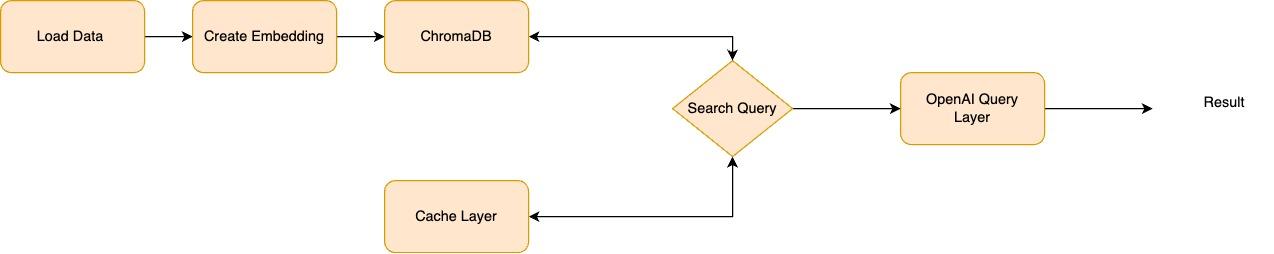
## Introduction

Creating a generative search system capable of searching a plethora of product descriptions to find and recommend appropriate choices against a user query. We are using this [Myntra dataset](https://www.kaggle.com/datasets/djagatiya/myntra-fashion-product-dataset) on Kaggle to build such a system.

## System design

Here is the system design block diagram.



We are using the data from Kaggle and uploading it on Google Drive to be further processed.

After reading the data, we are embedding it in the chromaDB vector store. Along with this, we are also using a separate vector store just for the sake of caching similar results so that we don’t need to search in the vector store every time.

Once the data is embedded, we can make the search query and first, we will check the cache if similar results exist, if it does, we will return the result from the cache for further processing.

If data doesn't exist, we will search in the main collection and upon receiving the result, we will cache it and then send it to the next layer.

Finally, we will sort and rank the result and pass it to the OpenAI prompt and get the result from ChatGPT.

## Implementation

We are using Openai, chromaDB, and sentence-transformer library to process the results. We are also using pandas for reading the data frame and processing it. The code is in the notebook file.

## Challenges and Lessons Learned

One of the key challenges was the data, the data size is huge and the openAI model I am using has a limit. So I shrunk the data to 1000 rows for now. What I learned is that the OpenAI embedding model has limits and we need to choose different models for production-size applications.

Another challenge was the image column, the dataset containing images was more than 4 GB in size and I couldn’t afford that in my Google account. I made a conscious call to remove that column as we won’t be using it for search but mainly for presentation.

Also, the links in the dataset were of bad URLs, so it won’t be of any use either.

Another key challenge was which data to use for documents and which for metadata. Initially, I made a mistake and combined multiple columns for documents and some for metadata. However, that confused the database even more and it was giving weird results.

To get the idea, I went to Myntra and started searching for items. What I found was that the title or name of the product is the key thing to look for and the rest is the metadata. Then I modified the data frame according to this and the results were really good.