Çankaya University

Computer Engineering Department
CENG 407 - Software Requirements Specification (SRS)

Multimodal RAG-Based Product Recommendation System

Hazal KANTAR - 202111036
Ahmet Doğukan GÜNDEMİR - 202111033
Ali Boran BEKTAŞ - 202111001
Hikmet Berkin BULUT - 202111057

Contents

1 Introduction

1.1 Purpose of This Document

The purpose of this document is to outline the software requirements for the development of a Multimodal Retrieval-Augmented Generation (RAG)-Based Product Recommendation System. This document serves as a reference for stakeholders, including project managers, developers, designers, and testers, ensuring a shared understanding of the project's goals, features, constraints, and technical requirements. It provides a comprehensive framework for system implementation, verification, and maintenance.

1.2 Scope of This Project

This project focuses on designing an intelligent recommendation platform for the fashion and cosmetics domains, using multimodal data sources such as textual descriptions, product images, sustainability certifications, and trend analytics. By utilizing advanced AI technologies, including LLMs and multimodal embeddings, the system provides eco-conscious and trend-aligned product recommendations tailored to individual user preferences. The platform consists of a Flask-based backend, a React-powered frontend, and a vector database for efficient data handling. Its ultimate goal is to promote sustainable consumer choices through a user-friendly, adaptive, and visually engaging interface.

2 General Description

2.1 Glossary (Definitions, Acronyms, and Abbreviations)

- **AI:** Artificial Intelligence.
- **API:** Application Programming Interface, a connection between computers or between computer programs.
- **CLIP:** Contrastive Language–Image Pretraining, a model that creates unified embeddings for text and images.[3]
- **Flask:** A lightweight Python web framework for backend API development.[4]
- **Frontend:** The part of the application that interacts directly with the user.
- **Backend:** The part of an application that is not directly accessed by the user, typically responsible for storing and manipulating data.
- **Gemini:** A suite of AI models developed by Google.[5]
- **Hugging Face:** A company providing natural language processing tools and models.[6]

- **HCI:** Human-Computer Interaction; the study of how people interact with computers and to design technologies that let humans interact with computers in novel ways.
- **LLM:** Large Language Model, a type of computational model designed for natural language processing tasks.
- **LLaMA:** Large Language Model Meta AI; a family of large language models developed by Meta AI.[7]
- **OpenAI:** An AI research and deployment company.
- **RAG:** Retrieval-Augmented Generation, A framework combining retrieval-based methods with generative models for improved contextual output.
- **React:** A JavaScript library for building user interfaces.
- **Vector Database:** A database optimized for storing and retrieving vector embeddings.

2.2 User Characteristics

The primary users of this system are environmentally conscious consumers seeking personalized recommendations for fashion and cosmetic products. They are comfortable with digital platforms and expect seamless, intuitive interactions. Secondary users include fashion retailers and brands interested in promoting sustainable products to a targeted audience.

2.3 Overview of Functional Requirements

The system involves a variety of functionalities to provide a seamless user experience and effective personalized recommendations. It includes user login and registration, enabling secure access to the platform while storing individual preferences and interaction history. Personalized and interactive recommendation generation lies at the core of the system, using multimodal data and advanced AI models to create suggestions tailored to the user's preferences. Additionally, users can review ongoing trends in the fashion and cosmetics domains, offering insights into popular and eco-conscious products.

Real-time recommendation updates ensure that users receive the most relevant suggestions based on current trends and new data. Historical recommendations are accessible through the platform, allowing users to revisit previous suggestions for convenience. Profile management features allow users to update their preferences, adjust their interaction history, and manage saved recommendations. Furthermore, a robust search and filtering mechanism is integrated to help users explore the product catalog more efficiently, ensuring a user-centric and dynamic recommendation platform.

2.4 General Constraints and Assumptions

Constraints

- **Data Availability:** The system's performance relies on continuous access to current and accurate product data, including images, descriptions, availability, and sustainability certifications.
- **Performance Limitations:** The efficiency and responsiveness of the recommendation engine are influenced by the capabilities of the employed AI models and the computational resources at hand, which may affect response times and recommendation quality.
- **Regulatory Compliance:** The system must adhere to data privacy regulations and ethical guidelines, ensuring responsible and transparent handling of user data.
- **Scalability:** Accommodating a growing user base and expanding product catalogs requires efficient data processing and storage solutions to maintain system performance.
- **Compatibility:** The system must ensure compatibility with a range of web browsers, including Chrome, Firefox, Safari, and Edge, to support a diverse user base.
- **Accessibility:** As an online platform, the system must provide uninterrupted access and maintain optimal functionality for users across different devices and network conditions.

Assumptions

- **User Connectivity:** It is assumed that users have reliable internet access and utilize modern web browsers to interact with the platform.
- **Accurate User Input:** The effectiveness of personalized recommendations depends on users providing precise and comprehensive preference information.
- **Retailer Collaboration:** Successful operation relies on effective collaboration with retailers to ensure the availability and accuracy of product sustainability information, which is critical for delivering environmentally conscious recommendations.
- **User Proficiency:** Users are presumed to possess a basic level of digital literacy, enabling them to navigate the platform and utilize its features effectively.
- **Stable Operating Environment:** The system is expected to operate within a stable technological environment, with minimal disruptions due to software updates or hardware failures.

3 Specific Requirements

3.1 Interface Requirements

3.1.1 User Interface

The system shall provide an intuitive and responsive user interface accessible via modern web browsers. The frontend will be developed using React to ensure dynamic content rendering and a seamless user experience. Users will interact with the platform through various components, including:

- **Dashboard:** Displays personalized recommendations, recent trends, and user activity summaries.
- **Search and Filter Panel:** Allows users to search for products and apply filters based on categories, sustainability certifications, price ranges, and popularity.
- **Product Detail View:** Provides comprehensive information about selected products, including images, descriptions, sustainability credentials, and user reviews.
- **User Profile Management:** Enables users to view and edit personal information, manage preferences, and review historical recommendations.

The interface will adhere to accessibility standards to accommodate users with varying needs, ensuring a user-friendly experience for all.

3.1.2 Hardware Interface

The system will integrate with various software components and services to deliver its functionalities:

- **User Devices:** Must support modern web browsers compatible with the React-based frontend.
- **Server Infrastructure:** The backend, developed using Flask, will be hosted on cloud servers (e.g., AWS, Google Cloud Platform, Microsoft Azure) to ensure scalability and reliability.

3.1.3 Software Interface

The system will integrate with various software components and services to deliver its functionalities:

- **Backend Services:** The Flask-based backend will handle API requests, manage business logic, and interface with the database.

- **Database:** A vector database (e.g., ChromaDB, Milvus) will store multimodal embeddings for efficient retrieval.
- **External APIs:** Integration with external services such as OpenAI API, Gemini, and Hugging Face for embedding services will enhance recommendation accuracy.
- **Authentication Services:** Implementation of secure authentication protocols to manage user login and registration.

These software interfaces will be designed to ensure seamless communication between components, maintaining data integrity and system performance.

3.1.4 Communication Interfaces

Client-Server Communication The frontend (React-based) will communicate with the backend (Flask-based) via RESTful APIs. These APIs will manage data retrieval, user requests, and recommendations. All communication will be secured using HTTPS to protect data integrity and confidentiality.

Backend-Database Interaction The backend will interact with the vector database (e.g., ChromaDB, Milvus) using database-specific APIs and libraries. These interactions will include storing, updating, and retrieving multimodal embeddings for recommendation generation.

Integration with External APIs The system will communicate with third-party services like OpenAI API, Gemini, and Hugging Face to access advanced AI models and embedding services. Calls to these APIs will be managed via asynchronous methods to optimize performance and ensure responsiveness.

User Notifications The system will include communication interfaces for delivering notifications to users, such as alerts for new trends, recommendations, or updates. Notifications will be sent through web-based mechanisms such as in-app messages or push notifications.

Error Handling and Logging Communication interfaces will include robust error-handling mechanisms to manage issues such as failed API calls, network disruptions, or database inconsistencies. A logging system will capture detailed logs of communication activities to support debugging and system maintenance.

3.2 Detailed Description of Functional Requirements

3.2.1 User Login and Registration

Name	User Login and Registration
Purpose/Description	Enables users to create accounts, securely log in, and access personalized features of the platform.
Inputs	• User-provided email, username, and password during registration. • Login credentials for authentication.
Processing	• Validation of input fields to ensure compliance with security policies (e.g., strong passwords). • Storage of user credentials using encrypted formats. • Authentication via secure protocols.
Outputs	• Confirmation of successful registration or login. • Error messages for failed attempts (e.g., incorrect credentials or duplicate accounts).
Error Handling	• Display specific errors for invalid inputs or registration conflicts. • Account recovery options in case of forgotten passwords.

3.2.2 Personalized and Interactive Recommendation Generation

Name	Personalized and Interactive Recommendation Generation
Purpose/Description	Generates personalized product recommendations based on user preferences, interactions, and historical data.
Inputs	• User preferences, profile data, and interaction history. • Multimodal product data (e.g., text, images, sustainability metrics).
Processing	• Retrieval of relevant data from the vector database using user embeddings. • Integration of results from AI models (e.g., RAG pipeline) to provide personalized recommendations. • Interactive updates based on user feedback (e.g., refining recommendations).
Outputs	• A list of products tailored to user preferences. • Real-time updates to recommendations when user preferences are adjusted.
Error Handling	• Data Gaps: Provide generic recommendations when user data is incomplete. • User Feedback: Allow users to report inaccurate or irrelevant recommendations. • Fallback recommendations in case of model or API failure.

3.2.3 Trends Review

Name	Trends Review
Purpose/Description	Allows users to review current trends in fashion and cosmetics, including eco-friendly options and popular products.
Inputs	• External Data Sources: Trend data from APIs like NewsAPI, social media platforms, and industry reports. • Internal Product Data: Details of products related to trending keywords, categories, or styles. • User Preferences: Personal interests or saved categories influencing trend presentation.
Processing	• Aggregation and filtering of trend data based on user preferences and sustainability criteria. • Visualization of trend summaries (e.g., trending product categories, seasonal highlights).
Outputs	• Trend Highlights: A list of popular trends, including associated products and descriptions. • Visual Summaries: Graphs showing the evolution and impact of trends over time.
Error Handling	• Data Source Failures: Display a fallback message with alternative suggestions. • Analysis Errors: Log issues in trend extraction and ensure retrying data processing.

3.2.4 Real-Time Recommendation Updates

Name	Real-Time Recommendation Updates
Purpose/Description	Provides users with up-to-date product recommendations that reflect the latest trends, availability, and user interactions, ensuring relevance and timeliness in suggestions.

Inputs	• User Interactions: Real-time data on user behaviors, such as clicks, views, purchases, and feedback. • Product Data: Continuous updates on product details, including availability, pricing, and sustainability certifications. • Trend Information: Current data on fashion and cosmetic trends sourced from industry reports, social media, and market analyses.
Processing	• Data Aggregation: Collect and integrate real-time inputs from various sources to form a comprehensive dataset. • Dynamic Analysis: Utilize AI algorithms to analyze aggregated data, identifying patterns and shifts in user preferences and market trends. • Recommendation Adjustment: Modify existing recommendations based on new insights, ensuring alignment with the most recent data.
Outputs	• Updated Recommendations: Present users with a refreshed list of product suggestions that mirror current trends and personal preferences. • User Notifications: Alert users to significant updates or changes in recommendations.
Error Handling	• Data Latency Management: Handle delays in data updates to ensure smooth functioning without noticeable lags. • Fallback Strategies: Revert to the most recent stable data if real-time data is unavailable or delayed. • User Feedback Integration: Monitor and address user-reported issues for continuous refinement.

3.2.5 View Historical Recommendations

Name	View Historical Recommendations
Purpose/Description	Enables users to view previously recommended products and revisit their preferences.
Inputs	• User Profile Data: Historical records of recommendations generated for the user, stored in the system database.
Processing	• Data Retrieval: Query the database to fetch historical recommendation records linked to the user profile. • Organization and Sorting: Arrange the retrieved records based on relevance, time, or user-defined criteria for easy navigation.
Outputs	• Historical Recommendations Display: A user-friendly visualization of past recommendations, categorized and sorted for clarity.
Error Handling	• Data Unavailability: Notify the user if no historical data is available and provide guidance on generating new recommendations. • Database Connectivity Issues: Display an error message and retry option if there is a temporary failure in fetching data. • Search and Filter Errors: Offer default views if user-defined filters return no results.

3.2.6 Profile Management

Name	Profile Management
Purpose/Description	Allows users to manage their personal details, preferences, and account settings.
Inputs	• User-Provided Data: Updates to personal information, preferences, and notification settings.

Processing	• Data Validation: Verify the accuracy and completeness of the new information provided by the user. • Profile Updates: Modify and save changes to user data in the database, ensuring secure storage and real-time synchronization. • Preference Integration: Adjust system settings and recommendation algorithms based on updated user preferences.
Outputs	• Updated Profile Information: Confirmation of changes made to user details and settings. • Enhanced Recommendations: Reflect updated preferences in the personalized recommendations provided by the system. • Notification Settings: Configure and display preferences for receiving updates or alerts.
Error Handling	• Input Errors: Prompt users to correct incomplete or invalid data entries. • Database Issues: Display a message if profile updates cannot be saved temporarily, and retry once the issue is resolved.

3.2.7 Search and Filter Options

Name	Search and Filter Options
Purpose/Description	Ensures users can locate items of interest by applying criteria such as categories, brands, and sustainability certification.

Inputs	• Search Query: Keywords entered by the user to locate specific products. • Filter Criteria: User-defined parameters, such as price range, brand, sustainability certifications, and trending status.
Processing	• Query Matching: Search the database for products matching the user's input keywords and filter criteria. • Filter Application: Refine search results based on selected attributes, such as relevance, price, or category.
Outputs	• Search Results: Display a list of products that match the user's search query and filter preferences. • Filter Summary: Show the applied filters alongside the results, allowing users to adjust them for refined searches.
Error Handling	• No Results Found: Notify the user if no products match the search query or filters. • Invalid Inputs: Prompt users to correct incomplete or inappropriate search queries or filter selections. • System Delays: Display a loading indicator if the search process takes longer than expected.

3.3 Non-Functional Requirements

Backend Server Constraints

The backend server must efficiently handle multiple concurrent requests to ensure smooth functionality during high-traffic periods. The database queries shall be optimized using indexing and caching techniques, minimizing data retrieval time to under 50 milliseconds per query on average. This ensures seamless interaction with real-time recommendation generation, even under significant user loads.

System Responsiveness

The system shall generate product recommendations for user queries within an average response time of 5 seconds. This responsiveness will be maintained by employing asynchronous processing and optimized API calls, ensuring user satisfaction and minimizing wait times.

Availability

The system shall maintain at least 95 percent uptime, ensuring consistent availability for users. This high availability will be achieved through the use of cloud-based infrastructure with failover mechanisms and regular monitoring of system health to preemptively address potential issues.

Ease of Use

The user interface shall be intuitive and user-friendly, requiring no more than 10 minutes for a new user to understand the main features. This will be achieved through clean design principles, tooltips, and a comprehensive help section integrated into the web platform.

Privacy

The system shall comply with GDPR standards for handling user data. It will ensure the confidentiality of personal information and allow users to access, modify, or delete their data upon request. Security measures, such as data encryption and secure login protocols, will be implemented to safeguard user information.

Compatibility

The system shall support the latest versions of major web browsers, including Chrome, Firefox, Safari, and Edge, ensuring broad accessibility. Cross-platform compatibility tests will be conducted to identify and address any discrepancies, guaranteeing a consistent user experience.

Ethical Compliance and Bias Mitigation

The recommendation algorithms shall actively monitor and minimize biases in product suggestions, especially concerning gender, ethnicity, or body type. Regular audits of the recommendation engine will be conducted to detect and rectify any unintentional biases, fostering a fair and inclusive platform.

Transparency

The system shall provide clear explanations for its recommendations, enabling users to understand the rationale behind suggested products. A "Why this recommendation?" feature will be integrated into the user interface, offering insights into how user preferences, trends, and sustainability metrics influence the suggestions.

4 ANALYSIS - UML

4.1 Use Cases

4.1.1 Use Case Diagram

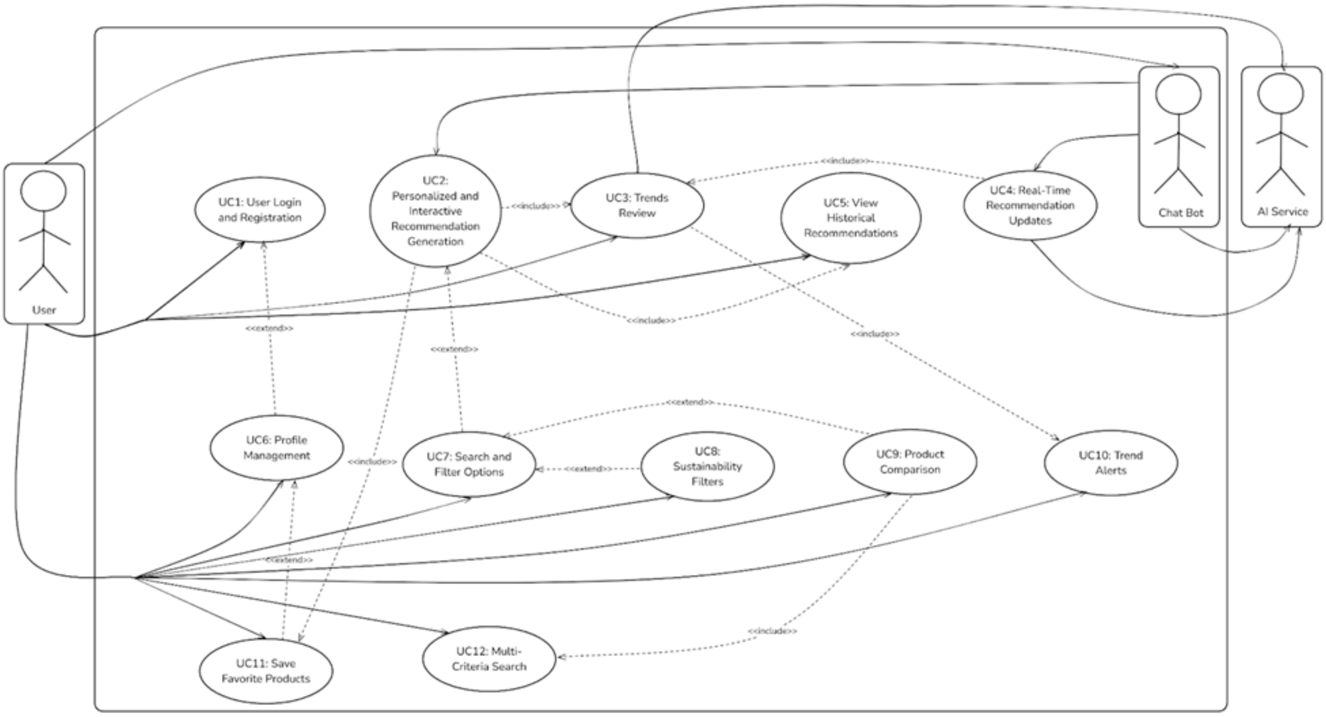
User
Chat Bot
AI Service
UC1: User Login and Registration
UC2: Personalized and Interactive Recommendation Generation
UC3: Trends Review
UC4: Real-Time Recommendation Updates
UC5: View Historical Recommendations
UC6: Profile Management
UC7: Search and Filter Options
UC8: Sustainability Filters
UC9: Product Comparison
UC10: Trend Alerts
UC11: Save Favorite Products
UC12: Multi-Criteria Search
<<include>>
<<extend>>

Figure 1: Main Use Case Diagram

4.1.2 Description of Use Cases

Use Case 1: User Login and Registration

Use Case Name	User Login and Registration
Use Case Number	1
Actors	User, System
Description	Users can create an account or log in to access personalized recommendations and manage their profile.
Precondition	The user accesses the system interface.
Scenario	1. The user navigates to the login/registration page. 2. The user enters their credentials or registers by providing required details. 3. The system authenticates the user and grants access.
Postcondition	The user is successfully logged into the system or registered as a new user.
Exceptions	1. Invalid login credentials. 2. User already exists during registration. 3. Network issues preventing communication with the backend.
Related Use Cases	Profile Management (UC6)

Use Case 2: Personalized and Interactive Recommendation Generation

Use Case Name	Personalized and Interactive Recommendation Generation
Use Case Number	2
Actors	User, System
Description	The system generates personalized recommendations based on user preferences, history, and current trends.
Precondition	The user has completed their profile and logged into the system.

Scenario	1. The user logs into their account. 2. The system analyzes user preferences and history. 3. Recommendations are displayed based on real-time trends.
Postcondition	The user receives personalized recommendations tailored to their preferences and trends.
Exceptions	1. Insufficient data in the user profile to generate meaningful recommendations. 2. System errors in retrieving or processing trend data.
Related Use Cases	Search and Filter Options (UC7), Save Favorite Products (UC11)

Use Case 3: Trends Review

Use Case Name	Trends Review
Use Case Number	3
Actors	User, System
Description	Users can view popular trends in fashion and cosmetics based on aggregated data from news sources, social media, and product sales.
Precondition	The system has up-to-date trend data available.
Scenario	1. The user accesses the trends page. 2. The system displays real-time trends. 3. The user interacts with the data for insights.
Postcondition	The user gains insights into current fashion and cosmetics trends.
Exceptions	1. Trend data is outdated or unavailable. 2. The system encounters issues while aggregating data from external APIs.
Related Use Cases	Personalized and Interactive Recommendation Generation (UC2), Real-Time Recommendation Updates (UC4)

Use Case 4: Real-Time Recommendation Updates

Use Case Name	Real-Time Recommendation Updates
Use Case Number	4
Actors	User, System
Description	The system updates recommendations dynamically as user preferences or external trend data changes.
Precondition	User preferences and trend data are updated in the system.
Scenario	1. The system monitors changes in trends or user interactions. 2. Recommendations are updated dynamically. 3. Updated recommendations are displayed to the user.
Postcondition	The user receives real-time recommendations that reflect the latest preferences and trends.
Exceptions	1. Delay in receiving updated trend data. 2. System overload due to high request volume.
Related Use Cases	Trends Review (UC3), Personalized and Interactive Recommendation Generation (UC2)

Use Case 5: View Historical Recommendations

Use Case Name	View Historical Recommendations
Use Case Number	5
Actors	User, System
Description	Users can view their past recommendations to revisit previous suggestions or track changes in trends over time.
Precondition	The system has a log of past recommendations.
Scenario	1. The user navigates to their history page. 2. The system retrieves and displays historical recommendations.
Postcondition	The user views a history of their recommendations.

Exceptions	1. Insufficient data in the history log. 2. System errors in retrieving historical data.
Related Use Cases	Personalized and Interactive Recommendation Generation (UC2)

Use Case 6: Profile Management

Use Case Name	Profile Management
Use Case Number	6
Actors	User, System
Description	Users can manage their personal details, preferences, and settings to customize their experience.
Precondition	The user is logged into their account.
Scenario	1. The user navigates to the profile section. 2. The user modifies preferences or updates personal details. 3. The system saves the updated preferences.
Postcondition	User preferences and settings are successfully updated in the system.
Exceptions	1. System fails to save the updated data due to database errors. 2. User inputs invalid data formats (e.g., non-numeric phone numbers).
Related Use Cases	User Login and Registration (UC1)

Use Case 7: Search and Filter Options

Use Case Name	Search and Filter Options
Use Case Number	7
Actors	User, System
Description	Users can search and filter products using criteria like price, color, brand, and sustainability attributes.
Precondition	The system has products and metadata for filtering.

Scenario	1. The user enters search criteria in the filter options. 2. The system retrieves products matching the criteria. 3. Results are displayed in real-time.
Postcondition	The user views products filtered by their selected criteria.
Exceptions	1. No products match the selected criteria. 2. Errors in retrieving product data due to system or database issues.
Related Use Cases	Personalized and Interactive Recommendation Generation (UC2), Product Comparison (UC9), Sustainability Filters (UC8)

Use Case 8: Sustainability Filters

Use Case Name	Sustainability Filters
Use Case Number	8
Actors	User, System
Description	Users can apply sustainability filters (e.g., eco-friendly materials, cruelty-free certifications) to refine product recommendations.
Precondition	The system contains sustainability data for the available products.
Scenario	1. The user accesses the filter options. 2. The user selects one or more sustainability criteria. 3. The system retrieves and displays filtered results.
Postcondition	Only products meeting the selected sustainability criteria are displayed to the user.
Exceptions	1. No products meet the selected criteria. 2. Incorrect filtering due to incomplete data in the database.
Related Use Cases	Search and Filter Options (UC7), Personalized and Interactive Recommendation Generation (UC2)

Use Case 9: Product Comparison

Use Case Name	Product Comparison
Use Case Number	9
Actors	User, System
Description	Users can compare multiple products based on attributes such as price, popularity, and sustainability.
Precondition	The user has added products to the comparison list.
Scenario	1. The user selects products for comparison. 2. The system retrieves the attributes of the selected products. 3. The system displays a comparison table.
Postcondition	The user views a detailed comparison of the selected products.
Exceptions	1. Insufficient product data for comparison. 2. System errors while retrieving product attributes.
Related Use Cases	Multi-Criteria Search (UC12), Search and Filter Options (UC7)

Use Case 10: Trend Alerts

Use Case Name	Trend Alerts
Use Case Number	10
Actors	User, System
Description	Users can subscribe to alerts for new trends in the fashion and cosmetics industry based on their preferences.
Precondition	The user has an active subscription and trend data is available.
Scenario	1. The user enables trend alerts in their profile settings. 2. The system monitors trend data. 3. Alerts are sent via notifications or emails.
Postcondition	The user is notified about relevant trends.

Exceptions	1. Trend data unavailable due to API errors. 2. Notifications fail to send due to connectivity issues.
Related Use Cases	Trends Review (UC3)

Use Case 11: Save Favorite Products

Use Case Name	Save Favorite Products
Use Case Number	11
Actors	User, System
Description	Users can save products to their favorites list for easy access and future reference.
Precondition	The user is logged into their account.
Scenario	1. The user clicks the favorite icon for a product. 2. The system saves the product to the user's favorites list. 3. The user accesses their favorites later.
Postcondition	The selected product is added to the user's favorites list.
Exceptions	1. Errors saving the product to the favorites list due to database issues. 2. The system fails to retrieve the favorites list when requested.
Related Use Cases	Personalized and Interactive Recommendation Generation (UC2), Profile Management (UC6)

Use Case 12: Multi-Criteria Search

Use Case Name	Multi-Criteria Search
Use Case Number	12
Actors	User, System
Description	Users can perform searches by combining multiple criteria, such as price range, brand, product category, and sustainability certifications.
Precondition	The system has a comprehensive database of product metadata and attributes.

Scenario	1. The user inputs multiple criteria in the search form. 2. The system processes the criteria and retrieves matching products. 3. Results are displayed.
Postcondition	The user views a filtered list of products matching the selected criteria.
Exceptions	1. No products match all the selected criteria. 2. Errors in data retrieval or processing.
Related Use Cases	Search and Filter Options (UC7), Product Comparison (UC9)

4.2 Functional Modeling and Data Flow Diagrams (DFD)

4.2.1 Level - 0 Data Flow Diagram (Context Diagram)

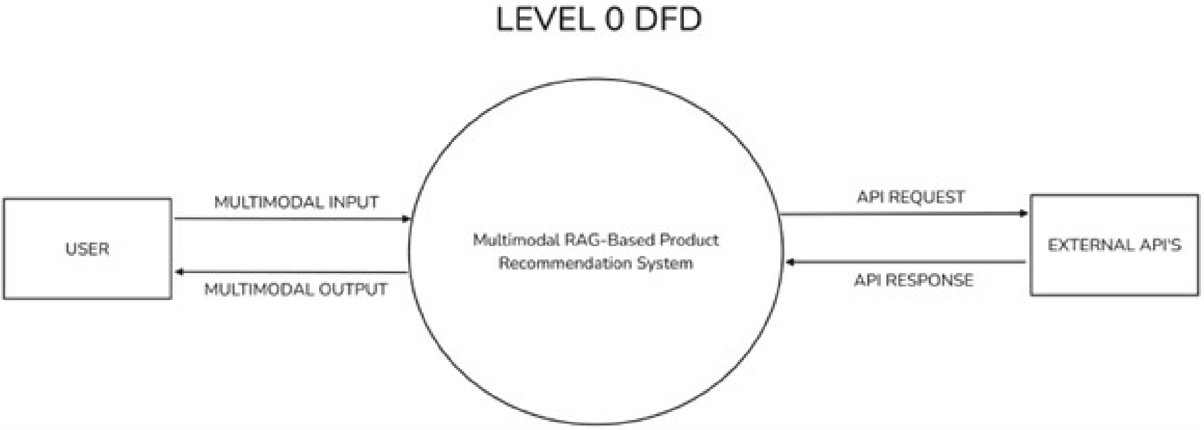

Figure 2: Level-0 Data Flow Diagram (Context Diagram)

4.2.2 Level - 1 Data Flow Diagram

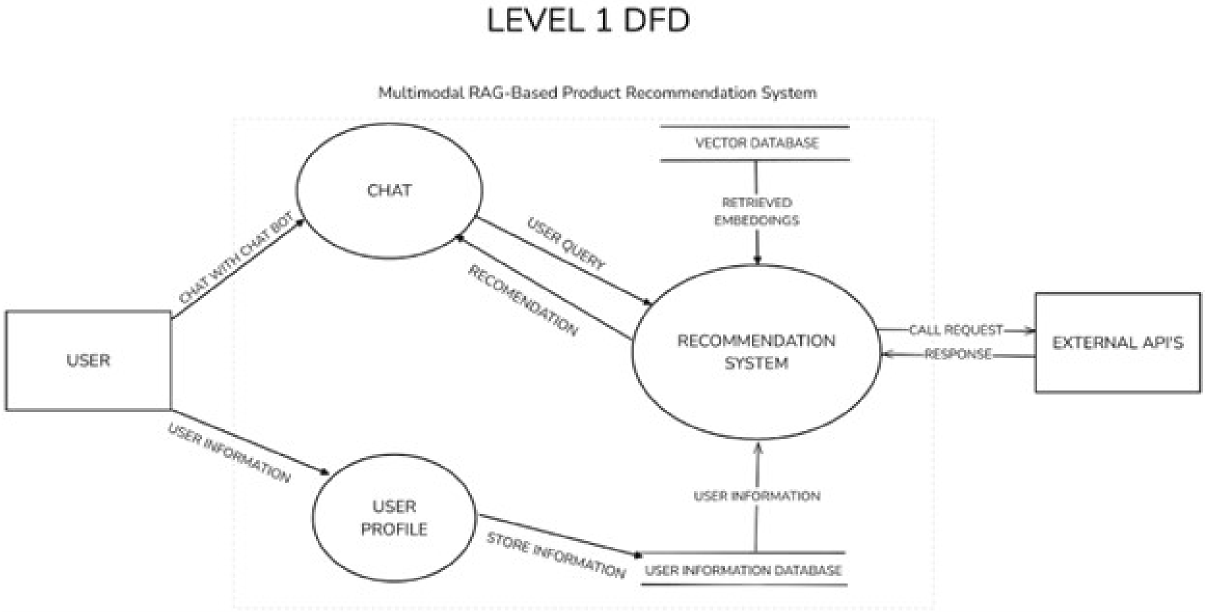

Figure 3: Level-1 Data Flow Diagram

5 CONCLUSION

The Multimodal Retrieval-Augmented Generation (RAG) system developed in this project represents a significant advancement in personalized recommendations for the fashion and cosmetics industries. By integrating textual descriptions, visual data, sustainability certifications, and real-time trend analysis, the system delivers accurate, user-centered, and environmentally conscious suggestions. Built on a scalable architecture using Flask, React, and vector databases, and powered by advanced AI services like OpenAI and Gemini, the platform ensures responsiveness and precision. This innovative system not only addresses critical gaps in traditional recommendation models but also promotes sustainable consumer behavior, setting a new benchmark for ethical and trend-aware AI applications in the industry.

References

[1] T. N. Prabhu, ”Google Trends API,” *Google Colab*, 2019. [Online]. Available: `https://colab.research.google.com/github/Tanu-N-Prabhu/Python/blob/master/Google_Trends_API.ipynb`. [Accessed: Dec. 4, 2024].

[2] Stack Overflow, ”What is the difference between functional and non-functional requirements?” *Stack Overflow*, 2013. [Online]. Available: `https://stackoverflow.com/questions/16475979/what-is-the-difference-between-functional-and-non-functional-requirements`. [Accessed: Dec. 4, 2024].

[3] OpenAI, "CLIP: Connecting Text and Images," *OpenAI*, 2021. [Online]. Available: `https://openai.com/research/clip`. [Accessed: Dec. 4, 2024].

[4] Pallets Projects, "Flask Documentation (3.1.x)," *Flask*, 2024. [Online]. Available: `https://flask.palletsprojects.com/en/stable/`. [Accessed: Dec. 4, 2024].

[5] Google AI, "Gemini Models," *Google AI for Developers*, 2024. [Online]. Available: `https://ai.google.dev/gemini-api/docs/models/gemini`. [Accessed: Dec. 4, 2024].

[6] Hugging Face, "The AI Community Building the Future," *Hugging Face*, 2024. [Online]. Available: `https://huggingface.co/`. [Accessed: Dec. 4, 2024].

[7] Meta AI, "Llama 3.2," *Meta*, 2024. [Online]. Available: `https://www.llama.com/`. [Accessed: Dec. 4, 2024].