IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.67

Volume 5, Issue 1, March 2025

Smart Multiagent Shopping System

Prof. Samita Patil[1], Akshay Sonawane[2], Aaditya Revandkar[3]
Assistant Professor, Department of Computer Engineering[1]
Students, Department of Computer Engineering[2,3]
Shivajirao S Jondhale College of Engineering (SSJOE), Dombivli East, Mumbai, India
patilsamita85@gmail.com, akshaysonawane2510@gmail.com , aadityarevandkar@gmail.com

Abstract*:* *The Smart Multiagent Shopping System is an efficient tool that acts as a sophisticated web application designed to significantly improve user interface experience during the shopping visit by bringing into combination through the single architecture APIs from leading online e-commerce platforms like Amazon, Flipkart, Myntra, and Meesho. The system is among the most modern interfaces since it easily gathers and compares the details of products at least in terms of price and availability, and eventually features. This also enables the system to recommend the products to the user, which seems helpful as feedback for earlier searches throughout its true Personalized Recommendations. This becomes possible due to rewarding usefulness for general decision-making in terms of information flow and similar other criteria entailing to result accuracy, efficiency, and ease of operations. The work of this system is to facilitate a wishful query from a user to search his needs dynamically and deliver the most intelligent and automatic means to him. The paper addresses a number of significant challenges that exist in the unmediated arena of online shopping. What seems discouraging to many online shoppers is the constant and excessive purchase and the robust complexity in carriers and agents that provide a very competitive marketplace. The paper is on novel and requisite algorithms designed to assimilate API calls from various websites and conquer the obstacle of vigorous-looking, intelligent data representation. In the meantime, integrated prototypes are suggested to deliver sorted and appropriate outcomes while offering good performance. The research essentially tries connecting the ramifications of using multiagent applications in business practices to shopping paradise.*

Keywords: API Integration, Online Shopping, Price Comparison, Product Recommendation, Multiagent Architecture, Real-time Data Analysis, User-friendly Interface

I. INTRODUCTION

The Smart Multiagent Shopping System offers an innovative way to simplify the online shopping process and make the experience better while using multiagent architecture and advanced API integration. In today's e-commerce marketplace, users face certain problems such as platform fragmentation and time-consuming, manual comparisons. Information overload also deters the proper search for various products on available online platforms. To address these issues, this project integrates APIs from popular e-commerce websites like Amazon, Flipkart, Myntra, and Meesho to create a unified system that offers a seamless shopping experience. The system is designed to intelligently retrieve and compare product information such as prices, availability, and features in real time, delivering accurate and personalized recommendations based on user queries. It saves users a great deal of time and effort, while at the same time it ensures that the best possible options are provided to users for their needs. The system's interface is user-friendly, and its navigation is easy, making for a highly intuitive shopping experience. This project also highlights technical challenges such as handling diverse APIs, managing heterogeneous data formats, and ensuring scalability and performance in a multiagent environment. The Smart Multiagent Shopping System, by addressing these complexities, demonstrates the potential of combining multiagent systems and API integration to provide a robust and efficient solution for modern online shoppers. This initiative not only optimizes the decision-making process for users but also paves the way for future advancements in the e-commerce industry.

DOI: 10.48175/IJARSCT-23620

II. MOTIVATION

The exponential growth of e-commerce platforms has posed challenges for users to compare the variety of products on websites, such as Amazon, Flipkart, Myntra, and Meesho. Comparing them manually is inefficient and time-consuming with a potential for errors, leading to dissatisfaction among users. This project seeks to solve these issues by harnessing the power of multiagent systems and API integration to automate comparisons on various products, thereby offering real-time, accurate recommendations. The Smart Multiagent Shopping System would aim to make the online shopping experience as perfect as possible and advance innovation in the e-commerce domain by enhancing the efficiency of decision-making and challenges such as handling heterogeneous data and scalability.

III. RELATED WORK

Various studies have focused on the integration of multiagent systems and API-based solutions to optimize e-commerce platforms. Multiagent architectures have been widely recognized for their ability to distribute tasks, manage dynamic processes, and handle heterogeneous data effectively. For example, studies on e-commerce systems that use agent-based frameworks have been shown to greatly improve the automation of product discovery and recommendation processes.

Price comparison engines such as Google Shopping and Price Runner use web scraping or API integrations for aggregating product details from multiple websites. However, they do not provide personalization or real-time updates, which are necessary for dynamic needs of users. Similar works [3] that implement recommendation systems generally focus on single-platform solutions, and they limit the users from acquiring definite decisions across multiple platforms.

Recent innovations in the integration of APIs in multi-platform e-commerce systems [4] present an opportunity for automated solutions for data aggregation. However, inconsistent data formats, scalability, and accuracy in real-time comparisons remain inadequately addressed. Moreover, previous research in personalized shopping systems [5] has shown that user-specific preferences are effective; however, most such systems do not incorporate multi-platform data cohesively.

Such as the strength of the multiagent architectures and strength of the API integrations, this system improves them both to create a unified, scalable, and user-friendly solution for actual-time price and product feature comparisons. This work will address the gaps identified on scalability, personalization, and interoperability to take the cutting-edge online shopping systems to limits.

IV. SYSTEM ARCITECTURE

The Smart Multiagent Shopping System is designed using a multiagent framework to integrate and interact with APIs from multiple e-commerce platforms. The architecture consists of the following components:

User Interface (UI): A user-friendly front-end application that allows users to input search queries, view results, and interact with the system. It provides features like product comparison, sorting, and filtering. Request Manager: This acts as an intermediary between the UI and backend. It accepts queries from end users, processes them, formats requests to APIs, and then posts them to the multiagent system. Multiagent System: A distributed framework involving several agents Query Agent: Deals with the search queries from the end-user and transfers those queries to the corresponding platform agents.

Platform Agents: Those interact with APIs of different e-commerce platforms such as Amazon, Flipkart, Myntra, and Meesho to fetch product information. Comparison Agent: This module aggregates data from several platform agents, processes it, and ranks the results based on user preferences, such as price, rating, and availability. Data Integration and Processing Module: This module is in charge of normalizing and cleaning the data received through various APIs to ensure consistent product comparison. Recommendation Engine: Generates recommendations based on the ranking algorithms above, considering user behaviour's and preferences. Output Module: Displays the final processed data to users in an organized and intuitive format. APIs: Fetches real-time information from e-commerce platforms about prices, availability, and specifications for the products.

DOI: 10.48175/IJARSCT-23620

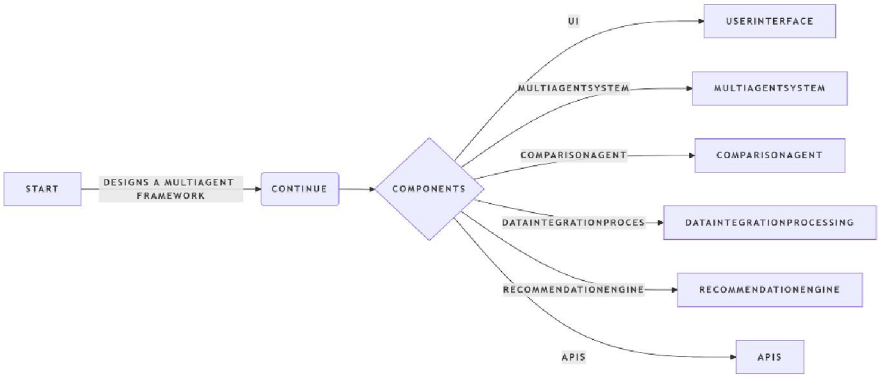


FIG IV. SYSTEM ARCITECTURE

IV. I DESIGN

The design for the Smart Multiagent Shopping System is based on modularity, scalability, and user-centric functionality with each constituent designed to perform specific tasks in an efficient manner while ensuring that it integrates seamlessly with the entire system. User Interface (UI): To give the user the simplest, most intuitive, and aesthetically pleasing form of interaction. Search bar for product queries.

Filters: Sorting results by price, rating, and availability A responsive design for usability on all devices, including mobile, tablet, and desktop. Request Manager: Objective: Acts as a communication bridge between the front-end and backend systems. Design Elements: Accepts search queries from the UI. Formats the query to suit API requirements. Routes requests to the Multiagent System. Multiagent System: Core Component: Composed of three specialized agents: Query Agent: Manages user queries and forwards them to the relevant Platform Agents. Platform Agents: Specialized for particular e-commerce platforms, such as Amazon, Flipkart, Myntra, Meesho. Each agent fetches product data via API. Comparison Agent: Combines and normalizes the information gathered by the Platform Agents into recommendation-ready form.

Data Integration and Processing Module: Goal: Cleaning, formatting, and harmonization of data pulled from a wide variety of APIs. Core Features: Tolerate heterogenous data formats of various platforms Solve differences in product names, prices, and stock. Goal: Developing personalized, ranked product recommendations

Features: Uses algorithms like collaborative filtering or rule-based ranking. Recommendations optimized against user preference parameters such as budget, brand, etc. Output Module: Objective: To present the processed and ranked data to the users in a visually appealing way. Design Details: Display results in a card-based layout with key product details (name, price, ratings, and a direct purchase link). API Integration: Objective: To enable real-time communication with e-commerce platforms. Features: RESTful APIs for secure and efficient data exchange. Handle rate limits, authentication, and error management.

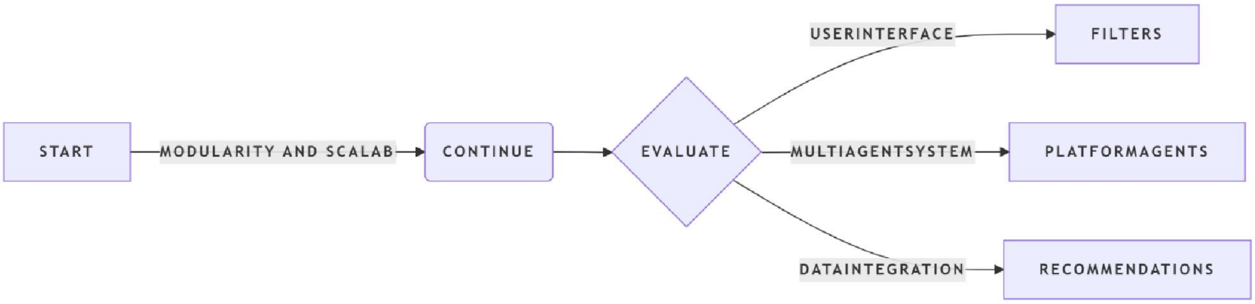


FIG IV.I. Design of Smart Multiagent Shopping System

DOI: 10.48175/IJARSCT-23620

V. SYSTEM IMPLEMENTATION

The design of the Smart Multiagent Shopping System is modularity, scalability, and user-centric functionality. Each component will be designed to perform specific tasks efficiently while integrating well with the whole system. User Interface (UI): Goal: It should be a simple, intuitive, and attractive interface for the users. Features: Search bar for product query, Filters for result sorting (price, rating, availability) A responsive design to be usable on various devices (mobile, tablet, desktop).

Request Manager: Objective: Act as a bridge between the frontend and the backends. Design Elements: Takes in the search query requested by the user interface. Re-shapes the query to fit the API requirements. Forwards the requests to the Multiagent System. Multiagent System: Component: The core of three types: Query Agent: Deals with end user queries and forwards them to the appropriate Platform Agents. Platform Agents: Entails each of the kind of e-commerce available namely Amazon, Flipkart, Myntra, Meesho. All the agents fetch product data through the API. Comparison Agent: It aggregates and normalizes data from Platform Agents, preparing data for recommendation. Data Integration and Processing Module: Task: It cleans, formats, and integrates data fetched from distributed APIs. Major Functions: Deal with different formats in heterogenous data coming from multiple platforms. Resolve inconsistencies in names of products, price, and availability. Recommendation Engine Task: Suggest personal and ranked product recommendations. Use Features: Algorithms like collaborative filtering or rule-based ranking. Optimize the recommendations as per user choice of budget, brand, etc. Output Module Objective: Present the processed and ranked data to users in a visually pleasing manner. Design Details: Display results in card form with key details of the products, such as name, price, ratings, and a link to buy the product directly. API Integration Objective: Realtime communication with the e-commerce sites.

Features: RESTful APIs ensure secure and effective data exchange. Deal with rate limits, authentication The implementation of the Smart Multiagent Shopping System is built on top of a multi-agent framework (JADE) for the autonomous search, comparison, and recommendation of products from diverse e-commerce websites, such as Amazon, Flipkart, Myntra, and Meesho. The details of the implementation with its main parts are described as follows:

Agent Framework Setup: The system uses the Java Agent Development Framework (JADE) to design several agents that simulate shopping behaviours'. Each agent is designated to interact with a specific e-commerce site and retrieve product information. Agents are designed for communication purposes with other agents and result sharing through a platform-independent messaging system. Hence, such coordination among agents is real-time for comparative product recommendation. Agent behaviour's involves the following: Search for products: Based on user queries, agents fetch product data such as name, price, description, and ratings. Product comparison: Agents compare products by price, features, and user reviews from different e-commerce platforms. Recommendation generation: After comparison, agents recommend the best product based on predefined criteria such as lowest price or highest rating.

Product Data Fetching and Comparison: The system fetches all real-time product data from APIs of multiple e-commerce platforms (Amazon, Flipkart, Myntra, Meesho). A comparison algorithm is implemented to evaluate the products based on their respective attributes like price, features, and ratings. It makes use of a custom comparator to compare the products based on the most relevant parameters. It will provide the optimal shopping suggestion in an optimal manner by minimizing the cost while considering product quality and users' preferences.

The backend uses Spring Boot for the management of HTTP requests to the multi-agent system, for interaction between agents, and between agents and the database. The frontend uses RESTful APIs for querying the results of product recommendations and comparisons. the frontend is developed using React.js to give the interface a look and feel intuitive and responsive for the users. Users are able to search for products, view product comparisons, and receive shopping recommendations. The interface communicates with the backend through RESTful API calls to output the recommended products in an easily readable format.

System Integration and Testing: The entire agent, backend, and frontend are integrated into a unified system. The system is also tested in real time using data fetched from the integrated APIs with the purpose of receiving proper product fetching, comparison, and recommendation. Latency reduction, improvement of user experience, and system performance are optimized techniques applied.

DOI: 10.48175/IJARSCT-23620

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.67

Volume 5, Issue 1, March 2025

VI. RESULT

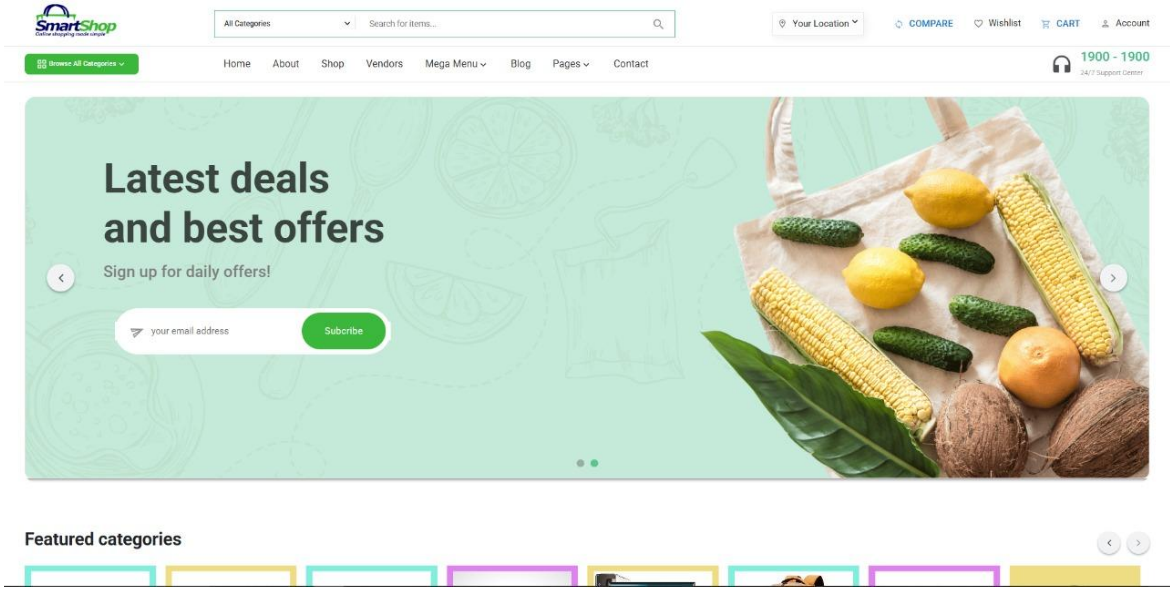


Fig VI.I : Home page

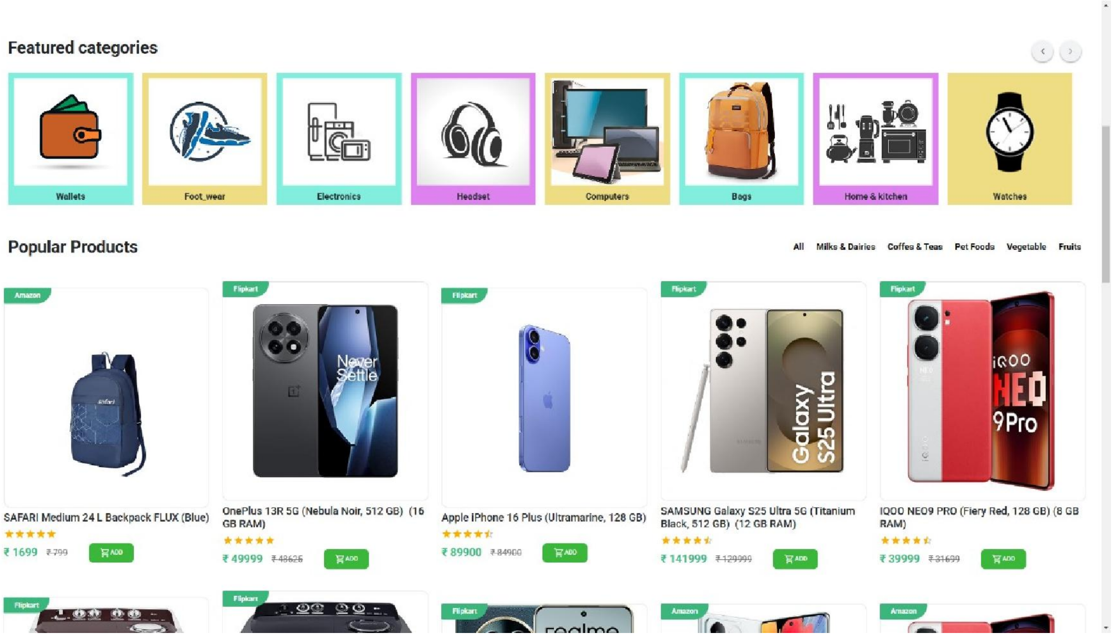
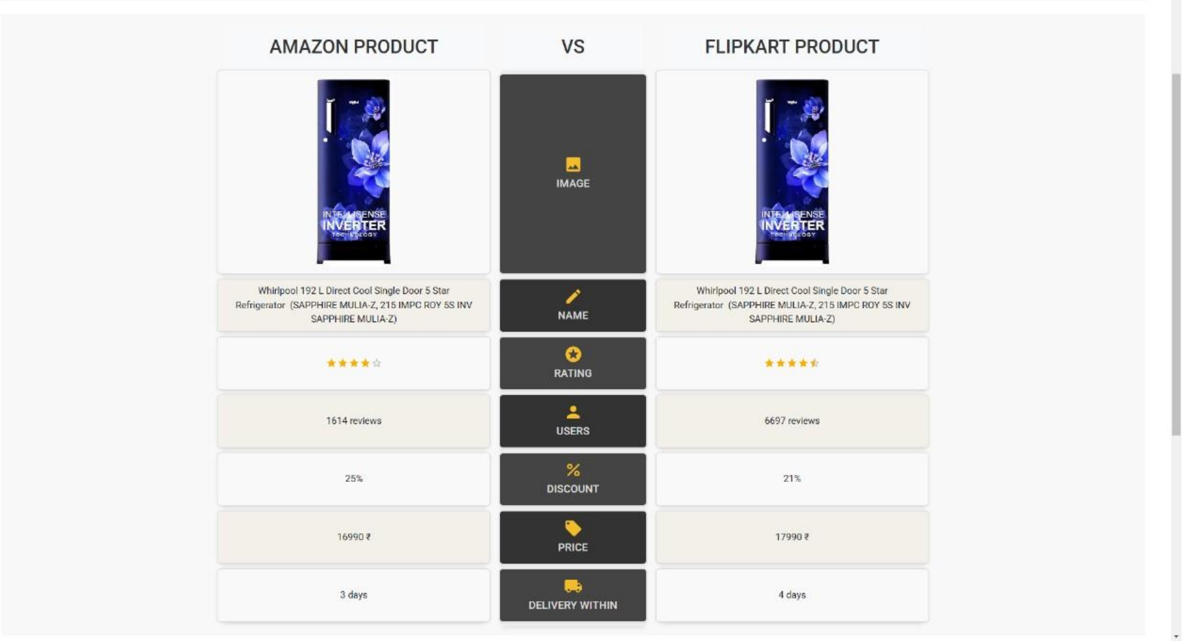
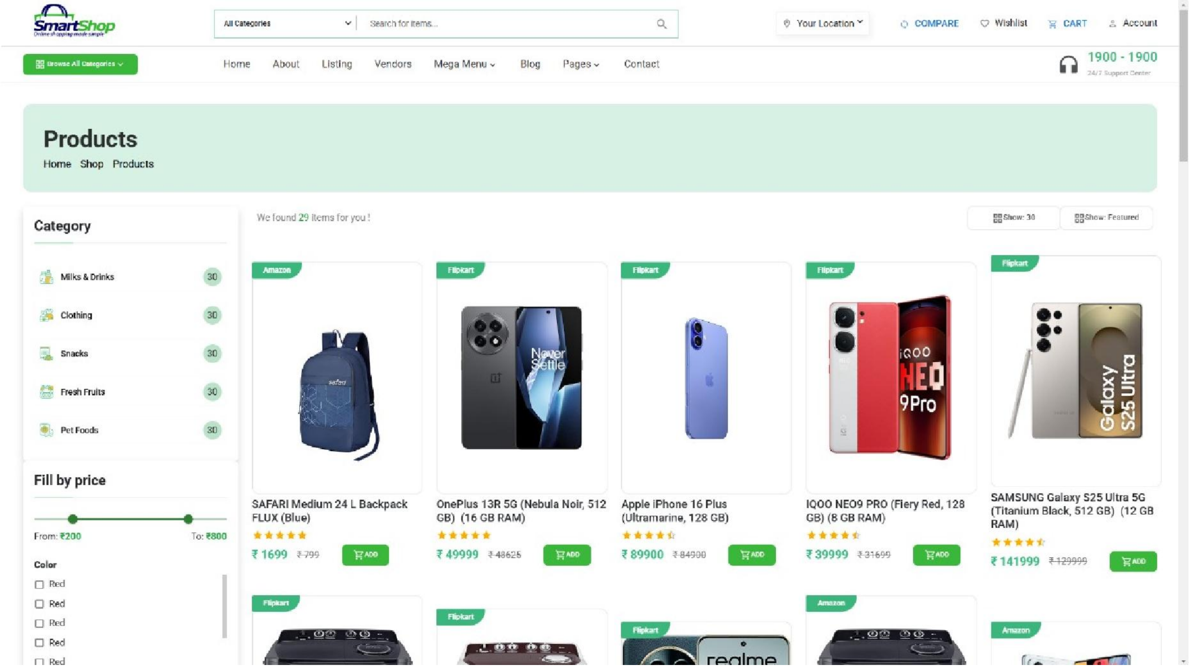


Fig VI.I: Product page



AMAZON PRODUCT	VS	FLIPKART PRODUCT
	IMAGE	
Whirlpool 192 L Direct Cool Single Door 5 Star Refrigerator (SAPPHIRE MULIA-Z, 215 IMPC ROY 5S INV SAPPHIRE MULIA-Z)	NAME	Whirlpool 192 L Direct Cool Single Door 5 Star Refrigerator (SAPPHIRE MULIA-Z, 215 IMPC ROY 5S INV SAPPHIRE MULIA-Z)
★★★★☆	RATING	★★★★½
1614 reviews	USERS	6697 reviews
25%	DISCOUNT	21%
16990 ₹	PRICE	17990 ₹
3 days	DELIVERY WITHIN	4 days

fig VI.III Compare Section



SmartShop
Online shopping made simple

All Categories | Search for items...

Your Location | COMPARE | Wishlist | CART | Account

Browse All Categories | Home | About | Listing | Vendors | Mega Menu | Blog | Pages | Contact

1900 - 1900
24/7 Support Center

Products
Home Shop Products

Category

Milks & Drinks	30
Clothing	30
Snacks	30
Fresh Fruits	30
Pet Foods	30

Fill by price
From: ₹200 To: ₹800

Color
□ Red
□ Red
□ Red
□ Red
□ Red

We found 29 items for you !

Show: 30 | Show: Featured

Amazon
SAFARI Medium 24 L Backpack FLUX (Blue)
★★★★★
₹ 1699 ~~₹799~~ ADD

Flipkart
OnePlus 13R 5G (Nebula Noir, 512 GB) (16 GB RAM)
★★★★★
₹ 49999 ~~₹48625~~ ADD

Flipkart
Apple iPhone 16 Plus (Ultramarine, 128 GB)
★★★★½
₹ 89900 ~~₹84900~~ ADD

Flipkart
IQOO NEO9 PRO (Fiery Red, 128 GB) (8 GB RAM)
★★★★½
₹ 39999 ~~₹31699~~ ADD

Flipkart
SAMSUNG Galaxy S25 Ultra 5G (Titanium Black, 512 GB) (12 GB RAM)
★★★★½
₹ 141999 ~~₹129999~~ ADD

Flipkart
Flipkart
Flipkart
Amazon
Amazon

fig vi. iv listing page

VII. CONCLUSION

We proposed the design and implementation of a Smart Multiagent Shopping System in this paper. The system utilized a multi-agent framework to search, compare, and recommend products across various e-commerce platforms autonomously. Agents were used for platform-specific data retrieval and a comparison algorithm based on product attributes such as price, features, and ratings to ensure optimal shopping choices are presented to users. Integration of JADE for agent-based communication, along with robust backend services developed in Spring Boot and a responsive frontend in React.js, gives it a very smooth user experience. The system has greatly improved the efficiency of shopping and decision-making through real-time product data fetching and intelligent recommendations. The proposed system is scalable, easy to adapt to new platforms, and can be further optimized for performance. Future work can be directed toward further developing the recommendation logic using more advanced machine learning techniques and further expanding the system to support more personalized user experiences.

VIII. ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my project guide, **Prof. Samita Patil**, for her valuable guidance, continuous support, and constructive feedback throughout the development of this Smart Multiagent Shopping System. Her expertise and encouragement were instrumental in shaping this project. I would also like to thank my college, **Shivajirao S. Jondhale College of Engineering**, for providing the necessary resources and an environment conducive to learning and innovation. Special thanks to the faculty members and my peers for their assistance and encouragement.

REFERENCES

[1] Priyanka Bingi [2021] “Ai based multi-agent online Shopping System “IRJET

[2] RamazanTeimouriYansari [2023] Design an Intelligent Multi-agent Computer-aided Model for recommender Systems.

[3] Bhavna Galhotra (2019) “Evolution of E-commerce in India”, Xplore, HTTP://ieeexplore.ieee.org/document.

[4] M. Altulyan, L. Yao, X. Wang, C. Huang, S. S. Kanhere, and Q. Z. Sheng, "A survey on recommender systems for Internet of Things: techniques, applications and future directions," The Computer Journal, vol. 65, no. 8, pp. 2098 2132, 2022.

[5] D. Roy and M. Dutta, "A systematic review and research perspective on recommender systems," Journal of Big Data, vol. 9, no. 1, p. 59, 2022/05/03 2022.

[6]E. Kozegar, H. Yarmohammadi, M. Bazargani, and Z. Homayounpour, "Presenting a novel method based on collaborative filtering for nearest neighbor detection in recommender systems," Intelligent Multimedia Processing and Communication Systems (IMPCS), vol. 1, no. 1, pp. 5564, 2020.

[7] R. TeimouriYansari and M. Ajoudani, "Development of a multiagent recommender system for intelligent shopping assistants " Intelligent Multimedia Processing and Communication Systems (IMPCS), vol. 4, no. 2, pp. 110, 2023.

[8] A. Tharwat, "Classification assessment methods," Applied Computing and Informatics, vol. 17, no. 1, pp. 168192, 2020.

[9]] B. Zhou, J. Bao, J. Li, et al., A novel knowledge graph-based optimization approach for resource allocation in discrete manufacturing workshops, Robot. Comput. Integr. Manuf. 71 (3) (2021).

[10]F.F. Zhang, S. Nguyen, Y. Mei, M.J. Zhang, Genetic Programming For Production Scheduling, Springer, Singapore, 2021.

[11] M. Dumic, D. Jakobovic, Ensembles of priority rules for resource constrained project scheduling problem, Appl. Soft Comput. 110 (2021).

[12] G.K. Yun, S. Lee, J. Son, et al., Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system, J. Manuf. Syst. 57 (2020) 440–450.

[13] A. Kuhnle, J.P. Kaiser, F. Thei, et al., Designing an adaptive production control system using reinforcement learning, J. Intell. Manuf. 32 (3) (2021) 855–876.

[14] Z.Z. Zhang, H. Liu, M.C. Zhou, et al., Solving dynamic traveling salesman problems with deep reinforcement learning, IEEE Trans. Neural Netw. Learn. Syst. (2021),

[15] R. Canlas, “Capturing security mechanisms applied to E-commerce: An analysis of transaction security,” International Journal of Security and its Applications, vol. 15, no. 2, pp. 1–10, 2021.

[16] P. D. Shah and R. S. Bichkar, “Secret data modification based image steganography technique using genetic algorithm having a flexible chromosome structure,” Engineering Science and Technology, An International Journal, vol. 24, no. 3, pp. 782–794, 2021.

[17] A. D. Salman and H. B. Abdulwahab, “Proposed copyright protection systems for 3D video based key frames,” Iraqi Journal of Computers, Communications, Control and System Engineering, vol. 20, no. 3, pp. 1–15, 2021.

[18] S. J. Pon, S. S. Ramya, A. V. Christal and K. Mythili, “Secured payment gateway for authorizing e-commerce websites and transactions using machine learning algorithm,” in IEEE Int. Conf. on Computer Communication and Informatics (ICCCI), Coimbatore, India, pp. 1–5, 2020.

[19] W. Jiang, Y. Wang, Y. Jiang, Y. Xu, J. Chon et al., “Mobil internet mobile agent system dynamic trust model for cloud computing,” Computers, Materials & Continua, vol. 62, no. 1, pp. 123–136, 2020.

[20] N. Saxena, H. Shen, N. Komninos, K. K. R. Choo, N. S. Chaudhari et al., “VPSMS: A batch verification protocol for end-to-end secure sms for mobile users,” IEEE Transactions on Dependable and Secure Computing, vol. 17, no. 3, pp. 550–565, 2020.

[21] J. Hu, J. Deng, N. Gao and J. Qian, “Application architecture of product information traceability based on block chain technology and a lightweight secure collaborative computing scheme,” in Int. Conf. on E-Commerce and Internet Technology (ECIT), Zhangjiajie, China, pp. 335–340, 2021.

[22]W.A.Hassan, Y. H. Ali and N. J. Ibraheem, “A survey of latest techniques in medical image classification,” in Int. Conf. on Communication & Information Technology (ICICT), Basrah, Iraq, pp. 1–6, 2021.

[23] H. Quas, J. Tarazi and V. Akre, “Secure E-auction system using blockchain: UAE case study,” in Advances in Science and Engineering Technology Int. Conf. (ASET), Dubai, United Arab Emeritus, pp. 1–5, 2020

[24] I. M. Alattar and A. M. S. Rahma, “New block cipher algorithm using magic square of order five and galois field arithmetic with dynamic size block,” International Journal of Interactive Mobile Technologies, vol. 15, no. 16, pp. 63–78, 2021.

[25]S. M. Kareem and A. M. S. Rahma, “New method for improving add round key in the advanced encryption standard algorithm,” Information Security Journal, vol. 30, no. 6, pp. 371–383, 2021.

DOI: 10.48175/IJARSCT-23620