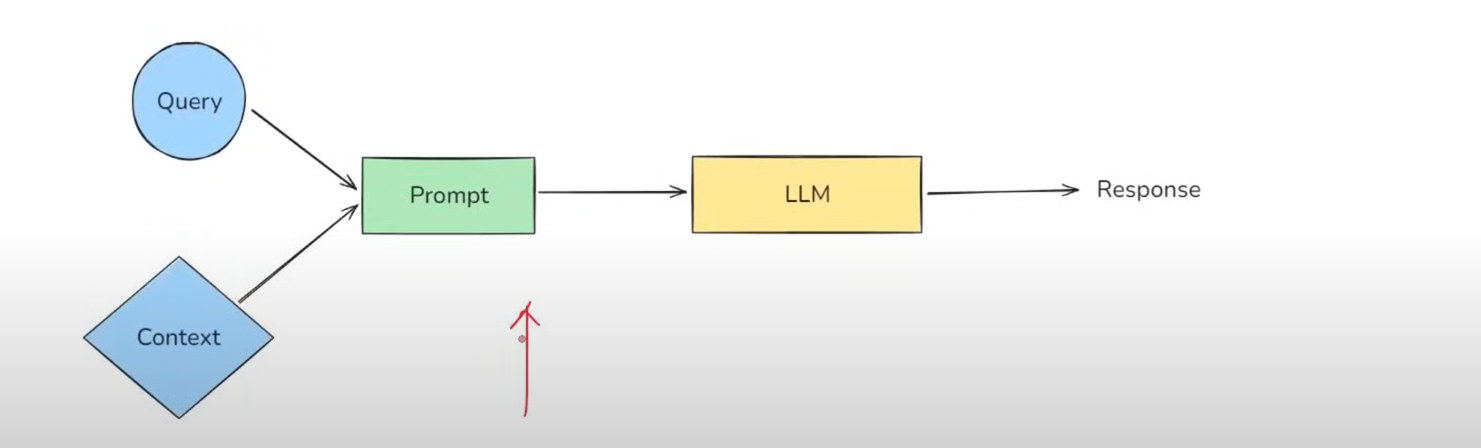
So ham LLM ko sirf prompt ni dete hai usko us prompt se related Context bhi dete hai jiski help se wo hamein curated answer de sakta hai

Ex – lets say hamein ek private video dekh rahe hai , agar doubt aya to ?? Direct LLM se poochoge ki is point pe kya hua to kya wo bata paega?? Ni bata paega tab ab kya karein??

Kuch ni bas jo prompt ya question bheja hai use transcript se sirf utna context bhi de dunga

Tab wo ache se answer kar paega

And this is **RAG**



Note:

Ham use example ni de rahe hai lekin usko utna data de rhe hai jisse wo query ko solve kar sakta hai

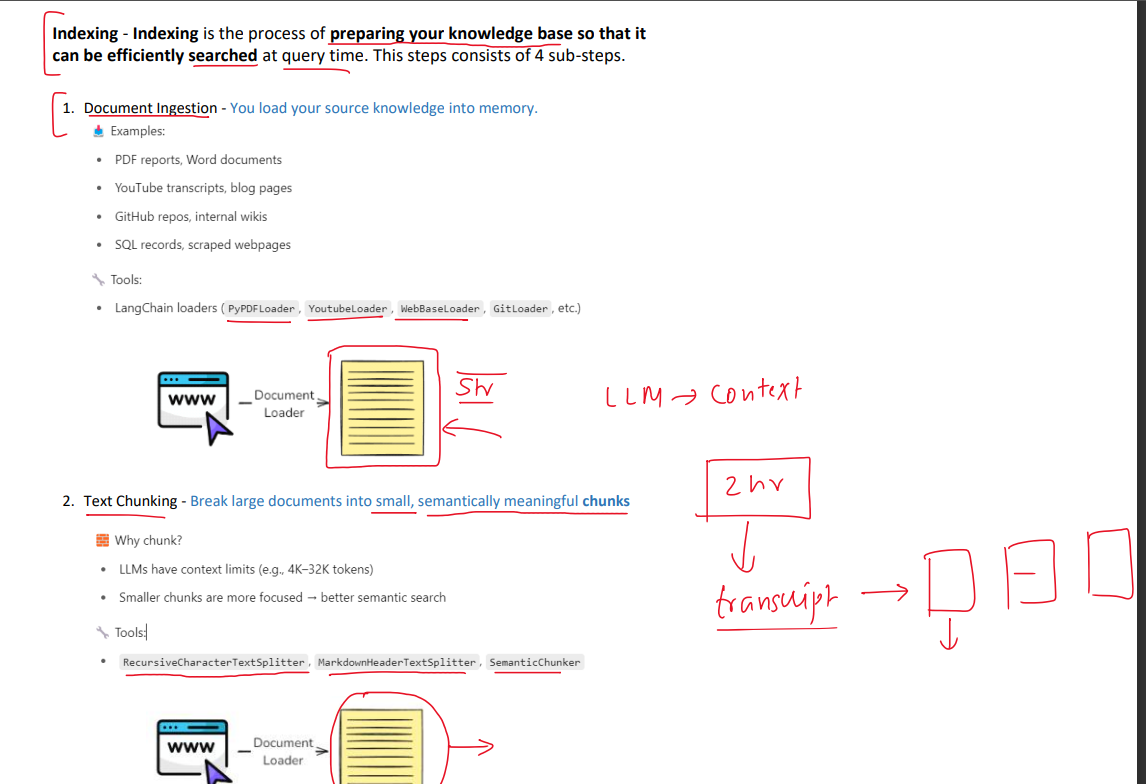
**Understanding RAG**

Broadly there are four steps in a RAG based application

1. **Indexing** – Jo ham data late hai alag alag sources se aur use store karte hai usi process ko indexing kahte hai – ( basically external knowledge base create karna )
2. **Retrieval** – Jab wo query ka use karte hue context(external knowledge base) mein dhoondhta hai ki kaunse wo chunks hai jo mere is solution ko solve karne mein madad karenge ( Man lo 2 hrs ka video hai to tum use poora 2hr ka transciprt to bhejoge ni, bas utna hi data bhejna jitna usko need hai )
3. **Augmentation –** Jab query+retrieved data ko add karke prompt banate ho
4. **Generation –** Response generation

**RAG PIPELINE IN DETAIL**

**INDEXING**

****

A screenshot of a computer screen

AI-generated content may be incorrect.

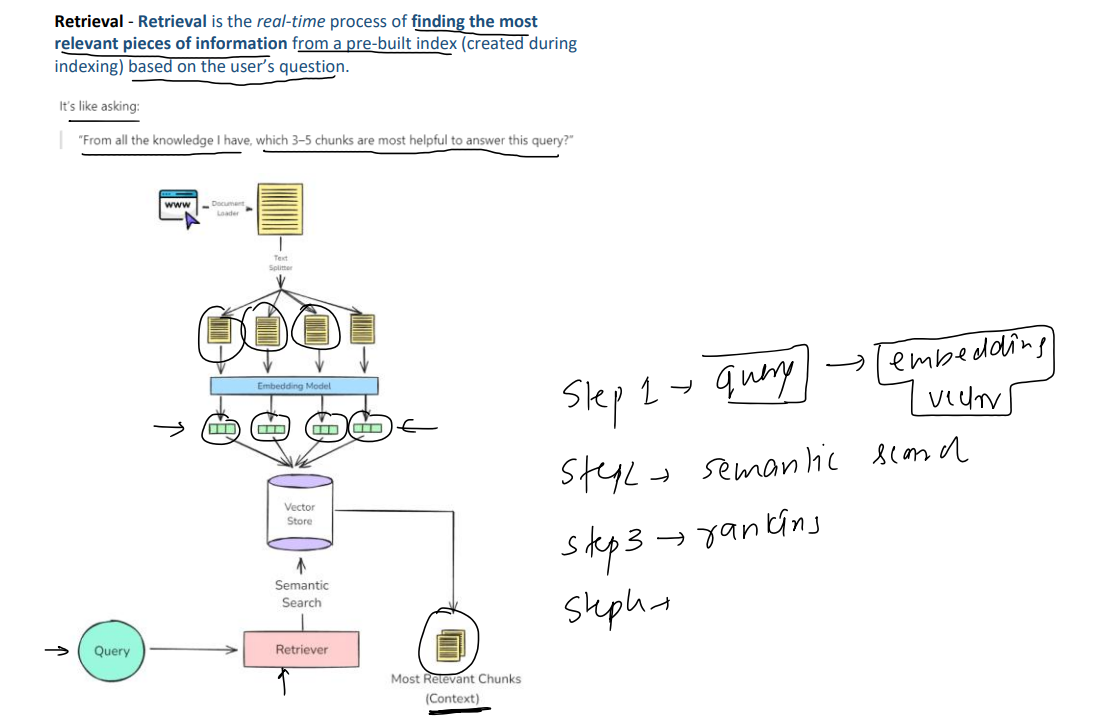
A screenshot of a computer

AI-generated content may be incorrect.

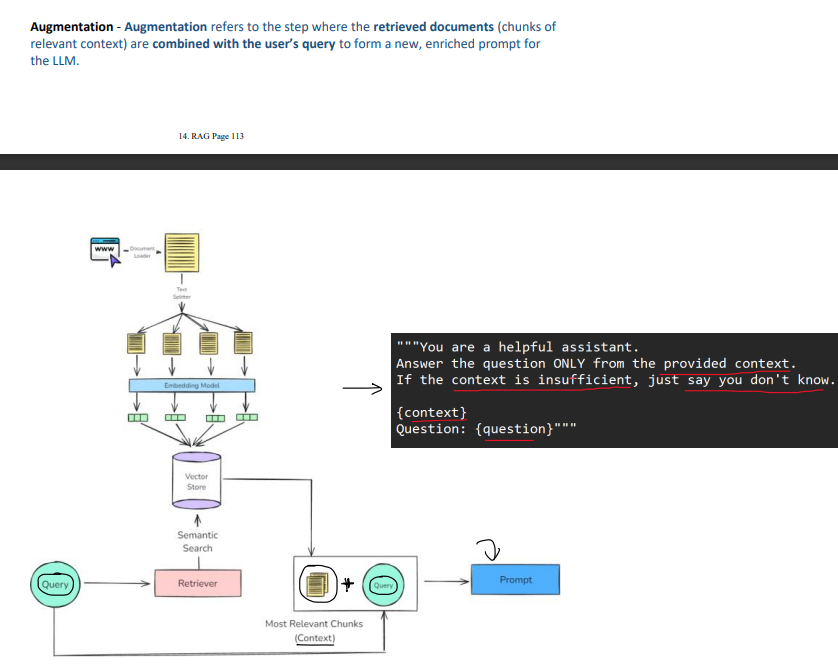
A screenshot of a computer

AI-generated content may be incorrect.

**RETRIEVAL**



**AUGMENTATION**

****

**GENERATION**

**A diagram of a software development

AI-generated content may be incorrect.**