# Project Report: Retrieval-Augmented Generation (RAG) System for Myntra Dress Recommendations

**1. Project Goals**

The primary objective of this project is to build a Retrieval-Augmented Generation (RAG) system that can efficiently recommend dresses from the Myntra dataset based on user queries. The system is designed to leverage a combination of retrieval mechanisms and generative models to provide relevant and contextually appropriate product descriptions to users.

**2. Data Sources**

* **Myntra Dress Dataset**: The dataset used in this project comprises various dresses listed on Myntra through kaggle. The key features include product descriptions, which are the primary text data used for generating embeddings and retrieving relevant results.
* **External API**: The OpenAI API is used to generate text embeddings and power the generative model. The model "text-embedding-ada-002" is utilized to convert textual data into high-dimensional embeddings, which can be efficiently indexed and retrieved.

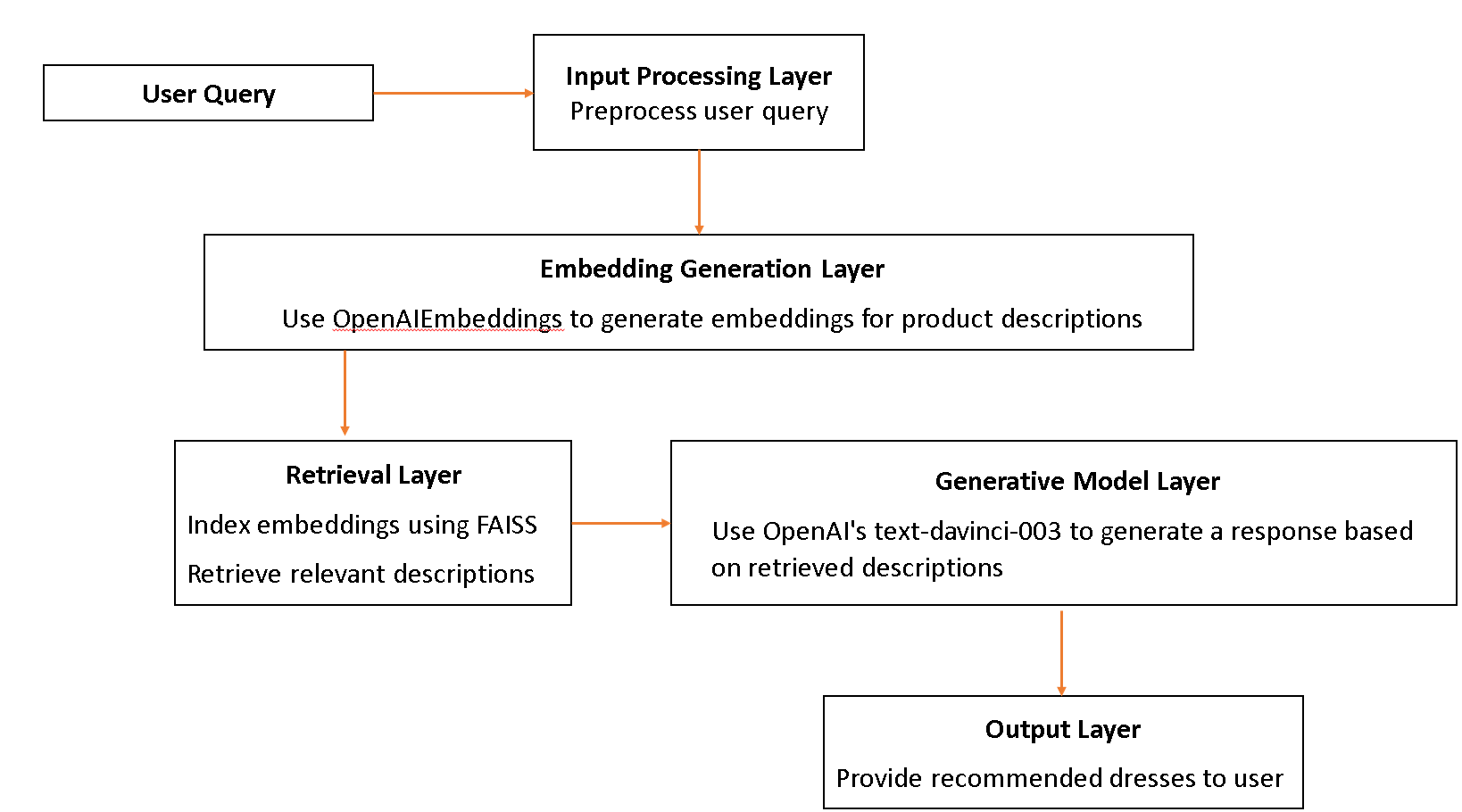
**3. System Design Choices**

* **Embedding Model**: The OpenAIEmbeddings model, specifically the "text-embedding-ada-002" variant, was selected for generating embeddings of product descriptions. This model is known for its balance of performance and computational efficiency.
* **Indexing and Retrieval**: The FAISS library was chosen to index the generated embeddings. FAISS (Facebook AI Similarity Search) is optimized for large-scale similarity searches, making it ideal for retrieving relevant dress descriptions based on user queries.
* **Generative Model**: The RAG system integrates a generative model (text-davinci-003) from OpenAI, which is responsible for producing human-like responses based on the retrieved data. This model is used to enhance the retrieval results by generating a more cohesive and contextually appropriate response.

**4. Challenges Faced**

* **Rate Limitations**: During the embedding generation phase, API rate limits were encountered, which led to RateLimitError. This required the implementation of retry logic and batch processing to manage the rate limits efficiently.
* **Data Volume**: The Myntra dataset is extensive, requiring careful management of memory and processing resources. Batching techniques were employed to handle large volumes of data without overwhelming system resources.

**Flow Chart**

****

**7. Conclusion**

This project demonstrates the implementation of a RAG system using the Myntra dataset. The integration of FAISS for efficient retrieval and OpenAI's models for embedding generation and generative responses creates a robust system capable of providing relevant and contextually rich recommendations based on user queries. The challenges faced, such as rate limitations and data volume, were addressed through strategic design choices and implementation of retry and batching logic.