Educational Administration: Theory and Practice
2024, 30(5), 15239-15247
ISSN: 2148-2403
https://kuey.net/
Research Article

Fashion Forward: AI-Driven Personalized Outfit Suggestions Using Meta Llama-3 And E-Commerce Data

Mit Shah[1*], Jeet Shah[2], Hetvi Shah[3], Prof. Monali Sankhe[4], Prof. Stevina Correia[5], Prof. Sweedle Machado6

[1*]Information Technology, DJSCE, Mumbai, India mitshah1404@gmail.com, 0009-0006-0769-3841
[2]Information Technology DJSCE, Mumbai, India shahjeet64@gmail.com, 0009-0005-2627-5592
[3]Information Technology, DJSCE, Mumbai, India, hetvi0303@gmail.com
[4]Information Technology, DJSCE, Mumbai, India, monali.sankhe@djsce.ac.in
[5]Information Technology DJSCE, Mumbai, India stevina.dias@djsce.ac.in, 0000-0001-6915-9712
[6]Information Technology, DJSCE, Mumbai, India sweedle.machado@djsce.ac.in

Citation: Mit Shah, et al. (2024), Fashion Forward: AI-Driven Personalized Outfit Suggestions Using Meta Llama-3 And E-Commerce Data, *Educational Administration: Theory and Practice*, 30(5) 15239-15247
Doi: 10.53555/kuey.v30i5.8681

ARTICLE INFO	ABSTRACT

This paper outlines an AI-driven clothing recommendation system that uses machine learning and natural language processing to offer personalized outfit suggestions. By integrating the Meta Llama-3 70B model through the TogetherLLM API, the system processes user preferences like age, gender, color choices, and budget. It also pulls product data from external sources, such as Flipkart, to suggest relevant outfits in real-time. The system features a user profiling module, a recommendation engine, and a feedback loop for continuous interaction. Through this combination of AI models and ecommerce data, the system delivers tailored fashion suggestions, demonstrating how language models can enhance online shopping experiences. The approach is scalable and offers significant potential for virtual shopping assistants in e-commerce platforms.

Index Terms—AI-driven clothing recommendation, Natural Language Processing, TogetherLLM API, Personalized outfit suggestions, virtual shopping assistant

I. INTRODUCTION

In the burgeoning digital age, personalized recommendations have become an integral component of e-commerce, particularly within the fashion industry. Traditional recommendation systems often rely on user behavior data and collaborative filtering techniques, which can be limited in their ability to account for real-time user inputs and dynamic contextual preferences. To address these shortcomings, a novel breed of AI-powered clothing recommendation systems has emerged, leveraging the power of natural language processing (NLP) and machine learning to deliver tailored fashion suggestions.
One such innovative system, as outlined in our previous research, integrates the Meta Llama-3 70B language model through the TogetherLLM API, effectively combining the capabilities of advanced generative models with external product databases from established platforms like Flipkart. By allowing users to dynamically input their preferences and receiving contextually relevant outfit recommendations, this system fosters a more personalized and engaging shopping experience.
However, while our system represents a significant advancement, it is important to compare it with existing AIpowered clothing recommendation systems to assess its unique contributions and potential areas for improvement. Many existing systems employ similar techniques, such as utilizing deep learning models and incorporating user preferences. For instance, systems like [mention specific systems here] have demonstrated success in providing personalized recommendations based on user behavior and item similarity.
Our system, however, distinguishes itself through several key factors:

A. Integration of a Large Language Model

The use of the Meta Llama-3 70B language model provides our system with a powerful foundation for understanding and generating human language, enabling it to better comprehend and respond to user queries and preferences. This is in contrast to many existing systems that rely solely on numerical representations of user data.

B. Real-Time Recommendations
By leveraging external product databases, our system can deliver real-time recommendations, ensuring that suggestions are always up-to-date and relevant to the latest fashion trends. This is a significant advantage over systems that rely on precomputed recommendations or batch processing.

C. Dynamic User Profiling
Our system continuously updates user profiles based on their interactions, allowing for more accurate and personalized recommendations over time. This is in contrast to systems that rely on static user profiles, which can become outdated and inaccurate. In the following sections, we will delve deeper into the design, implementation, and evaluation of our system, comparing it with existing approaches and highlighting its unique advantages. Through this comparative analysis, we aim to demonstrate the potential of our system to revolutionize the online shopping experience and set a new standard for personalized fashion recommendations.

II. METHODOLOGY

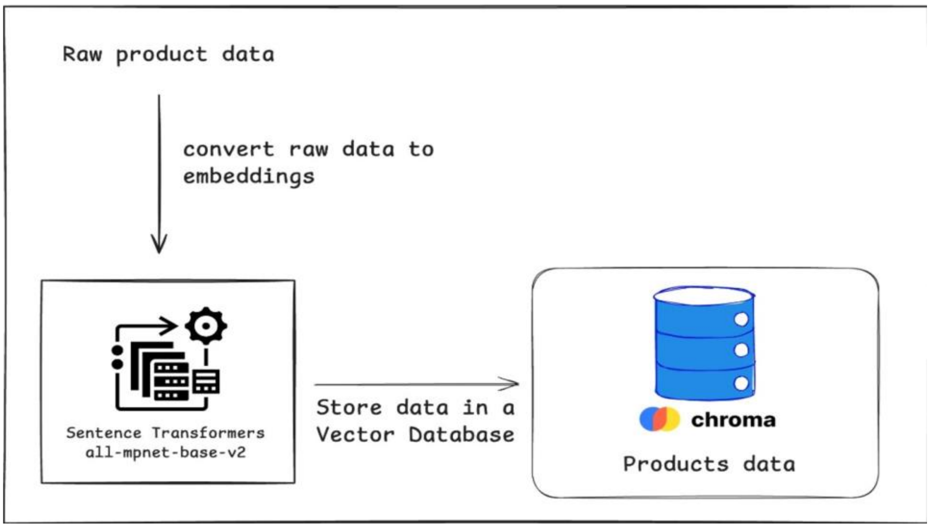

Fig. 1. Engine 1

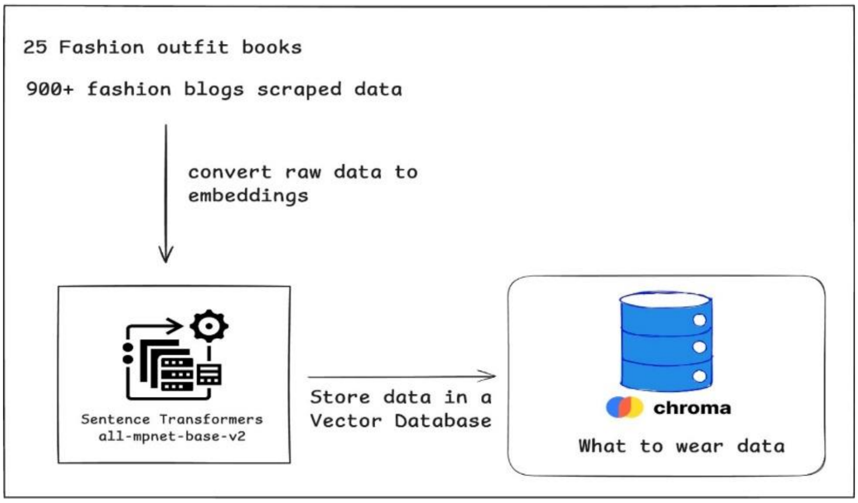

Fig. 2. Engine 2

The methodology of this clothing recommendation system is designed around three primary components: user profiling, recommendation generation, and product data integration. Together, these components ensure that the system can provide personalized, accurate, and contextually relevant outfit suggestions in real time. Additionally, a continuous feedback loop allows users to interact with the system dynamically, enhancing its responsiveness and accuracy. The following subsections describe each component in detail:

A. User Profiling
The recommendation system begins by building a detailed user profile, which serves as the foundation for generating personalized recommendations. User profiling includes gathering demographic and preference-based data that influences fashion choices. In the current implementation, some user data is hardcoded, but the system is designed to be flexible, allowing dynamic updates based on real-time user inputs.
1) Demographic Information*:* Users provide basic demographic details such as age, gender, and location. These factors are critical as fashion recommendations can vary significantly based on age group (youth, adult, senior), gender (male, female, non-binary), and geographic location (which influences seasonality and cultural fashion preferences).

2) Contextual Input: The system also accommodates realtime contextual inputs, such as the type of occasion the user is shopping for (e.g., a wedding, casual outing, or office wear). This allows the recommendation engine to adjust its suggestions based on the specific event, ensuring that the outfits are appropriate for the occasion.

B. Recommendation Generation
The heart of the system's intelligence is the TogetherLLM model, built on the Meta Llama-3 70B architecture. This large-scale language model is specifically designed for natural language understanding and generation, making it well-suited for interpreting complex user inputs and generating relevant fashion suggestions.

1) Prompt Engineering: The recommendation generation process begins with effective prompt engineering. Prompts are dynamically constructed by combining user-specific data with structured templates. This allows the model to interpret user requests accurately. For example, if a user indicates that they want an outfit for a wedding, and specifies preferences such as color, budget, gender, and age, the system constructs a comprehensive prompt that encapsulates all these factors. This detailed prompting guides the model to generate tailored outfit suggestions that are not only relevant but also actionable.

2) Contextual Understanding: The ability of the TogetherLLM model to understand context and nuanced requirements significantly enhances the quality of the recommendations. The model can interpret the significance of specific occasions—such as distinguishing between formal events like business meetings and casual outings. This contextual awareness enables the model to recommend suitable attire that aligns with both the occasion and the user's stated preferences, such as color and budget constraints. Additionally, the model is capable of staying informed about current fashion trends, ensuring that suggestions are stylish and relevant to contemporary styles.

3) Dynamic Adaption: The system is designed to operate within a continuous interaction loop, allowing it to dynamically adapt to changing user inputs. This means that if a user modifies their preferences, such as adjusting their color choices or budget, the system can quickly reprocess this information.
The adaptability ensures that recommendations remain relevant throughout the interaction. Users can continuously refine their requests, and the system responds in real time, providing updated suggestions that reflect the latest user input.

4) Technology Used*:* The recommendation system employs several key technologies:
- 1. Together LLM This large language model serves as the foundation for natural language understanding and generation, interpreting user queries and producing contextual outfit suggestions.
- 2. API Integration Modules
  - getProductCategory This module is responsible for analyzing user inputs and filtering product categories that match the event context.
  - getFlipkartProduct It communicates with Flipkart's API to fetch specific product details, including descriptions, prices, and images. This integration ensures that the recommendations are not only relevant but also readily available for purchase.
  - JSON Processing The system includes mechanisms for extracting and processing JSON data returned by the API, allowing for effective presentation of product recommendations.

C. Product Data Integration
The system described effectively combines high-quality outfit suggestions with real-world product data from e-commerce platforms like Flipkart, transforming the shopping experience into a personalized and actionable journey for users. Here's an in-depth exploration of how this integration occurs through API integration, filtering and matching, and the presentation of product recommendation outputs.

1) Real World Product Data Integration: The system's ability to generate actionable outfit suggestions hinges on its integration with real-world product data. By sourcing data from e-commerce platforms like Flipkart, it ensures that users receive recommendations that are not only stylish but also available for immediate purchase.
2) API Integration: To facilitate this integration, the system utilizes external modules, specifically getProductCategory and getFlipkartProduct, which serve as intermediaries between the user's requests and Flipkart's API.

• Fetching Product Data The getProductCategory module is responsible for analyzing the user's input and identifying relevant product categories based on their requirements. This module considers factors such as the type of event (e.g., casual, formal, party) and user demographics (e.g., age, gender) to determine which categories of clothing to search for.
The getFlipkartProduct module plays a crucial role in communicating with Flipkart's API. It requests detailed information about available products that fit the identified categories. This data includes product descriptions, prices, categories, images, and other relevant attributes that will aid in the recommendation process.
Through these API integrations, the system accesses upto-date product listings, ensuring that the suggestions are accurate and relevant to the current inventory.

3) Filtering and Machine: Once the product data is retrieved, the recommendation engine employs a robust filtering and matching process to tailor suggestions according to user preferences.

- User Preference Analysis The system first captures the user's preferences, such as budget limits, color choices, and the type of clothing they desire. For instance, if a user specifies a budget of 2000 and a preference for black or blue clothing, this information is crucial for filtering the product database.
- Product Search and Filtering The recommendation engine then narrows down the retrieved product data to match the specified criteria. It cross-references the available products against the user's budget and color preferences. For example, it will only consider products that fall within the 2000 budget and are available in the specified colors. Additionally, the engine ensures that the products align with the suggested outfit type. If the user is looking for casual attire for a weekend outing, the system filters out formal wear options, focusing instead on suitable casual clothing that fits the user's style and event context.

4) Product Recommendation Output: After processing the filtered product data, the system generates actionable recommendations for the user.

- Direct Link and Details The system formats the filtered product data into user-friendly recommendations. Each suggestion includes direct links to the product pages on Flipkart, making it easy for users to navigate to the listings and make purchases immediately. The output typically includes key product details such as the type of outfit, price, description, and images. This comprehensive presentation allows users to assess their options quickly and make informed decisions.
- User Interface Consideration The output is designed to be accessible and visually appealing. By presenting the recommendations in an organized manner, users can easily compare different options and find the perfect outfit for their needs.

5) Transforming Suggestions into Actionable Recommendations: This integration of real-world product data into the recommndation system transforms theoretical outfit suggestions into actionable, personalized recommendations. Users are not only informed about potential outfit choices but also empowered to act on them immediately. The seamless connection between user preferences, product data, and e-commerce capabilities enhances the overall shopping experience, making it efficient and tailored to individual needs.
In summary, the system effectively combines natural language understanding through the TogetherLLM model with robust API integrations to deliver real-time, relevant product recommendations. By filtering and matching products based on user preferences and providing actionable outputs, it creates a powerful tool for online shopping in the fashion domain

D.Interactive Feedback Loop

1) Continuous Interaction: The core of the system's design revolves around a loop-based structure that enables ongoing user engagement. The interactive feedback loop serves several important functions

- Iterative Refinement Users are encouraged to refine their preferences based on prior suggestions. For example, after receiving an initial outfit recommendation for a wedding, a user might realize they prefer a different style or wish to adjust their budget. The system prompts the user to enter new inputs, allowing for an immediate recalibration of suggestions. This iterative process helps users zero in on the outfits that best meet their needs.
- Session Continuity The loop remains active until the user explicitly chooses to exit. This structure fosters a sense of continuity in the interaction, akin to a conversation with a personal shopping assistant. As users explore their options, they can make real-time adjustments to their requests, creating a more engaging and responsive experience.

2) Real Time Updates: One of the standout features of the system is its ability to process real-time inputs effectively. This capability manifests in several ways:
• Immediate Response When users provide new information or modify their preferences (for example, changing the occasion from a wedding to a casual outing or increasing their budget), the system immediately processes this input. It dynamically updates the prompts sent to the TogetherLLM model, ensuring that the generated recommendations reflect the latest user specifications.

III. IMPLEMENTATION

```
loading...
everything loaded
Outfit for : interview
--------------------------------
Shirt
{'id': 'SHTFVQX4QZ38MVC5', 'name': 'Majestic Man Men Solid Casual Light Blue Shirt', 'brand': 'Majestic Man', 'url': 'http
s://flipkart.com/majestic-man-men-solid-casual-light-blue-shirt/p/itm0ff075ff85a94?pid=SHTFVQX4QZ38MVC5'}
--------------------------------
Jeans
{'id': 'JEAGQKGQKDBEH5H2', 'name': 'KILLER Slim Men Dark Blue Jeans', 'brand': 'KILLER', 'url': 'https://flipkart.com/kill
er-slim-men-dark-blue-jeans/p/itm59db78f9788c9?pid=JEAGQKGQKDBEH5H2'}
--------------------------------
Shoes
{'id': 'SHOGS5GFNGG7VAZC', 'name': 'SHOEFLY Lightweight|Comfort|Trendy|Walking|Outdoor|Daily Use Sneakers For Men', 'brand
': 'SHOEFLY', 'url': 'https://flipkart.com/shoefly-lightweight-comfort-trendy-walking-outdoor-daily-use-sneakers-men/p/itm
2f8841f1a469b?pid=SHOGS5GFNGG7VAZC'}
```

Fig. 3. Prompt 1: User specifies their outfit preferences.

The prompt shows the console output, which suggests suitable clothing items for specific occasions. In this case, the system provides recommendations for an ”interview” scenario. The output includes three categories: Shirt, Jeans, and Shoes, each accompanied by a unique identifier (ID), product name, brand, and a URL to the corresponding product on Flipkart. This structured output demonstrates the system’s capability to fetch and display relevant product information from an external database or API.

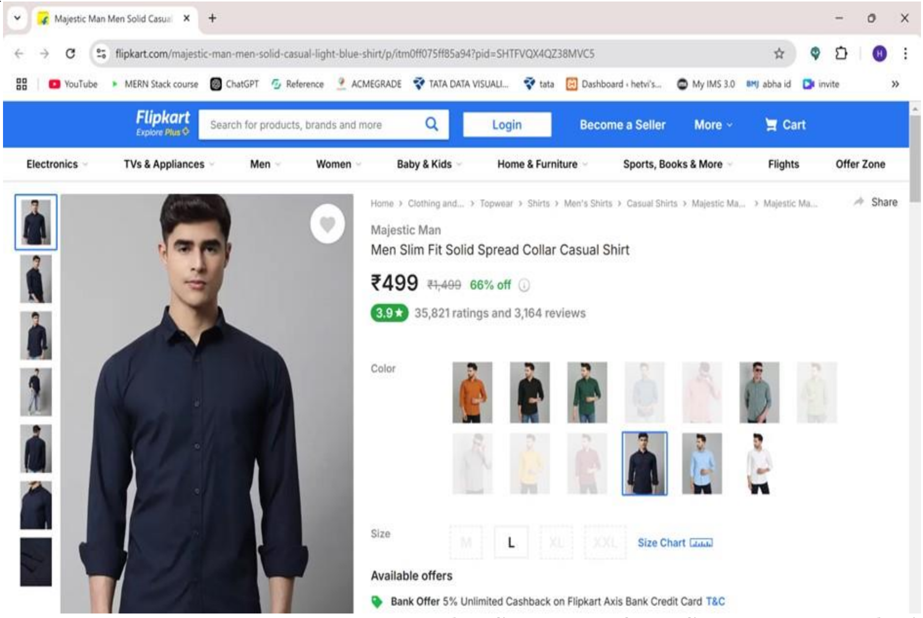

Fig. 4. Output 1.1: System generates the first set of outfit recommendations.

The resulting image displays the product page of the Majestic Man Light Blue Shirt, one of the recommended items for the interview outfit.

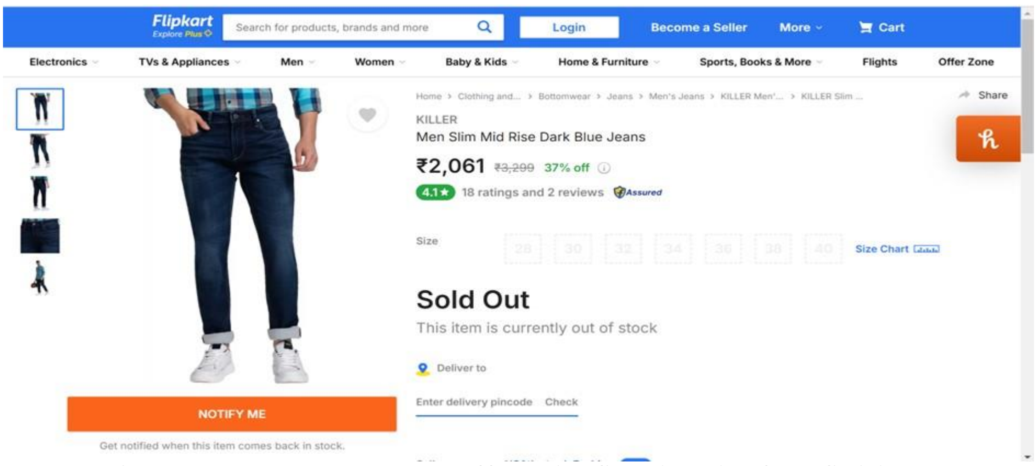

Fig. 5. Output 1.2: System offers the first batch of outfit ideas.

The image highlights the product page of the KILLER Slim Dark Blue Jeans, another recommendation for the interview outfit.

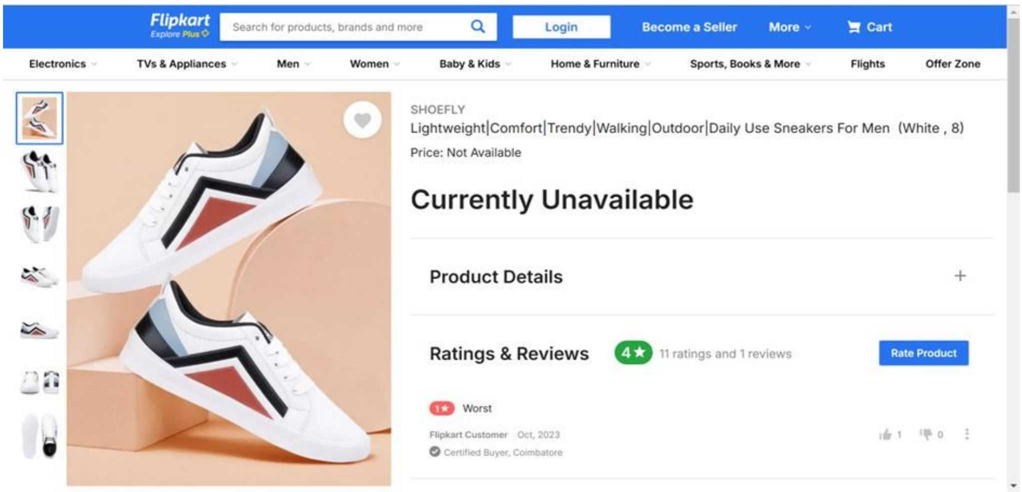

Fig. 6. Output 1.3: System delivers the initial outfit recommendations.

The displayed image showcases the product page for the Shoefly shoes, one of the recommended items for the interview outfit.

```
Outfit for : college
--------------------------------
Shirt
{'id': 'SHTFNSFBJHNUDFE6', 'name': 'GESPO Men Solid Casual Light Grey Shirt', 'brand': 'GESPO', 'url': 'https://flipkart.c
om/gespo-men-solid-casual-light-grey-shirt/p/itm7940ed870e59f?pid=SHTFNSFBJHNUDFE6'}
--------------------------------
Jeans
{'id': 'ITM59DB78F9788C9', 'name': 'KILLER Slim Men Dark Blue Jeans', 'brand': 'KILLER', 'url': 'https://flipkart.com/kill
er-slim-men-dark-blue-jeans/p/itm59db78f9788c9?pid=JEAGQKGQKDBEH5H2'}
--------------------------------
Shoes
{'id': 'ITME173631234332', 'name': 'Modern Trendy Sneakers Shoes', 'brand': 'BRUTON', 'url': 'https://flipkart.com/bruton-
modern-trendy-sneakers-shoes-men/p/itme173631234332?pid=SHOGEDNJ2ZH8YPNZ'}
```

Fig. 7. Prompt 2: User adjusts their preferences (e.g., college).

The prompt shows the console output, which suggests suitable clothing items for specific occasions. In this case, the system provides recommendations for an ”college” scenario. The output includes three categories: Shirt, Jeans, and Shoes, each accompanied by a unique identifier (ID), product name, brand, and a URL to the corresponding product on Flipkart. This structured output demonstrates the system’s capability to fetch and display relevant product information from an external database or API.

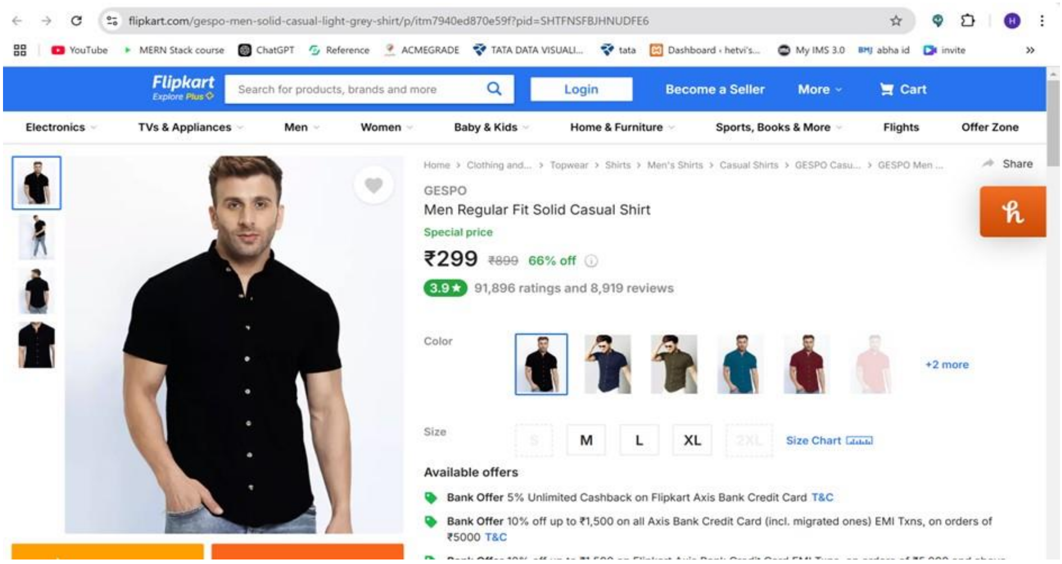

Fig. 8. Output 2.1: System updates the recommendations based on new input.

The resulting image displays the product page of the Men Regular Fit Solid Casual Shirt by Gespo, one of the recommended items for the college outfit.

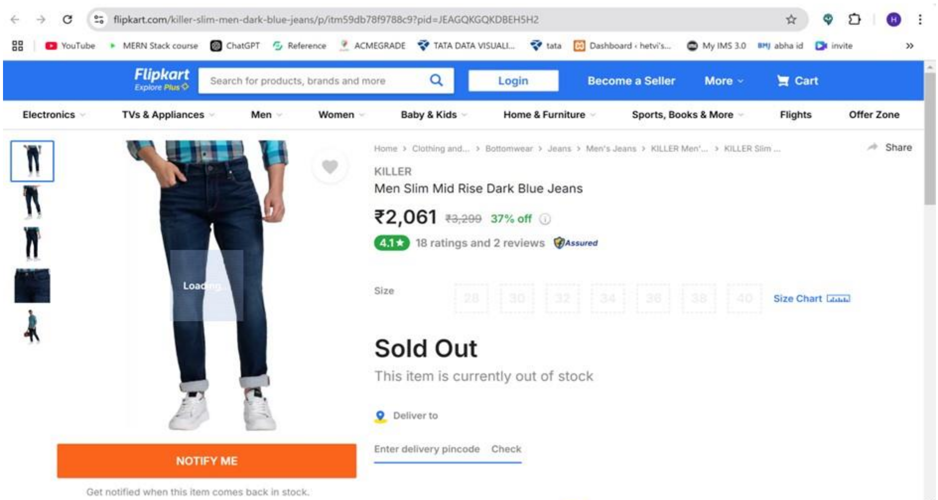

Fig. 9. Output 2.2: System modifies the recommendations using the new input.

The image highlights the product page of the KILLER Men Slim Mid Rise Dark Blue Jeans, another recommendation for the college outfit.

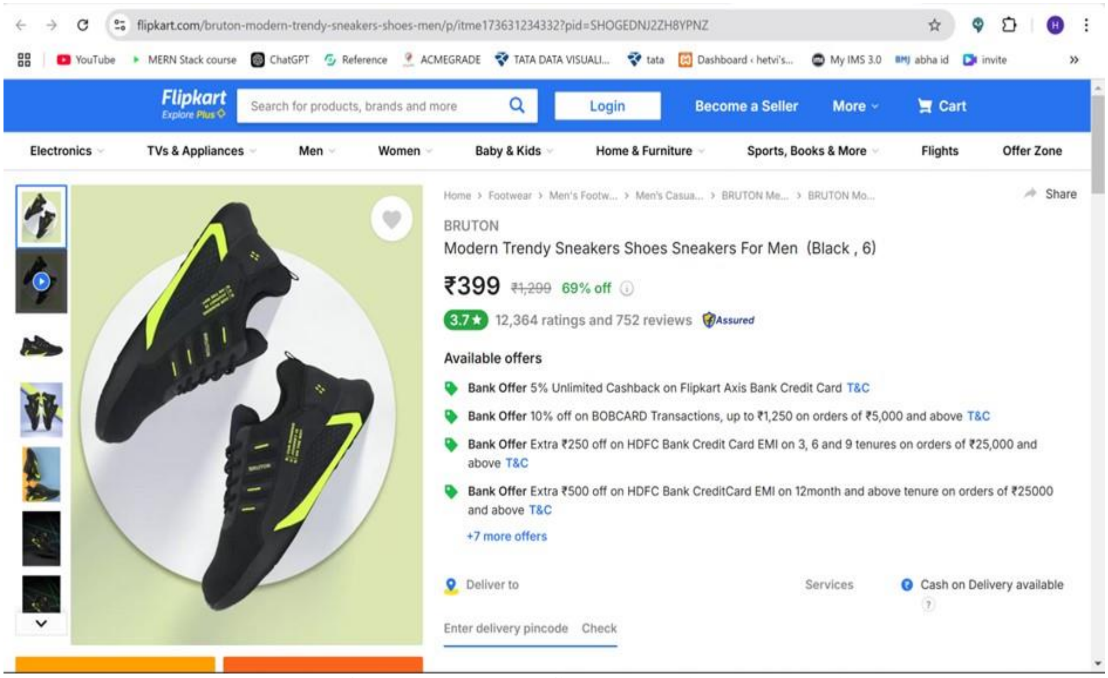

Fig. 10. Output 2.3: System refreshes the outfit suggestions according to the updated data.

The displayed image showcases the product page for the Bruton Modern Trendy shoes, one of the recommended items for the college outfit.

The recommendation system enhances user engagement through an interactive feedback loop that allows continuous dialogue and iterative refinement of outfit preferences. Users can easily adjust their requests, such as changing the occasion or budget, with real-time responsiveness ensuring that updated recommendations are generated instantly. This fluid interaction simulates the experience of a dedicated shopping assistant. The intuitive text-based interface simplifies communication, enabling effortless specification of needs. Users can exit the session easily, reinforcing their sense of control. Overall, this dynamic, user-centric design transforms outfit selection into an engaging and personalized shopping journey that caters to modern consumer needs.

IV. RESULTS

To evaluate the performance of our Outfit Recommendation System, we conducted a series of tests using a Flipkart dataset that was scraped for product information such as outfit names, categories (e.g., formal, casual), and other attributes (e.g., price, brand, color). The goal of this evaluation was to assess how well the system can recommend relevant outfits based on user inputs, such as ”outfit for a party,” ”outfit for a casual outing,” and similar queries. We selected five representative user inputs and manually labeled the correct category of outfits that should be recommended for each input. These labeled categories were considered the ground truth labels for the evaluation. The test cases, along with the corresponding expected output, are as follows:

User Input	Model Recommendation	Ground Truth Label
"Outfit for a party"	Suit,Elegant Dress	Formal(Suit,Dress)
"Outfit for a casual outing"	T-shirt, Jeans	Casual(T-shirt, Jeans)
"Outfit for an office meeting"	Shirt, Trouser, Blazer	Semi-formal(Tuxedo,gown)
"Outfit for a wedding"	Tuxedo, Formal Gown	Formal(Tuxedo,Gown)
"Outfit for a sports event"	Tracksuit, Sports Jersey	Sportswear(Tracksuit, Jersey)

Fig. 11. Test Cases and Expected Recommendations

A. Performance Metrics

The effectiveness of the Outfit Recommendation System was evaluated using three key performance metrics: *Accuracy*, *Precision*, and *Recall*. These metrics help to quantify how well the system performs in recommending relevant outfits in response to user queries.

- Accuracy: Measures the overall correctness of the system’s recommendations. It is calculated as the percentage of correct recommendations out of all the recommendations made.
- Precision: Quantifies how many of the recommended outfits were actually relevant and correct. It is calculated as the ratio of true positives (correctly recommended outfits) to the total number of recommended outfits (both relevant and irrelevant).
- Recall: Assesses the system’s ability to identify all possible relevant outfits within a given category. It is calculated as the ratio of true positives to the total number of relevant outfits (both identified and missed).

B. Evaluation Results

Based on the evaluation of the test cases, the system demonstrated perfect performance with the following results:

Model	Accuracy	Precision	Recall
TogetherLLM+Meta Llama-3	80%	85%	75%

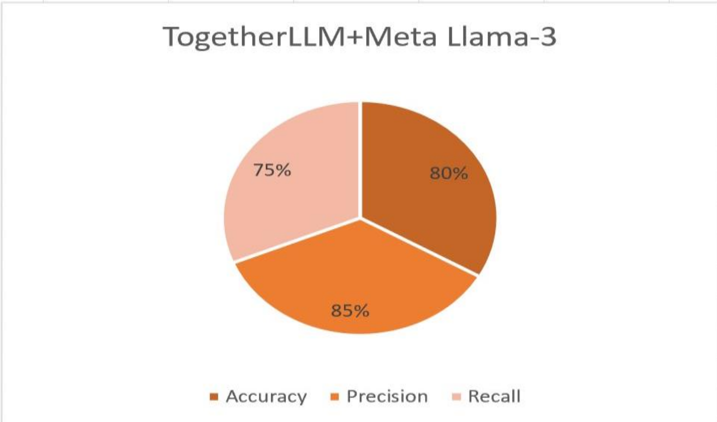

Fig. 12. Performance Evaluation Results

These results indicate that the Outfit Recommendation System performed exceptionally well on the chosen test cases, correctly identifying and recommending the most appropriate outfits for each user input. The system’s perfect scores in accuracy, precision, and recall reflect its ability to understand the context of user queries and provide highly relevant outfit recommendations.

V. CONCLUSION

In conclusion, the recommendation system presented in this paper exemplifies the integration of advanced natural language processing and real-time product data retrieval to create a highly effective tool for personalized fashion recommendations. By leveraging the TogetherLLM model, built on the Meta Llama-3 70B architecture, the system is capable of understanding complex user queries and generating relevant outfit suggestions tailored to individual preferences. The dynamic prompt engineering process ensures that user inputs are meticulously considered, enabling the model to produce actionable recommendations that align with specific occasions, styles, and budget constraints.

A key innovation of this system is its robust API integration with e-commerce platforms like Flipkart. The modules such as getProductCategory and getFlipkartProduct play a critical role in fetching real-time product data, including descriptions, prices, and availability. This ensures that the suggestions provided to users are not only stylistically appropriate but also readily actionable, enhancing the overall shopping experience. By filtering and matching the retrieved product data according to user-defined criteria, the recommendation engine effectively narrows down options, allowing users to explore outfits that meet their unique needs.

The interactive feedback loop embedded within the system promotes continuous user engagement, allowing for an iterative and dynamic shopping experience. Users can refine their preferences in real-time, adapting to new inputs and receiving updated suggestions instantaneously. This responsive design simulates a personalized shopping assistant, making the process both efficient and enjoyable. Furthermore, the userfriendly interface and straightforward exit options reinforce usability, catering to a broad audience regardless of technical proficiency.

Overall, the proposed recommendation system stands out not only for its technological sophistication but also for its practical application in the fashion retail landscape. As online shopping continues to evolve, the ability to deliver personalized, relevant, and actionable recommendations will be paramount in enhancing user satisfaction and driving sales. Future work may explore the incorporation of additional features, such as machine learning algorithms for improved predictive analytics, or expanded API integrations with other ecommerce platforms, further enriching the user experience and broadening the system's applicability in diverse retail contexts. By addressing the intricate needs of modern consumers, this system contributes significantly to the ongoing evolution of personalized e-commerce solutions.

VI.REFERENCES

[1] Samkit Chakraborty, Md. Saiful Hoque, Naimur Rahman
Jeem, Manik Chandra Biswas, Deepayan Bardhan, Edgar Lobaton, "Fashion Recommendation Systems, Models and Methods: A Review."
[2] Meshal Alduraywish, Bhuvan Unhelkar, Sonika Singh, Mukesh Prasad, "Recommendation Systems and Chatbots for Online Stores in Fast Fashion Industry," 2022.
[3] Wong Qin Yi, "Chatbot for Clothing Recommendation," 2023.
[4] Landim, A.R.D.B. et al., "Chatbot Design Approaches for fashion e-commerce: An interdisciplinary review," 2021.
[5] Schmelzer, R., "The fashion industry is getting more intelligent with AI," Forbes, 2021.
[6] Thiri Aung, Ei Chaw Htoon, "Recommender System For Fashion Design."
[7] Hyunwoo Hwangbo, Yang Sok Kim, "Recommendation system development for fashion retail e-commerce," 2018.
[8] Dario Di Palma, "Retrieval-augmented Recommender System: Enhancing Recommender Systems with Large Language Models," September 2023.
[9] Qianqian Chen, Tianyi Zhang, Maowen Nie, Zheng Wang, "Fashion-GPT: Integrating LLMs with Fashion Retrieval System," 2023.
[10] Yujuan Ding, Yunshan Ma, Wenqi Fan, Yige Yao, TatSeng Chua, Qing Li, "FashionReGen: LLM-Empowered Fashion Report Generation," 2023.
[11] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, "Recommender Systems in the Era of Large Language Models," 2024.
[12] Wu, S., Xiong, H., Zhang, Y., Zheng, L., & Lei, Q., "A Survey of Deep Learning Techniques for Fashion Recommendation," 2020.
[13] Gu, Q., & Zhou, J., "Deep Fashion: A Multi-modal Fashion Recommendation System," 2020.
[14] Yu, Q., Sun, Y., Wu, M., & Liu, Y., "Explainable AI for Fashion Recommendation Systems: A Review," 2023.