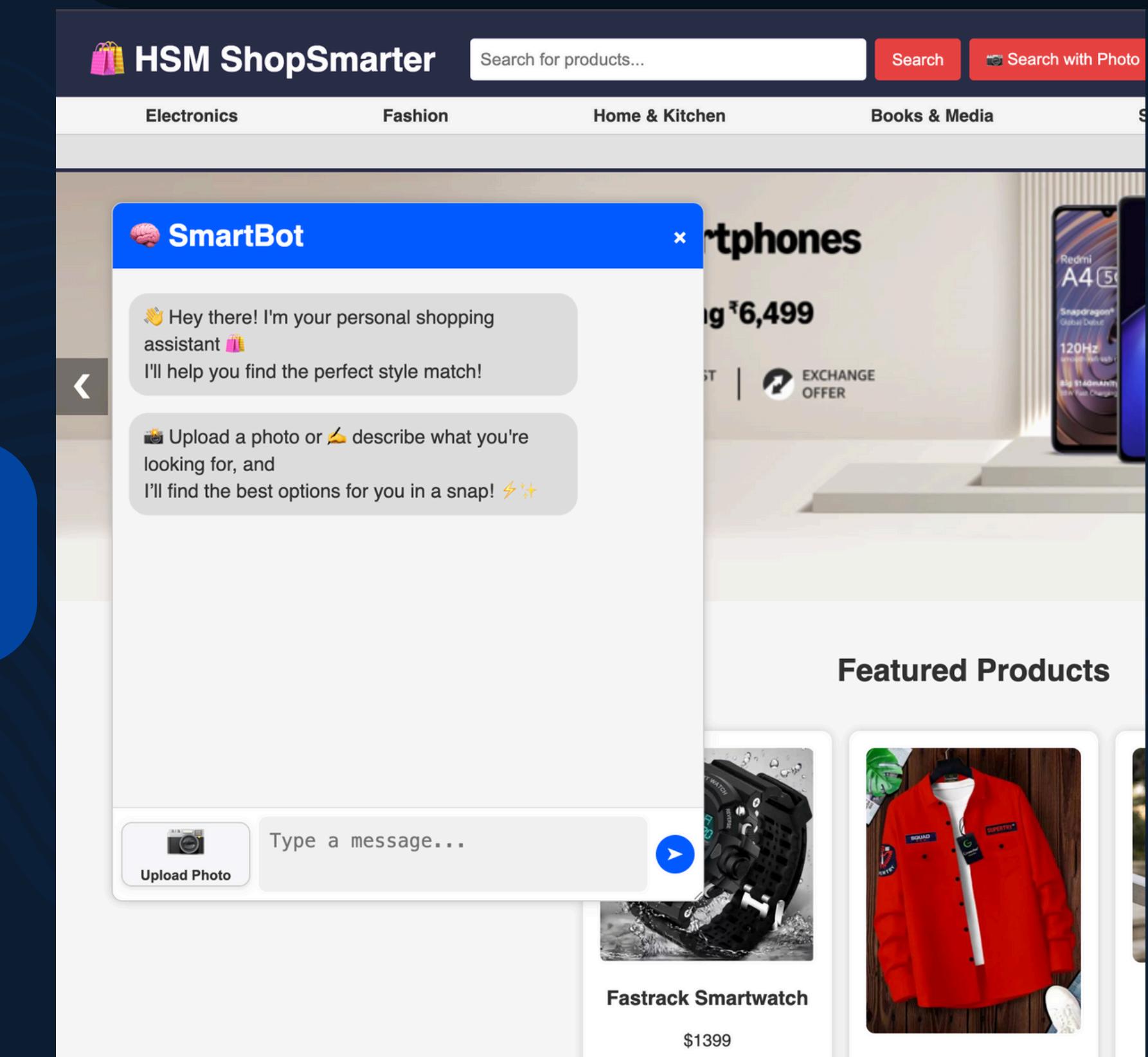


# AIgnite

Team: HSM

Link to repo: [https://github.com/Hemanth-2706/Appian\\_Hackathon](https://github.com/Hemanth-2706/Appian_Hackathon)

appian X AI CLUB IITM



# PROBLEM STATEMENT:

## **"ShopSmarter: AI-Powered Personal Shopping Assistant for E-Commerce"**

**Despite the growth of online shopping, users still struggle to:**

### **1. Describing Exactly What the User Wants**

Online shoppers often struggle to express their needs using precise keywords. Visual preferences like color tone, texture, or fit are hard to verbalize, especially for fashion and decor. This leads to vague or incorrect search results, creating frustration and making product discovery inefficient and time-consuming.

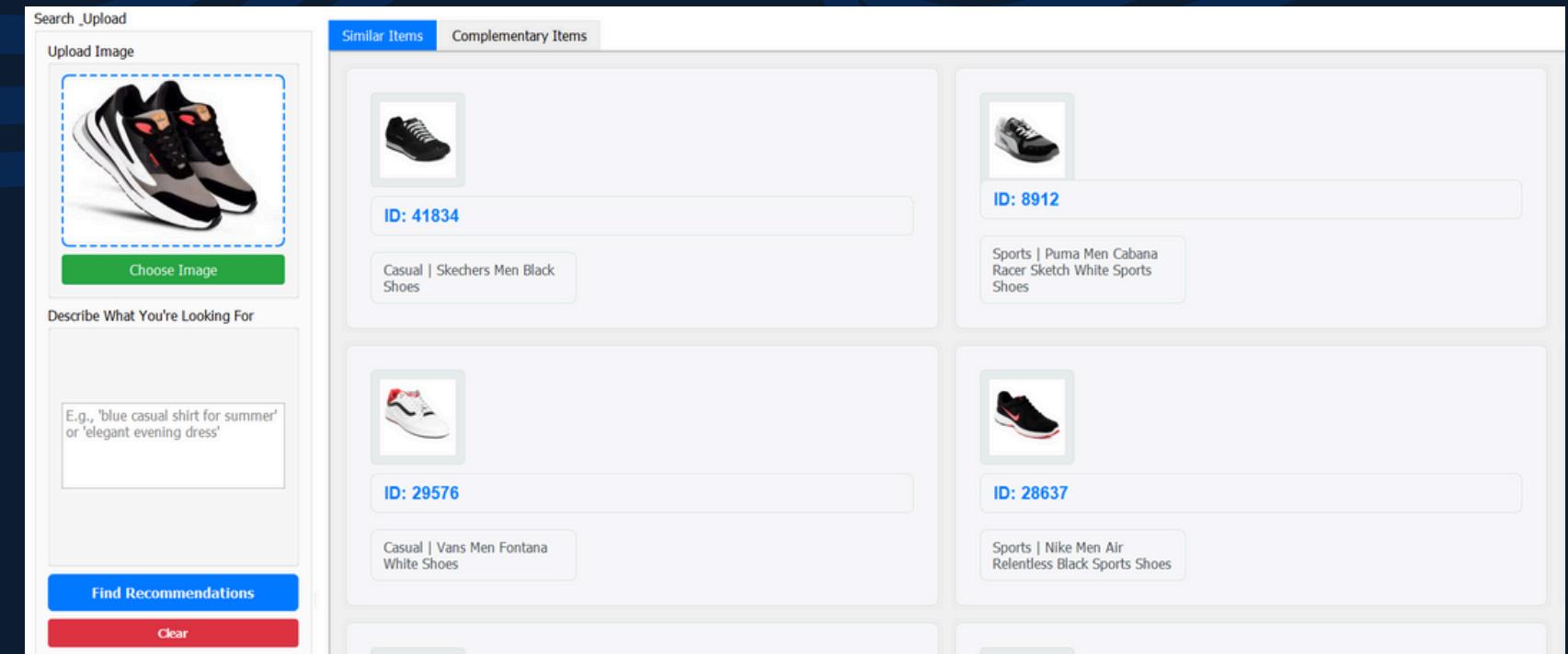
### **2. Finding Visually Similar or Complementary Products**

Customers frequently see an item they like—on social media or in real life—but can't easily find similar or matching products online. Current systems lack deep visual understanding, making it difficult to suggest aesthetically coherent or complementary items, which limits inspiration-based shopping and decreases overall satisfaction with e-commerce platforms.

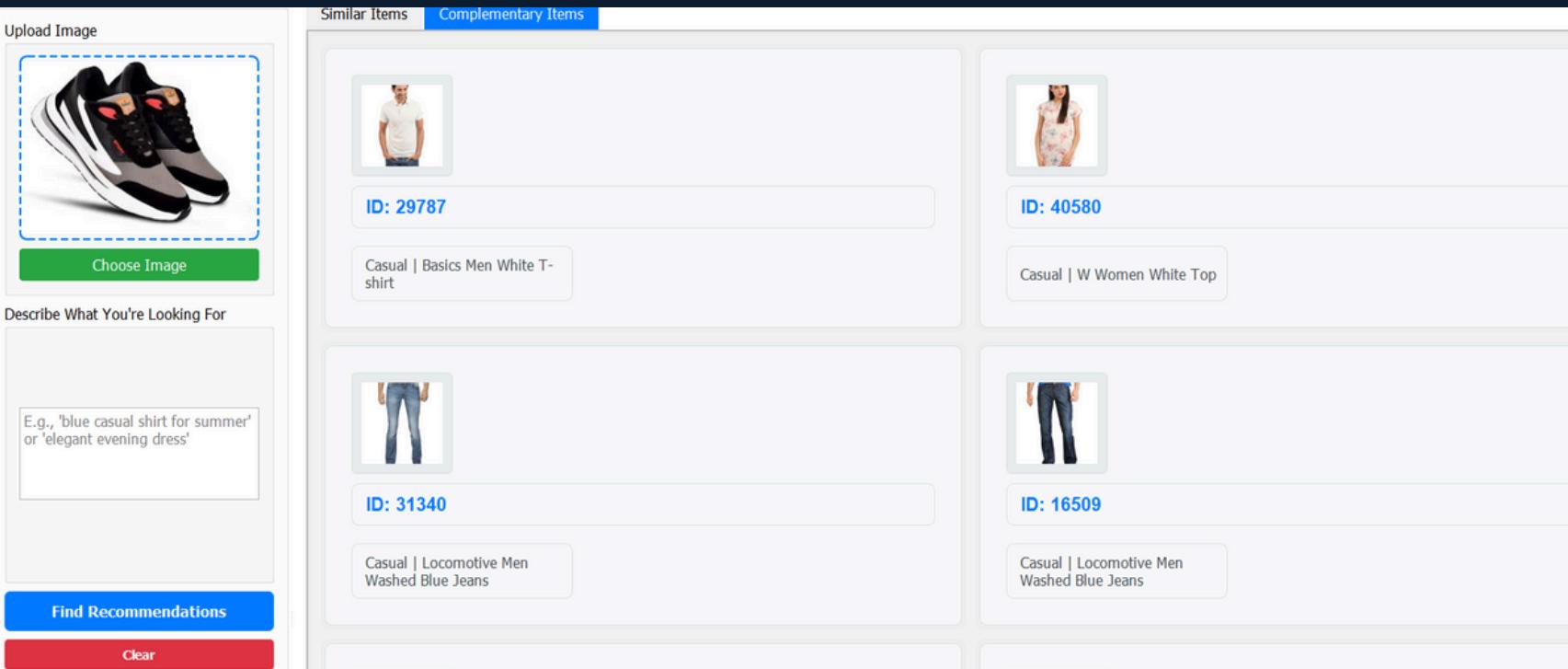
# SOLUTION:

## An AI-powered assistant that enhances the e-commerce experience by:

- Accepting image uploads (e.g., apparel, gadgets).
- Identifying key visual features (color, texture, category, etc.).
- Recommending similar or complementary items available on the e-commerce platform.
- Provide an interactive and user-friendly interface that enables easy uploads, chat-based refinement, and effortless product exploration
- Supporting interactive user prompts (e.g., "show me similar jackets").
- Delivering a seamless and intuitive shopping experience.

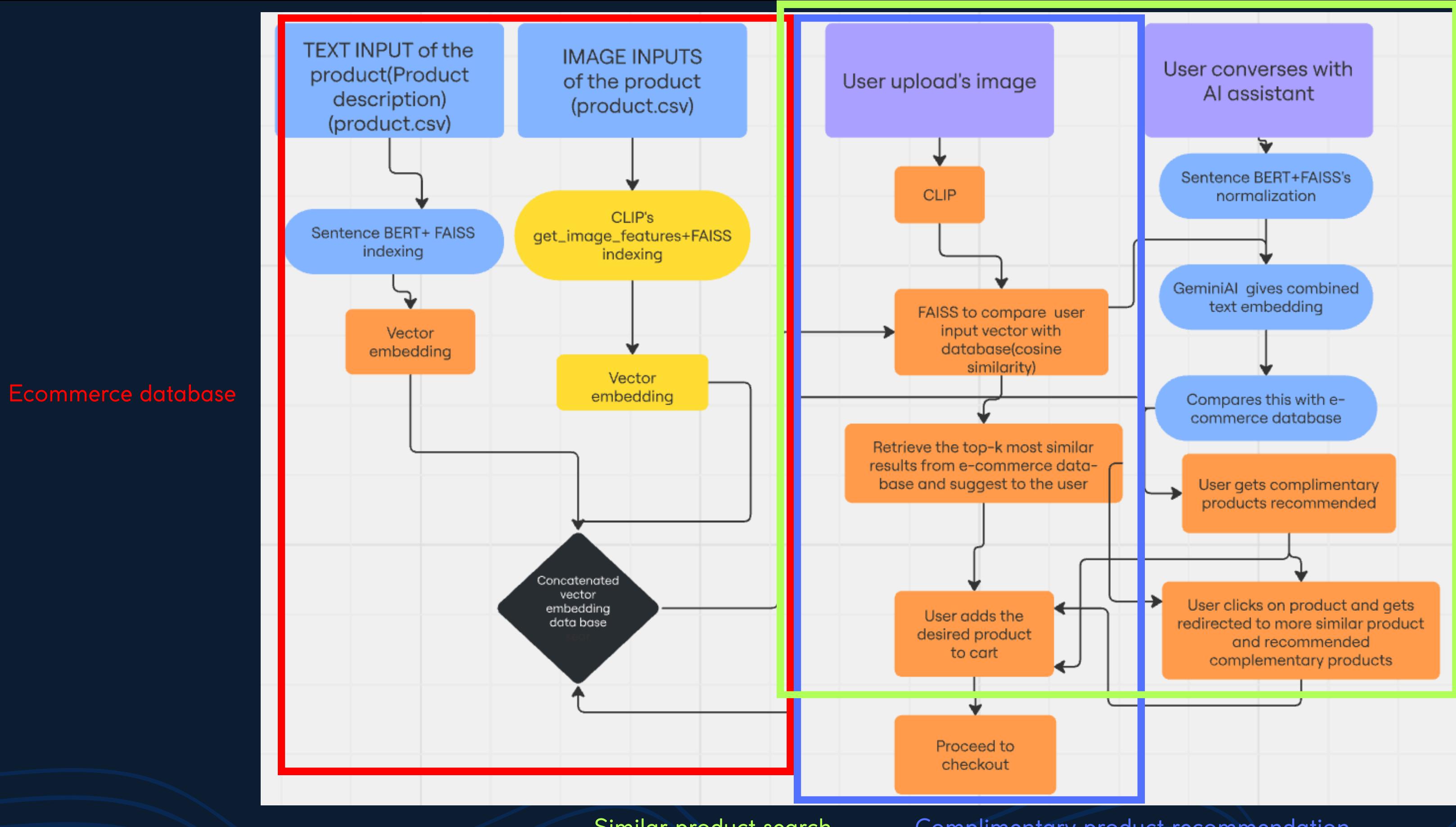


## Similar products

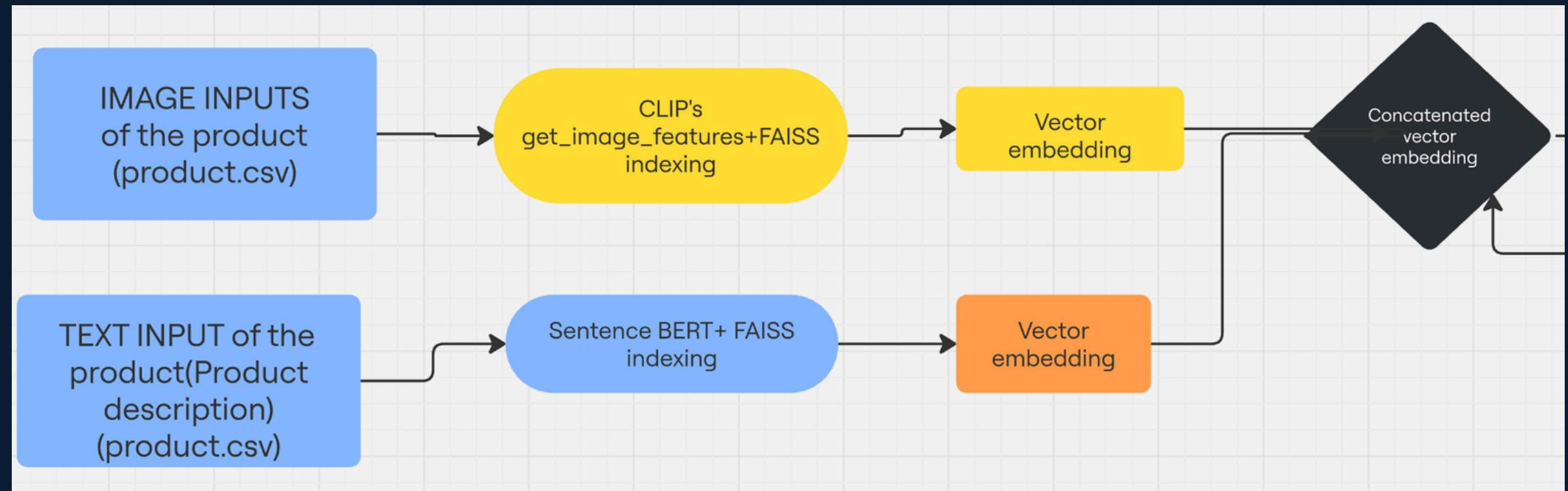


## Recommended products

# APP FLOW:



# E-COMMERCE DATA BASE GENERATION



- Extract image and text embeddings using text language models **CLIP** and **Sentence-BERT** from the product image and text info of product. (We combine them to form a detailed query.)
- Normalize and store all embeddings in a **FAISS** indexing.
- Storing multi modal data as **vector embedding** is by far the most efficient method in NLP and CV.
- Later the AI model will find similar or complementary products by correlating the input vector with vector in our database.
- The **vector** would encode various features such as category, color, texture, size, age, etc.
- Our data-base is dynamic, if we add a new product to our database, the vector embedding will be auto-generated using the text info of product and product image.

# THE SIMILAR PRODUCT SEARCH

Welcome    Fashion Recommendations

Search \_Upload

Similar Items    Complementary Items

Upload Image



Choose Image

Describe What You're Looking For

how me some shoes that would go well with this jersey

Find Recommendations    Clear

Chat with your AI Stylist

Type your message here...    Send

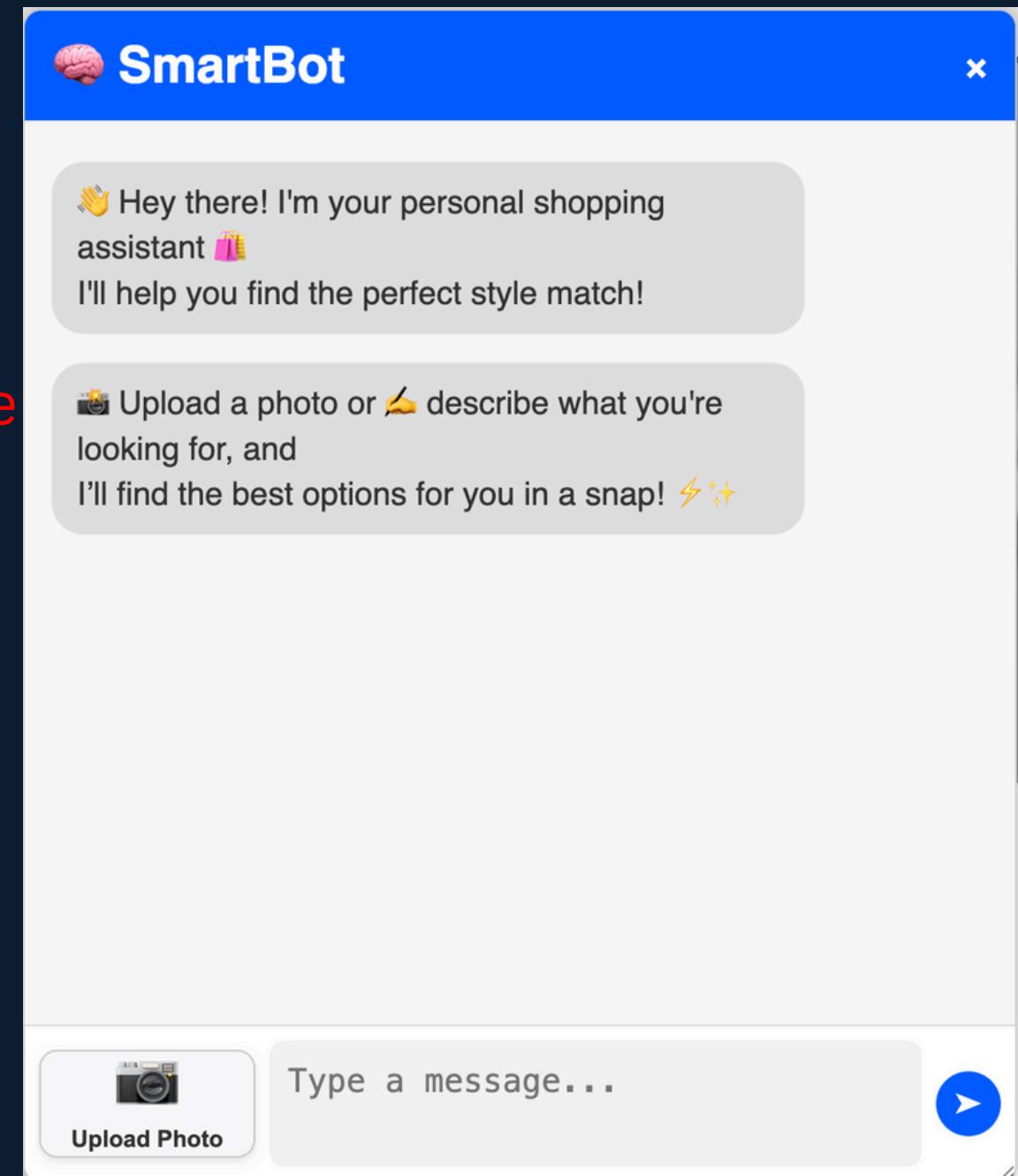
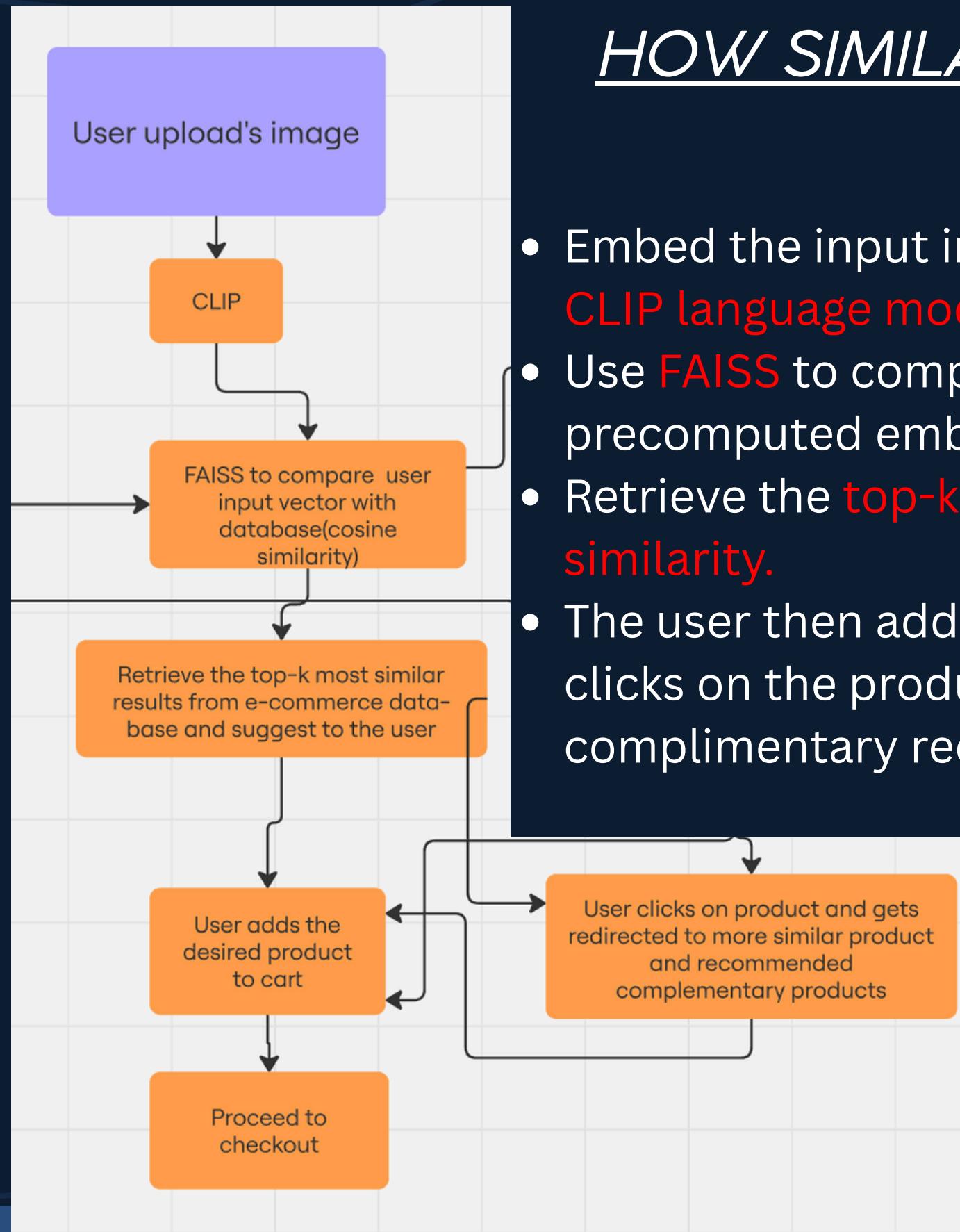
Similar Items Grid:

-  ID: 15349  
Sports | Nike Men Blue Team India Cricket Jersey
-  ID: 3731  
Sports | CSK Mens Premium Jersey
-  ID: 4124  
Sports | Kochi Tuskers Kids Replica Jersey
-  ID: 14001  
Sports | Nike Men Solid Navy Blue Jerseys
-  ID: 2274  
Sports | Nike Mens AS White T-shirt
-  ID: 10602  
Sports | Nike Men Football Soccer Red Jerseys
-  ID: 4125  
Sports | Lotto Men Kochi Tuskers Authentic Orange Jersey

accurate results!!

# HOW SIMILAR PRODUCT SEARCH WORKS?

- Embed the input image or text given by the user using **CLIP language model** to obtain a normalized vector
- Use **FAISS** to compare this vector with a database of precomputed embeddings
- Retrieve the **top-k most similar results based on cosine similarity.**
- The user then adds the desired products to cart or clicks on the product to get the product specific complimentary recommendations or similar products.



# THE RECOMMENDER SYSTEM

AI Fashion Stylist

Welcome Fashion Recommendations

Search Upload

Similar Items Complementary Items

Upload Image

Choose Image

Describe What You're Looking For

I want to go to a party this evening and I have this dress recommend a similar dress for my friend and show me some belt bag and saddle bags

Find Recommendations Clear

Chat with your AI Stylist

Type your message here... Send

Recommendations ready

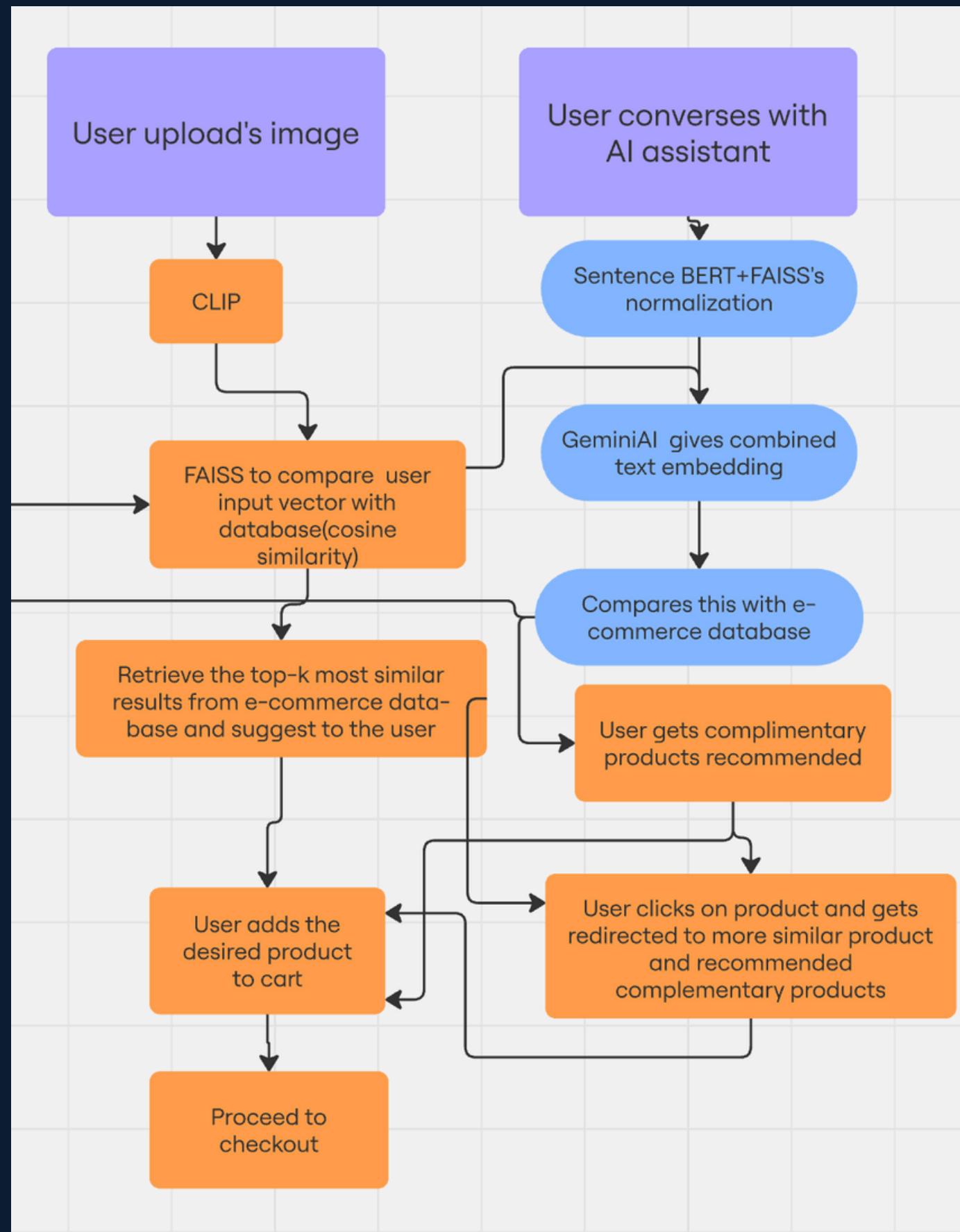
The screenshot shows a user interface for a fashion recommender system. On the left, there's a section for uploading an image and describing what you're looking for. The main area displays a grid of recommended items under the 'Similar Items' tab. Each item is shown with a thumbnail, ID, and a brief description. The recommended items are all casual handbags, mostly brown or black, from brands like Baggit, Mod'Acc, Murda, and Murcia. Below the grid, there's a chat interface where the user can interact with the AI stylist.

ID	Category	Item Description
47368	Casual   Baggit Brown Handbag	Brown handbag
52647	Casual   Mod'Acc Women Black Handbag	Black handbag
33443	Casual   Baggit Women Brown Handbag	Brown handbag
47359	Casual   Baggit Women Brown Handbag	Brown handbag
27678	Casual   Murda Women Brown Handbag	Brown handbag
27679	Casual   Murcia Women Brown Handbag	Brown handbag
47732	Casual   Murcia Women Brown Handbag	Brown handbag
43333	Casual   Murcia Women Brown Handbag	Brown handbag
47794	Casual   Murcia Women Brown Handbag	Brown handbag

A detailed prompt and accurate results!!

Our very own HSM fashion stylist!!

# HOW RECOMMENDATION SEARCH WORKS ?



- Generate a caption from the uploaded image using a vision-language model like **BLIP**.
- Combine this caption with the user's prompt to form a detailed query.
- Feed the combined context into a **chat model** acting as a **personalized stylist**.
- It returns 5 structured complementary item descriptions based on the input product and user intent.
- Retrieve the top-k most similar results based on **cosine similarity**
- Convert each recommendation into text embeddings using **CLIP's text encoder**.
- Use FAISS to find the most similar products in your pre-embedded product database.
- **Merge retrieved items, remove duplicates, and rank them by similarity score**.
- Return the **top-N complementary items** to the user

# THE E-COMMERCE PLATFORM

HSM ShopSmarter

Electronics Fashion Home & Kitchen Books & Media Sports & Outdoors Healthcare Car & Motorbike

Discover Your Style • Personalized Recommendations • AI Shopping Assistant

Starting ₹149 Household, cooking needs & more

TOP BRANDS GREAT PRICES

SmartBot

Hey there! I'm your personal shopping assistant! I'll help you find the perfect style match!

Upload a photo or describe what you're looking for, and I'll find the best options for you in a snap!

SmartBot

Type a message...

Upload Photo

Featured Products

Fastrack Smartwatch \$1399 Add to Cart

Men Regular Fit Self Design Built-up Collar \$1499 Add to Cart

Shoes \$1499 Add to Cart

USER FRIENDLY

Fastrack Smartwatch

Category: Electronics  
Price: \$1399  
Details: Stylish and functional

Quantity: 1 Add to Cart

Similar Products

Premium Sports Watch

Advanced fitness tracking and style  
\$1599  
Qty: 1 Add to Cart

Products We Recommend

Stylish T-Shirt  
Comfortable casual wear

Classic Jeans  
Perfect fit denim

Casual Sneakers  
Everyday comfort and style

# THE WEB APP- FRONT END- HSM SHOPSMARTER

## 1. Languages n structure :

- **HTML (via EJS template)**-Dynamic HTML generation
- Plain HTML is static—what you write is what gets served. You can't insert data dynamically unless you modify the HTML using JavaScript in the browser or generate it server-side. EJS, on the other hand, allows you to embed JavaScript logic directly in your HTML file.
- **Java-script**- client side- static assets
- **CSS**- styling
- Clear separation and clean documented code, between presentation and logic.

## 2. Novelty-

- Uses EJS for server-side rendering, blending dynamic content with static HTML for better SEO and user experience. Avoids heavy frameworks (like Streamlit, PyQt), resulting in a lightweight, web-native solution.
- Real, user-friendly UI instead of easy MVP Streamlit.

## 3. Expandability-

- Easily integrates new UI components or third-party libraries.
- Can be upgraded to modern frameworks (React, Vue) if needed.
- Supports modular addition of features.

# THE BACKEND AND STRUCTURE

The image shows two side-by-side code editors, likely from Visual Studio Code, illustrating the structure and code of the ShopSmarter application.

**Left Editor (Client Side):**

- Explorer:** Shows the project structure under "SHOPSMARTER - Smitali".
- File:** "home.ejs" is open, displaying the HTML template for the home page. It includes a header section with a logo, search bar, and login button, followed by a sticky container containing a category navbar and a list of electronics items.
- Bottom Status Bar:** Shows file statistics: 37 lines, 2 spaces, 60 columns, and encoding: UTF-8. It also includes icons for Connect, Go Live, Prettier, and a notification bell.

**Right Editor (Server Side):**

- Explorer:** Shows the project structure under "SHOPSMARTER".
- File:** "mockRoutes.js" is open, defining routes for the application.
- File:** "recommend.js" is open, containing logic for recommending products based on user input and a model.
- Bottom Status Bar:** Shows file statistics: 35 lines, 2 spaces, 60 columns, and encoding: JavaScript. It also includes icons for Connect, Go Live, Prettier, and a notification bell.

# THE WEB APP- BACK END

## 1. Languages n structure :

- **JavaScript(Node.js)**
- Express.js routes and controllers in the server/ directory.
- Middleware functions for request processing.
- Separation of concerns between routing, logic, and data handling.

## 2. Novelty-

- Built on Node.js, enabling **non-blocking, event-driven** architecture.
- Uses **Express.js for rapid API and route development.**
- Handles multiple concurrent requests efficiently.
- Lightweight, with minimal dependencies. **Fast response times** due to asynchronous processing.

## 3. Expandability-

- Scalable to handle more users or data with minimal changes.
- Can integrate with databases, authentication, or external APIs.

# THE WEB APP- MIDDLE WARE

## 1. Languages n structure :

- **JavaScript(Node.js, Express.js)**
- Middleware functions defined and used in the server/ directory.
- **Chained processing:** each middleware can pass control to the next, creating a **flexible pipeline.**

## 2. Novelty-

- Uses Express.js middleware for **flexible request/response handling.**
- Custom middleware can be added for logging, authentication, error handling, etc.
- Streamlines request processing by handling tasks (auth, parsing, etc.) before reaching route handlers.

## 3. Expandability-

- Supports **easy addition or removal of third-party middleware** for features like security, sessions, or analytics.

# FUTURE PLANS

## 1. Smarter Recommendations through Behavioral Embedding

We plan to achieve this by leveraging the embedding vectors of products the user has previously purchased. By calculating the proximity of new products to these vectors, we can recommend items that closely match their preferences. Additionally, we will perform a comprehensive behavioral analysis of the user's purchase history to further enhance the relevance of future suggestions. We will do this by finding the user purchase behavioral vector in the embedding vector space based on purchase history. We will classify user purchase behavior into temporary behavior and long term character behavior and fine tune our chat bot parameters and recommendation system accordingly.

## 2. Instant One-Click Checkout:

We are building a fully automated and fast checkout system that lets users complete purchases with just one click. By using secure payment methods, saved user details, and real-time product availability checks, we aim to remove manual steps and offer a smooth and quick buying experience.

# THANK YOU

Presented by:

Hemanth M ([ee24b024@mail.iitm.ac.in](mailto:ee24b024@mail.iitm.ac.in))

Mothish([cs24b033@mail.iitm.ac.in](mailto:cs24b033@mail.iitm.ac.in))

Shaurya Pavan([mm24b048@mail.iitm.ac.in](mailto:mm24b048@mail.iitm.ac.in))

Smitali Bhandari ([ce24b119@mail.iitm.ac.in](mailto:ce24b119@mail.iitm.ac.in))