# Fashion Search AI System Using Gen AI: Project Overview and Analysis

### **Objectives:**

The primary goal of this project is to create an Al-powered fashion query response system that delivers detailed and user-friendly answers to fashion-related questions. The system is designed to improve the user experience by generating contextually relevant and informative responses, helping users discover fashion items that match their preferences.

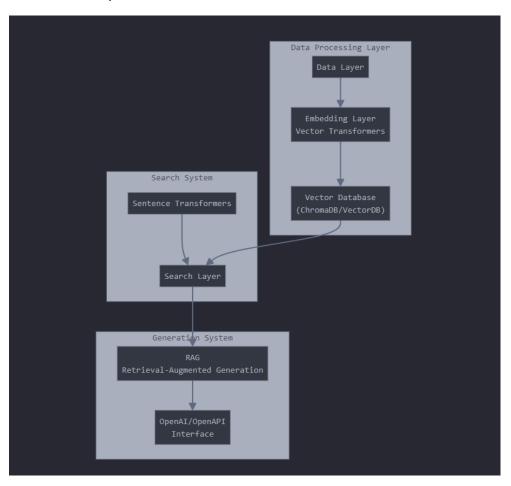
### System Design:

The project is structured into two key layers: the search layer and the generation layer.

- Search Layer: This layer retrieves relevant fashion items from the dataset using keyword matching or other predefined criteria.
- Generation Layer: Leveraging advanced AI models like GPT-3.5, this layer generates detailed, context-aware responses to user queries, producing natural language outputs that enhance user understanding.

### **Architecture:**

Architectural Framework Implemented in this Solution:



### Implementation:

The implementation comprised several key steps, including data preprocessing, model integration, and the generation of query responses.

### Data Preprocessing:

The CSV dataset was reformatted to improve its quality and readability. Blank entries were filled, decimal points were standardized, and irrelevant columns were removed. Additionally, text columns were cleaned to eliminate HTML tags and extraneous characters.

### Model Integration:

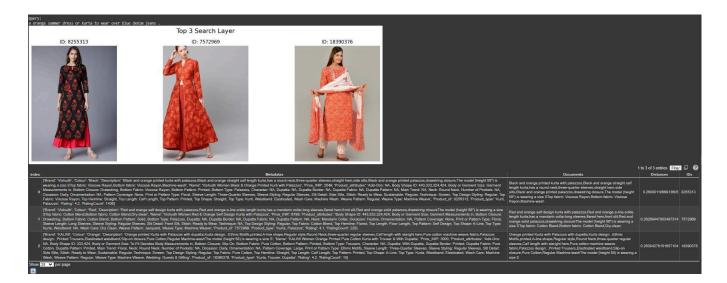
The system was enhanced by integrating advanced AI models like GPT-3.5 to generate responses to user queries. These queries were processed by the model, which produced detailed and contextually appropriate answers.

### Query Response Generation:

User queries were processed through both the search and generation layers. The search layer identified relevant fashion items from the dataset, while the generation layer used AI models to craft detailed responses. These responses were contextually rich and presented in natural language. Here are the few queries that were used to test the model:

#Query 1: an orange summer dress or kurta to wear over blue denim jeans.

Search Layer:



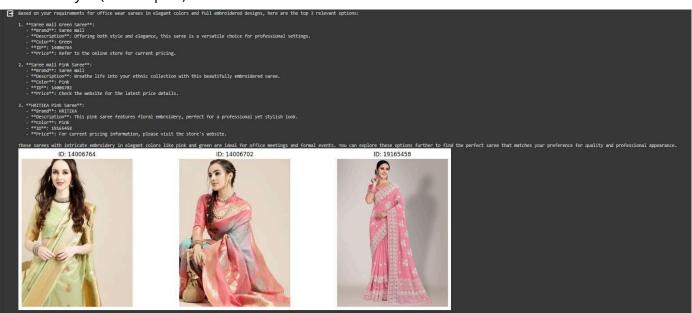
Generation layer:

#Query 2: I'm looking for office wear sarees in elegant colors like pink, violet, or green. I prefer sarees with full embroidered designs, suitable for professional settings. There's no specific price range, as quality is my priority. These sarees will be worn for office meetings and formal events.

### Search Layer:



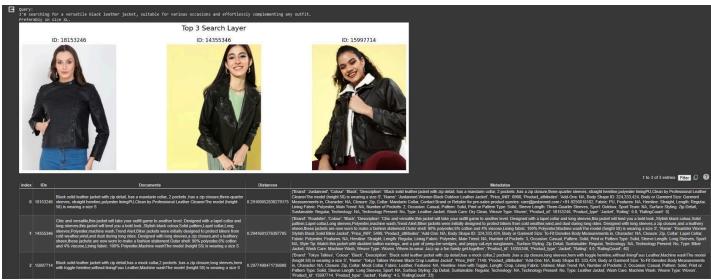
### Generation layer:(2 examples)

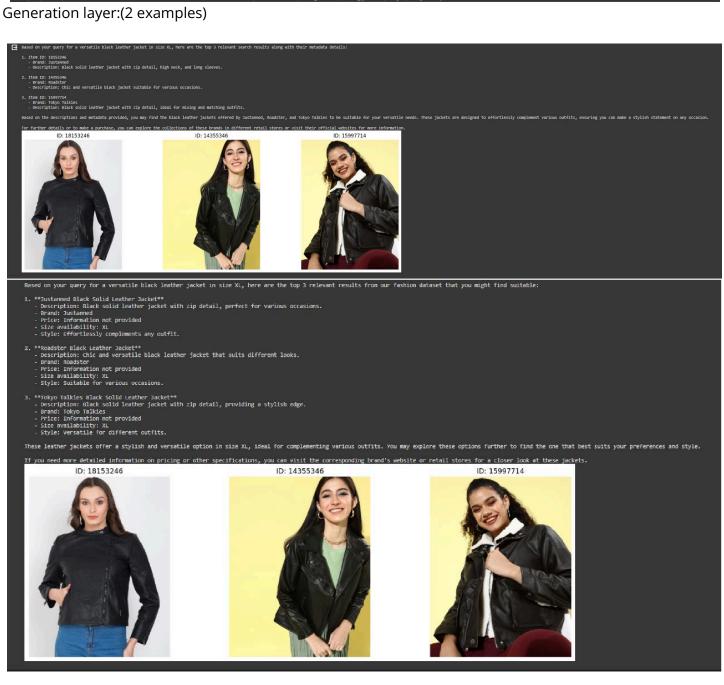




#Query 3: I'm searching for a versatile black leather jacket, suitable for various occasions and effortlessly complementing any outfit. Preferably in size XL.

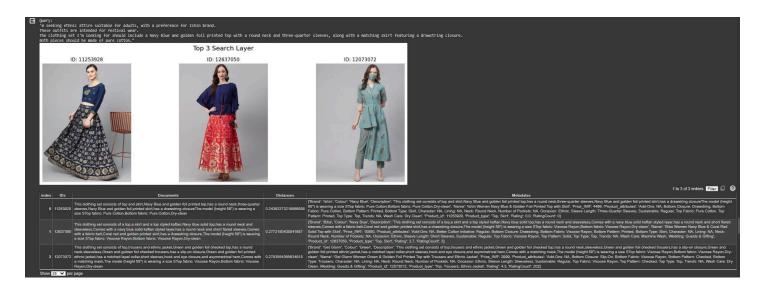
#### Search Layer:



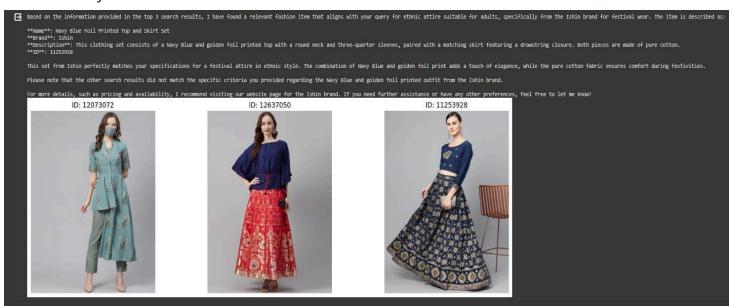


#Query 4: "I'm seeking ethnic attire suitable for adults, with a preference for the Ishin brand. These outfits are intended for festival wear. The clothing set I'm looking for should include a Navy Blue and golden foil printed top with a round neck and three-quarter sleeves, along with a matching skirt featuring a drawstring closure. Both pieces should be made of pure cotton.

#### Search Layer:



#### Generation layer:



Several other queries were used during the project, but for the sake of brevity, I have listed only a few here, as well as in the code terminal.

### Challenges:

Several challenges arose during the implementation process:

- Metadata Processing: Initially, there were issues with processing metadata, which impacted the
  display of results. This was addressed through debugging and refining the metadata handling
  code.
- Dataset Chunking: Loading the entire dataset without chunking presents memory constraints.
   However, due to time limitations and the manageable size of the dataset, chunking was not implemented.

### Lessons Learned:

- Data Preprocessing: Ensuring proper data preprocessing is essential for maintaining data quality and readability.
- Al Model Integration: Incorporating advanced Al models can greatly improve the system's ability to generate detailed and contextually relevant responses.
- Handling Large Datasets: Managing large datasets requires careful attention to memory constraints and the implementation of efficient data processing strategies.

## Future Scope:

As a future scope, the project can be extended by implementing it as a Flask web application and hosting it on a website. This would provide better interaction and aesthetics for users. Additionally, rephrasing prompts and introducing interactive sessions with criteria-based filters could further enhance the system's performance and user experience.

### Conclusion:

- Comparison of Search and Generation Layers:
  - a. The generation layer produces more detailed and comprehensible results compared to the search layer.
  - b. The search layer retrieves relevant information through keyword matching or predefined criteria.
  - c. The generation layer uses advanced AI models like GPT-3.5 to dynamically generate context-aware responses with tailored, informative answers.
- Advantages of the Generation Layer:
  - a. It understands the context of queries and generates natural language responses, improving overall readability and user engagement.
  - b. The flexibility of the generation layer allows it to adapt to various query formats and provide customized responses based on user preferences.
- Value and Versatility:
  - a. The generation layer is highly valuable for tasks requiring a comprehensive understanding of queries and the effective communication of information.
  - b. Its ability to generate detailed, user-friendly responses makes it ideal for tasks that demand clarity and engagement.

#### Summary:

- a. While the search layer is efficient for retrieving relevant information, the generation layer significantly enhances the output quality and readability.
- b. The generation layer is the preferred choice for tasks that require detailed, well-structured, and user-friendly responses.