Advanced Certificate Program in Generative AI - December 2023

Vu Tuan Anh

PROBLEM STATEMENT

Our main objective is to bridge the gap between the convenience of online shopping and the personalized experience of offline shopping. Here are some of the major difficulties faced by online shoppers:

- **Lack of personalized assistance:** Unlike a salesperson in a physical store, online shopping often lacks individual guidance in choosing the right product.

- **Difficulty in product evaluation:** Assessing a product's quality and suitability can be tedious, often requiring sifting through numerous reviews.

- **Uncertain fit and look:** For fashion items like clothing and sunglasses, online shoppers struggle to visualize how the product would look on them.

SOLUTION

Shopping Assistant: Bridging the Online Shopping Gap

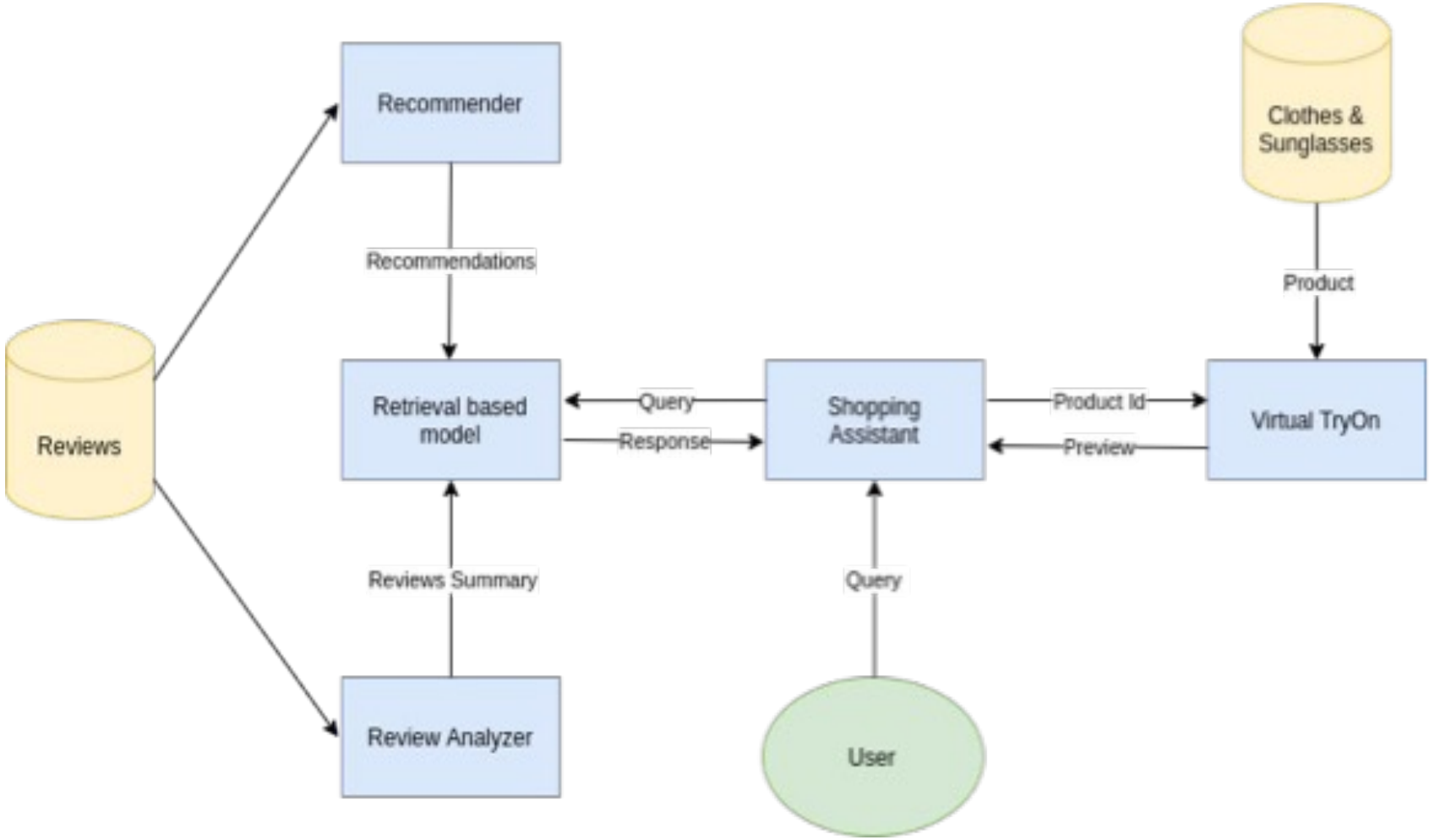
The Shopping Assistant, a chatbot, can address the challenges of online shopping by offering personalized assistance to consumers.

- **Product Recommendations:** Based on a user's needs and preferences, the Shopping Assistant can suggest relevant products, similar to a helpful salesperson.

- **Review Analysis:** The Assistant can provide summaries of product reviews, saving consumers time and effort in sifting through individual reviews. This helps them make informed decisions.

- **Virtual Try-On Experience:** For fashion items like dresses or spectacles, the Shopping Assistant can offer a virtual try-on experience. This allows users to see how a product would look on them in real-time, overcoming a major limitation of online shopping.

WORKFLOW


Recommender
Clothes & Sunglasses
Recommendations
Product
Reviews
Retrieval based model
Query
Response
Shopping Assistant
Product Id
Preview
Virtual TryOn
Reviews Summary
Query
Review Analyzer
User

METHODOLOGY

CHATBOT

Retrieval-Based Model for Efficient Responses

The Shopping Assistant utilizes a retrieval-based model to efficiently deliver responses to user queries. This model employs a technique called **weighted TF-IDF** in conjunction with **cosine similarity**. Weighted TF-IDF helps identify the most relevant keywords within product descriptions and reviews, while cosine similarity measures the similarity between a user's query and these keywords. This combined approach allows the Shopping Assistant to quickly retrieve the closest pre-defined response that best addresses the user's needs.

Pre-defined Response Categories

The Shopping Assistant's pre-defined responses fall into four main categories:

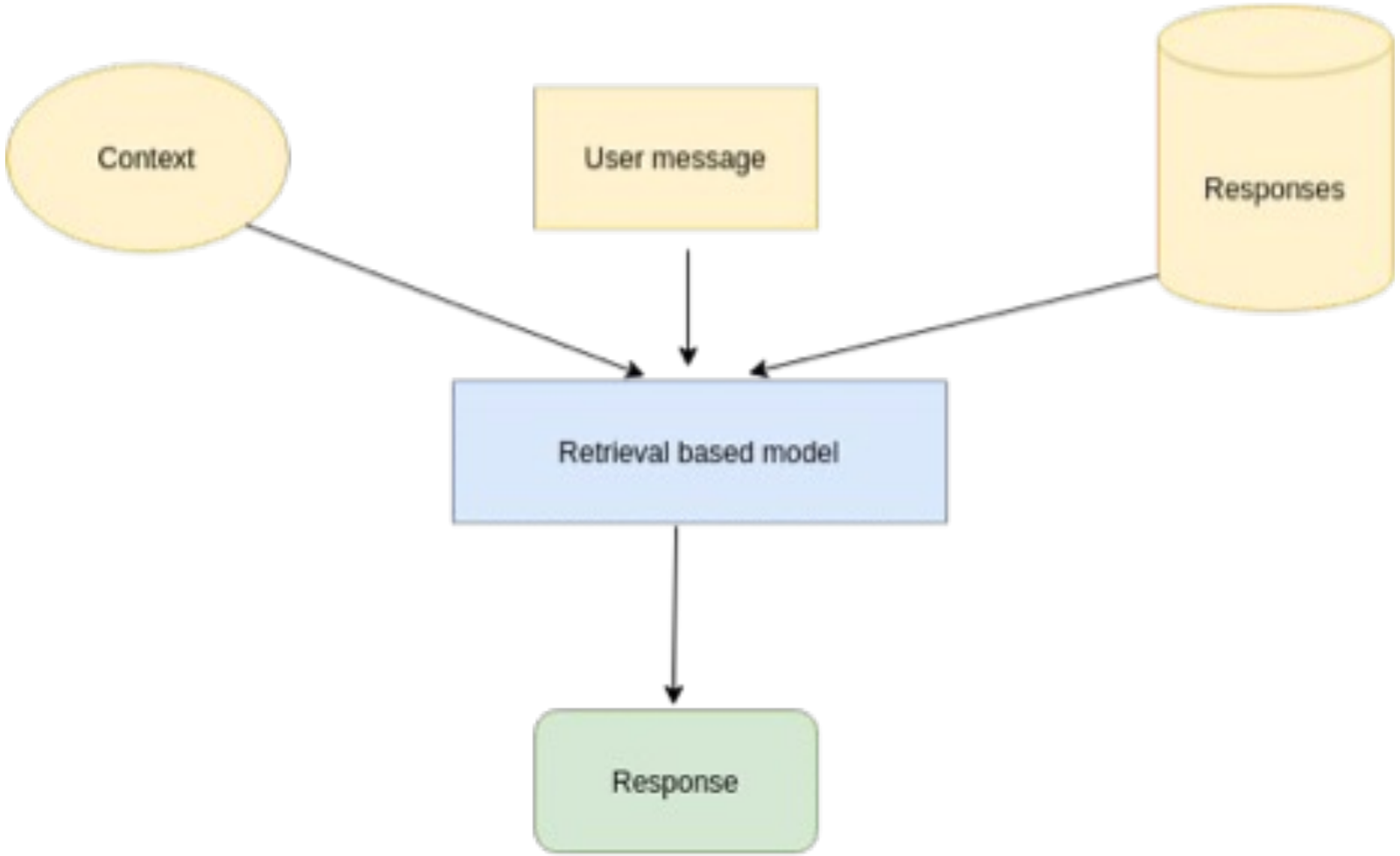
1.Product Recommendations: Suggesting relevant products based on user preferences.

2.Review Summaries: Providing concise summaries of product reviews to aid informed decisions.

3.Virtual Try-On Experience: Offering real-time virtual try-on functionality for specific fashion items like T-shirts and sunglasses.

4.General Conversation: Handling greetings, thanks, and other general conversational interactions.

RETRIEVAL BASED MODEL



VIRTUAL TRY-ON

DATA PREPROCESSING

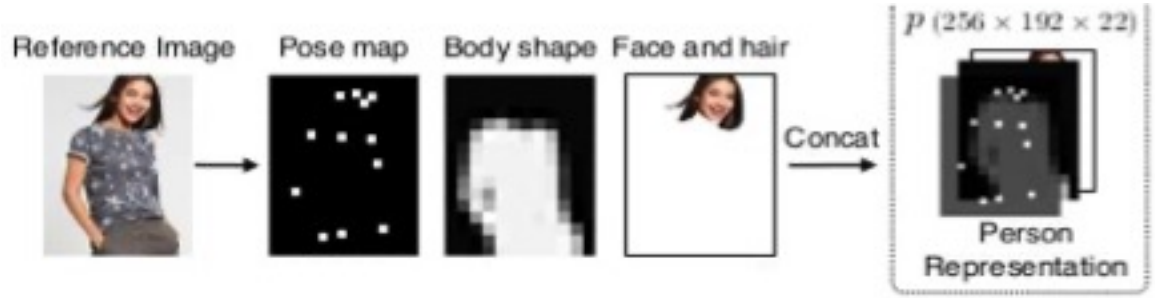
Clothing-Agnostic Person Representation

The GMM (Geometric Matching Module) utilizes a clothing-agnostic person representation to understand the user's body shape and pose. This representation combines three key components:

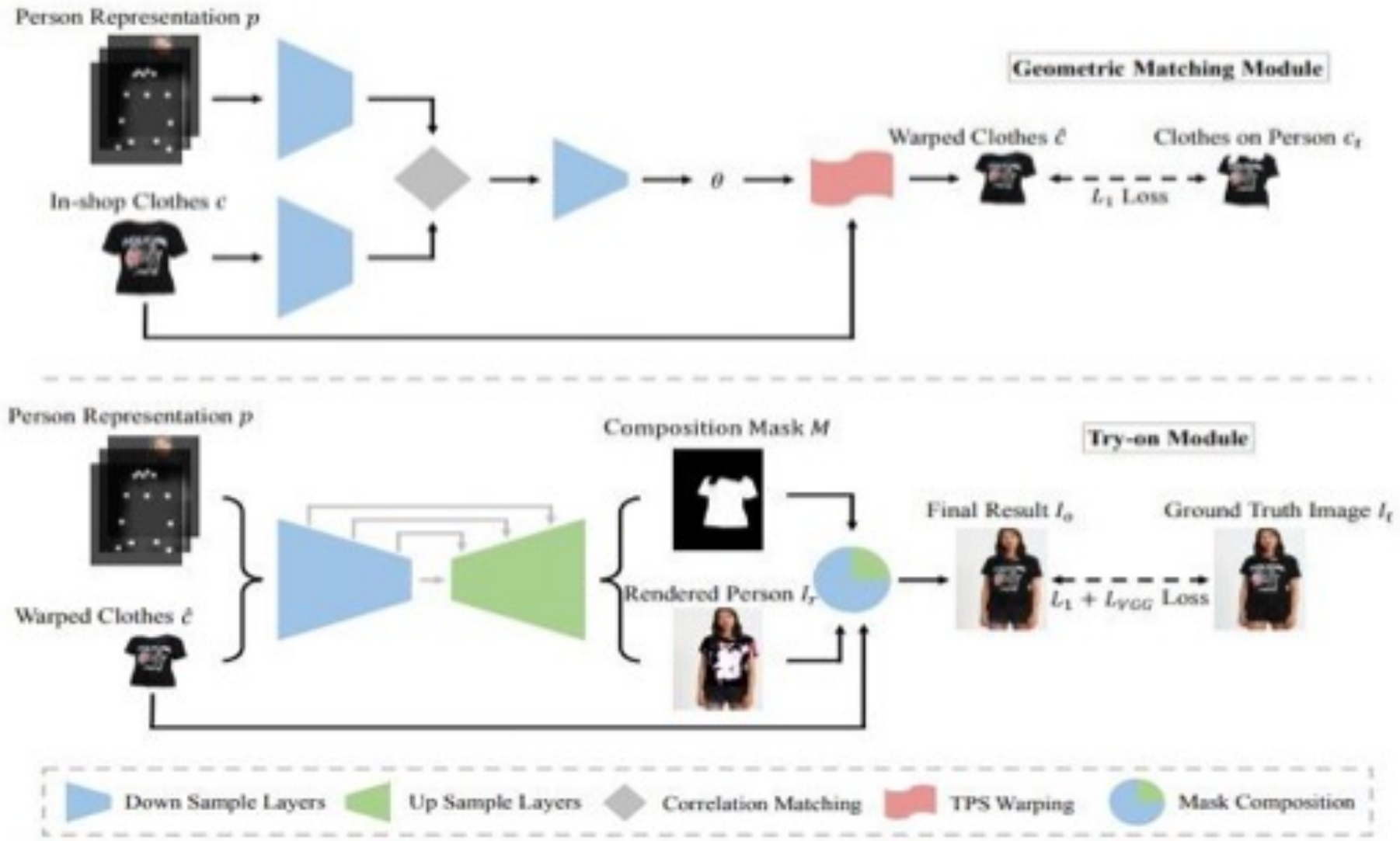
Pose Heatmap: This is a heatmap generated using the "OpenPose" library. It captures the location and intensity of **18 key body joints**, providing information about the user's posture.

Body Mask: This is a single-channel binary mask that identifies different body parts. It essentially acts as a segmentation map, separating the body from the background.

Reserved Regions: This is a three-channel RGB image that preserves the user's face and hair. This information is crucial to maintain a realistic appearance in the final try-on image.



VIRTUAL TRY-ON ARCHITECTURE



VIRTUAL TRY-ON ARCHITECTURE

The architecture consists of two main components:

Geometric Matching Module (GMM): This module transforms the target garment into a warped form that aligns with the person's body shape and pose.

Try-On Module (TOM): This module integrates the warped garment with the image of the target person and synthesizes the final try-on image:

- Mainly consists of changed to consists of for a more concise structure.
- Cloth changed to garment for a more specific term in clothing.
- Warped cloth changed to warped form for improved readability.
- Try-on result changed to try-on image for a more specific output.

GEOMETRIC MATCHING MODULE (GMM)

GEOMETRIC MATCHING MODULE (GMM)

GMM is used to transform the target clothing **c** into a warped version **ĉ** that is roughly aligned with the input person representation **p**. GMM consists of four main components:

1.Feature Extraction Networks: Two separate neural networks are used to extract high-level features from the target clothing **c** and the person representation **p**. These networks utilize downsampling convolutional layers to achieve this.

2.Correlation Layer: The extracted features from both networks are combined into a single tensor using a correlation layer. This combined tensor is then fed into the regressor network.

3.Regressor Network: This network predicts the spatial transformation parameters, denoted by **θ**. These parameters define how the target clothing needs to be warped.

4.Thin-Plate Spline (TPS) Transformation Module: This module utilizes the predicted parameters **θ** to warp the target clothing **c** into the final output **ĉ**. The transformation is achieved using the Thin-Plate Spline (TPS) method.

Try-On Module (TOM)

Try-On Module (TOM)

The Try-On Module (TOM) acts as a **synthesizer**, not a generator, to create the final output image. This image represents the desired clothing virtually "tried on" by the input person.

- Architecture: TOM utilizes an encoder-decoder architecture similar to U-Net. This architecture takes a concatenated input of the person representation **p** and the warped clothing **ĉ**.

- Image Rendering and Mask Prediction: The network simultaneously renders a person image I_r and predicts a **segmentation mask** denoted by **M**. This mask defines the regions where the warped clothing and the person image should be combined.

- Image Synthesis: Finally, the rendered person image. I_r and the warped clothing **ĉ** are fused together using the segmentation mask **M** to synthesize the final try-on image I_o. This is achieved using the following formula:

$$I_o = M * \hat{c} + (1 - M) * I_r$$

DATASET

DATASET

The dataset used for training the model is MPV (Multi-Pose Virtual Try-on).

It contains 37,723 images of people in various poses, along with 14,360 corresponding clothing images. All images have a resolution of 256x192 pixels.

	Train / Test Set	No of Samples
Resolution: 256 x 192	Training Set	52,236
	Test Set	10,544

TRAINING

TRAINING

The GMM module was trained using the pixel-wise L1 loss between the warped result ĉ and ground truth (c_t).

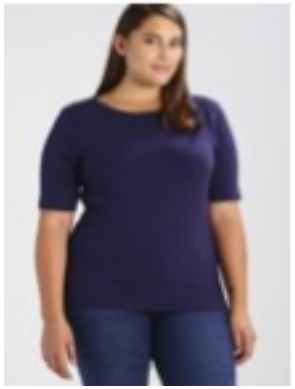
$$\mathcal{L}_{GMM}(\theta) = \|\hat{c} - c_t\|_1 = \|T_\theta(c) - c_t\|_1$$

TOM is trained adversarially **in competition with** a discriminator. This discriminator receives three inputs: the TOM result image (Io), the input clothing image (c), and the person representation (p). The discriminator's task is to **distinguish** whether the resulting image (Io) is a realistic try-on image or a fake one generated by TOM.

- Optimizer used : Adam
- Final loss of generator on validation : 3.62001
- Final loss of discriminator on validation: 0.003821

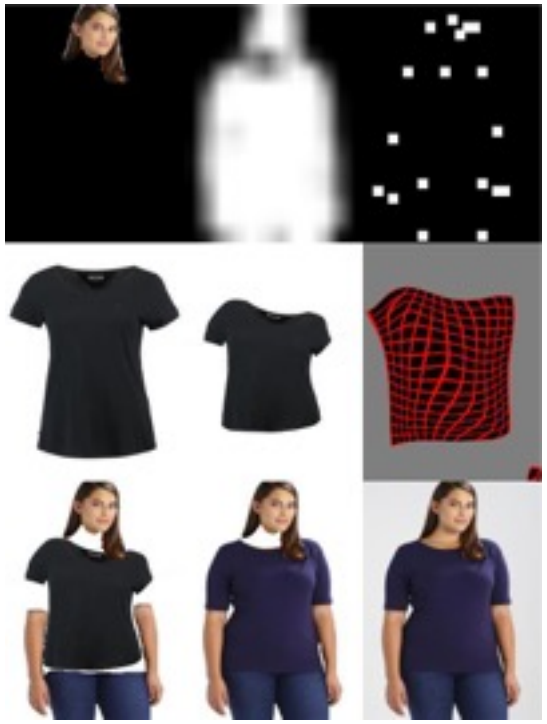
VISUALISATION

VISUALISATION

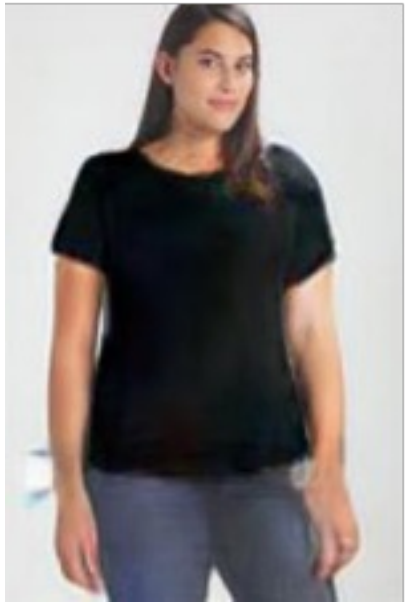


Person

Cloth



GMM result



TOM result

VIRTUAL TRY-ON OF SUNGLASSES

VIRTUAL TRY-ON OF SUNGLASSES

Leverages the Jeeliz library: This library enables real-time face detection and placement of virtual glasses frames onto the user's face, creating an augmented reality (AR) experience.

Minimal Latency: The system operates with very low latency, delivering near real-time performance for a smooth user experience.

Versatile Functionality: The underlying technology can be easily adapted to support virtual try-on for other facial wearables such as masks, jewelry, and caps.

REVIEW ANALYSIS

REVIEW ANALYSIS

The system employs a multi-step process to generate concise and informative review summaries:

1.Data Preprocessing: We leverage the NLTK library to pre-process the raw review data obtained from an online shopping website. This preprocessing step involves removing unnecessary words such as articles (a, an, the) and prepositions (in, on, at, etc.) to focus on the core content of the reviews.

2.Noun Phrase Extraction: An inbuilt tokenizer is utilized to identify relevant nouns and noun phrases within the preprocessed reviews. These phrases typically describe the consumer's experience with the product.

3.Sentence Scoring: The system assigns a score to each sentence based on the frequency of these relevant nouns and noun phrases within the sentence. Sentences containing more frequently occurring product-related words receive higher scores.

4.Summary Generation: Finally, the system ranks the sentences based on their calculated scores. The sentences with the highest scores are then compiled to form the final review summary. This approach ensures that the summary captures the most important aspects of the user reviews.

RECOMMENDER SYSTEM

RECOMMENDER SYSTEM

Product Recommendation System

This module utilizes a collaborative filtering approach to recommend products to users. The system relies on a dataset containing user ratings for various products.

Here's how it works:

1.Similarity Calculation: The system employs the Pearson Correlation Coefficient (PCC) to measure the similarity between users based on their past product ratings. Users with high PCC scores are considered to have similar preferences.

1.Recommendation Generation: For a specific user, the system identifies users with high PCC scores, essentially finding users with similar tastes. Products that these similar users have highly rated are then recommended to the target user. This approach leverages the idea that users with similar preferences are likely to find similar products appealing.

WORKING PROTOTYPE

WORKING PROTOTYPE

See it in action!

Demo Video: https://www.youtube.com/watch?v=o7hb1808GUE

Code and Setup Instructions: https://github.com/MrVuTuanAnh/GEN_AI

SCREENSHOTS

SCREENSHOTS

Available at:

https://github.com/MrVuTuanAnh/GEN_AI/tree/main/screenshots

TECH STACK

TECH STACK

Frontend: React.js - Leverages a virtual DOM for efficient rendering and simplifies complex UI creation.

Backend: FastAPI - This high-performance Python framework streamlines web server development and facilitates seamless Machine Learning model integration.

Virtual Try-On: PyTorch - Chosen for its dynamic computational graph, ideal for this application.

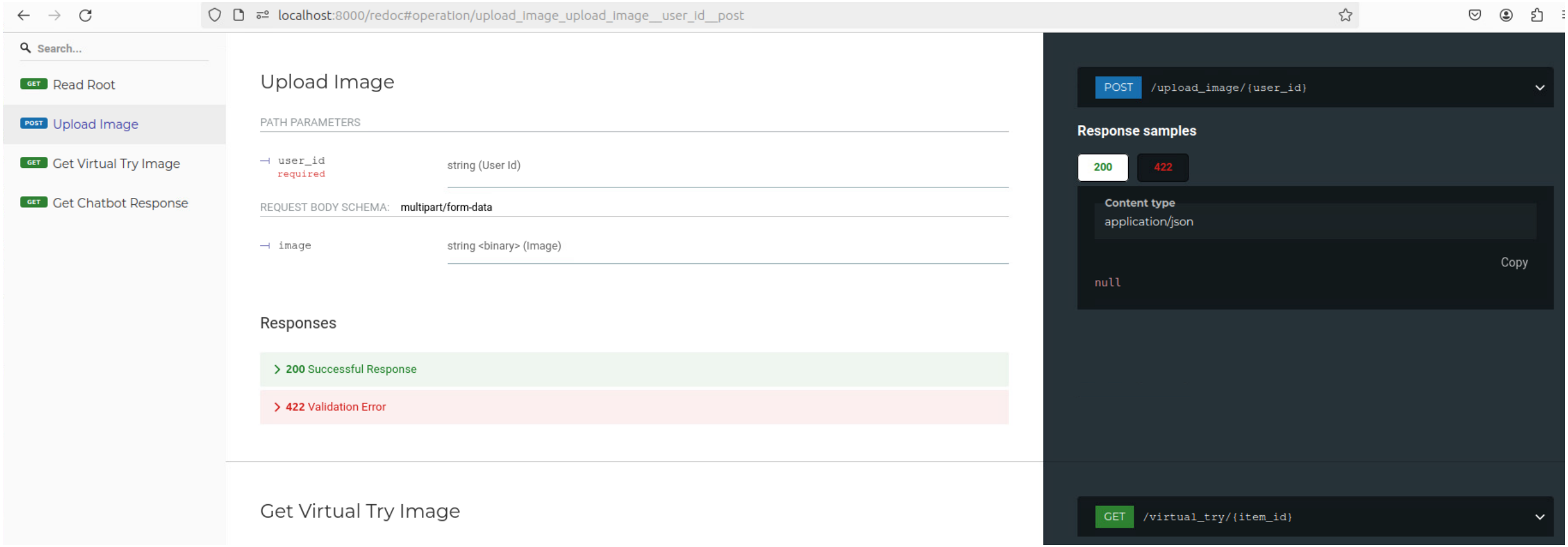
Natural Language Processing (NLP): NLTK - Provides pre-trained models and extensive data corpora, enabling swift and effortless text processing and analysis.

Deployment: Docker & Docker Compose - Ensure application portability by containerizing both the web application and server.

API DOCUMENT

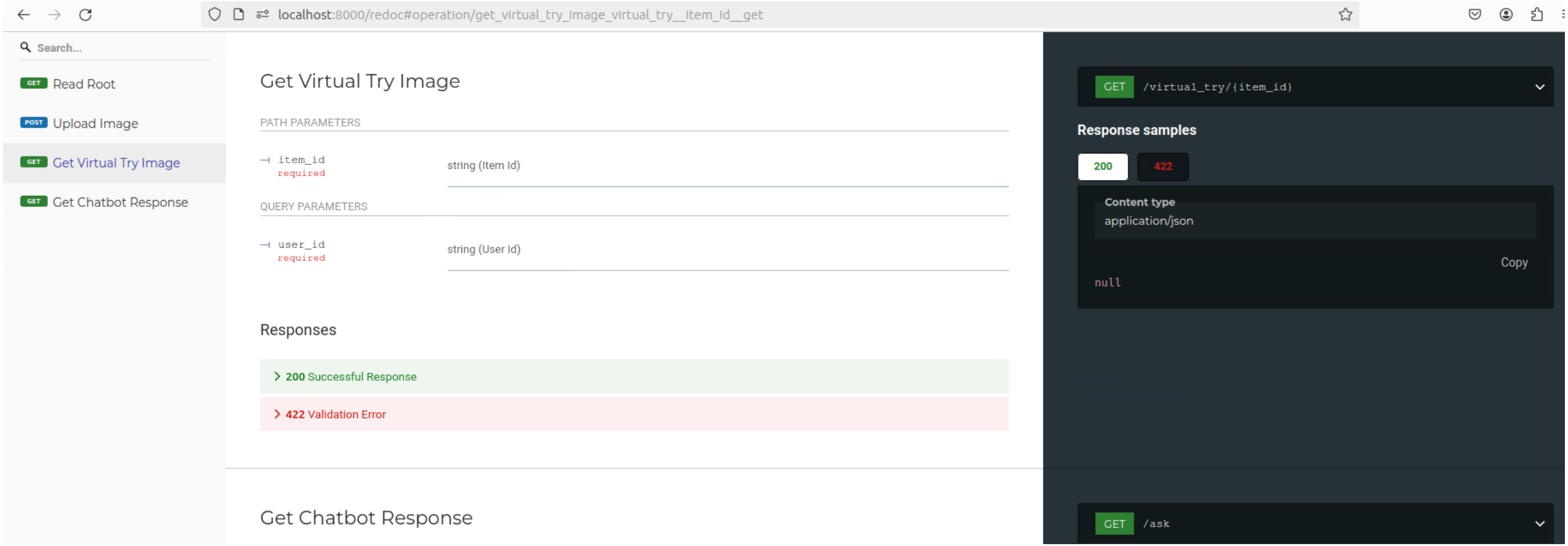
API DOCUMENT

Upload Image API: http://localhost:8000/redoc#operation/upload_image_upload_image__user_id_post



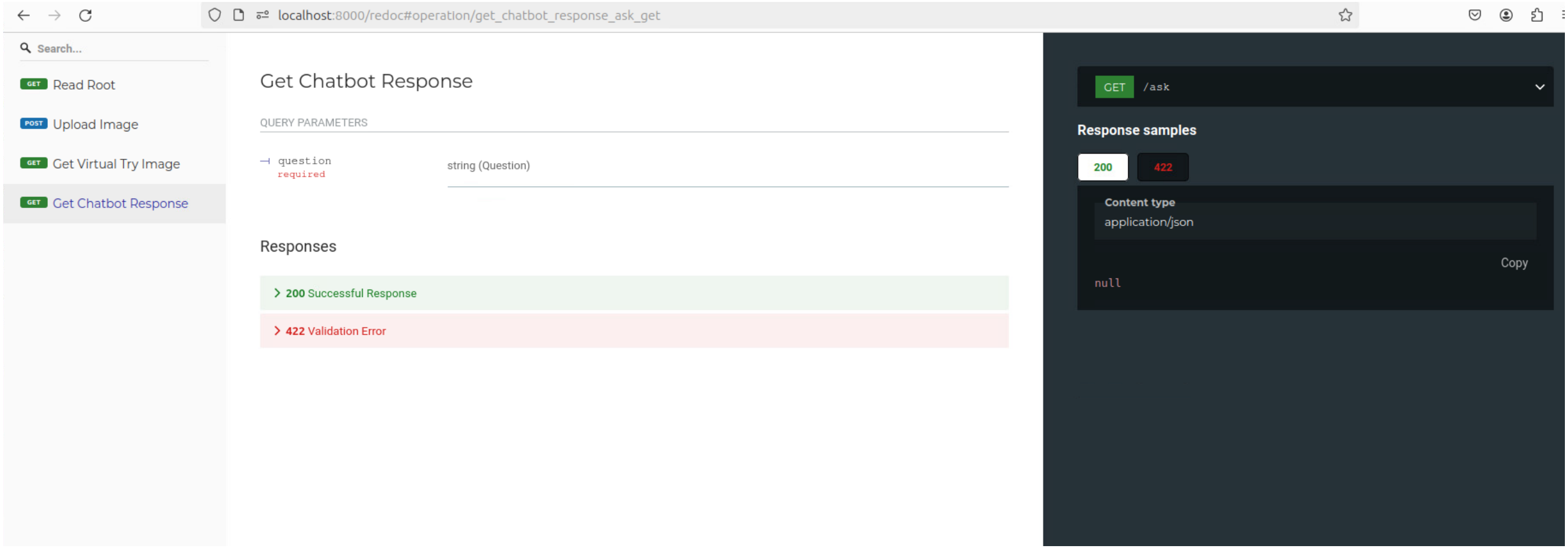
API DOCUMENT

Get Virtual Try Image API: http://localhost:8000/redoc#operation/get_virtual_try_item_id_get



API DOCUMENT

Get Chatbot Response API: http://localhost:8000/redoc#operation/get_chatbot_response_ask_get



EXTENT OF SCALABILITY

EXTENT OF SCALABILITY

Our application leverages Docker containers for easy scaling. Spinning up new containers to handle increased service capacity takes mere seconds. Additionally, Kubernetes can be integrated for automatic scaling of these Docker containers across multiple hosts, providing horizontal scaling for the application.

Furthermore, our initial use of JSON data files allows for a smooth transition to a NoSQL database like MongoDB. By creating multiple replicas of the database, we can achieve horizontal scaling, ensuring the system can accommodate a growing user base.

IMPACT

IMPACT

The shopping assistant offers a multitude of benefits for both users and retailers:

Enhanced Shopping Experience: Users can enjoy a more convenient and personalized shopping experience.

Increased Safety: By enabling virtual try-on and reducing the need for physical store visits, the platform promotes safety during pandemics or for those seeking a contactless shopping experience.

Boosted Platform Engagement: The interactive features of the shopping assistant can lead to increased user engagement on the platform.

Reduced Costs: Both retailers and customers can save money by minimizing returns and exchanges facilitated by the virtual try-on functionality.

Eliminated Infrastructure Expenses: Retailers can avoid the significant costs associated with setting up physical trial rooms or showrooms.
Time-Saving Efficiency: The shopping assistant allows users to shop efficiently, saving valuable time.
Increased Sales: By streamlining the shopping experience and offering a more engaging platform, the shopping assistant can contribute to increased sales for retailers.

Future Enhancements

Future Enhancements

The shopping assistant has the potential for further development in several areas:

Improved Chatbot Responses: Integration of generative models can enhance the chatbot's ability to provide more precise and informative responses.

Natural Language Interaction: Speech-to-text and voice synthesis functionalities can be incorporated to create a more natural and interactive user experience.

Mobile App Development: Expanding accessibility through a mobile application that communicates with the existing server.

Automated Review Summarization: Utilizing web workers to run the product review summarization script at regular intervals, ensuring up-to-date summaries.

Multilingual Support: Catering to a wider audience by enabling the chatbot to interact in multiple languages.

Reduced Latency in Virtual Try-On: Leveraging GPUs can potentially reduce the processing time for virtual try-on, leading to a smoother user experience.

THANH YOU !!!