International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Smartkart: Chatbot Integrated in E-Commerce Platform Using Simple AI

Mrs.K.Surekha[1], Annamneedi Venkata Sai Rama Keerthi[2], Adari Divya[3], Busa Yaswanth [4], Ganisetti Mounika[5], Guttula Satish[6]

[1]Associate Professor, Cse Department, Vsm College Of Engineering, Amachandrapuram, Dr.B.R.Ambhedkar Konaseema Dt, Andhra Pradesh - 533255
[2,3,4,5,6]Student, Cse Department, Vsm College Of Engineering, Amachandrapuram, Dr.B.R.Ambhedkar Konaseema Dt, Andhra Pradesh - 533255

ABSTRACT
The rapid evolution of digital commerce has led to an increased demand for intelligent, personalized, and efficient online shopping platforms. SMARTKART is an AI-driven e-commerce system that leverages machine learning algorithms to enhance user experience through personalized product recommendations, real-time chatbot assistance, and automated fraud detection. Developed using Django and Python, the platform adopts a modular architecture that supports scalability and maintainability. SMARTKART analyzes user behavior, purchase history, and interaction data to dynamically generate relevant product suggestions and facilitate seamless transactions. The integrated chatbot enhances customer engagement by providing instant, AI-powered support. With robust admin features for managing products and users, SMARTKART serves as a comprehensive solution for modern e-commerce challenges. The system was rigorously tested and demonstrated high accuracy in recommendations and system responsiveness, validating its real-world applicability. This project sets the foundation for future enhancements including augmented reality features, multilingual support, and predictive analytics, positioning SMARTKART as a scalable and intelligent retail solution**.**

Keywords: E-commerce, Artificial Intelligence, Personalization, Chatbot, Django, Machine Learning, Recommendation.

INTRODUCTION
The proliferation of internet technologies and the increasing demand for digital convenience have led to a remarkable transformation in global commerce. E-commerce, in particular, has become an integral part of modern life, allowing consumers to shop anytime, anywhere, with access to a vast array of products and services. However, traditional e-commerce platforms often fall short in meeting the personalized needs and expectations of users, relying heavily on static interfaces, generic product suggestions, and delayed support systems. To address these challenges, the SMARTKART project introduces an AI-driven e-commerce solution that delivers intelligent, real-time personalization and automation. By integrating machine learning algorithms,), and chatbot support into the shopping ecosystem, SMARTKART enhances customer engagement, boosts recommendation accuracy, and streamlines overall operations. This platform not only provides dynamic product suggestions based on individual user behavior but also

automates customer assistance through a responsive chatbot interface.

SMARTKART is developed using Django—a robust Python web framework—alongside scalable backend and frontend technologies. It adopts a modular architecture that facilitates secure user authentication, seamless product browsing, real-time order management, and insightful administrative analytics. The implementation of this system demonstrates the practical potential of AI in reshaping digital retail environments into smart, adaptive, and customer-centric platforms.

EXISTING SYSTEM

Traditional e-commerce platforms operate primarily on static content and rule-based recommendation systems. These platforms present the same user interface and product suggestions to every customer, regardless of individual preferences or shopping behavior. As a result, users often receive irrelevant recommendations, leading to reduced engagement, poor user satisfaction, and lower conversion rates.

In most conventional systems:

- Product discovery is limited to keyword-based search and manual filtering.
- Personalization is nearly absent or extremely basic, based solely on categories or popularity.
- Customer support is often restricted to FAQs or delayed manual interactions, lacking real-time assistance.
- Fraud detection mechanisms are minimal or entirely absent, creating risks in payment and order processes.

Additionally, these systems lack adaptive learning or real-time analytics capabilities. They are not designed to learn from user actions or optimize experiences dynamically. This stagnancy results in a subpar shopping journey, with high cart abandonment rates and limited trust from users.

Disadvantages of Existing Systems

- Limited Decision Support: No intelligent product recommendations based on user history or behaviour.
- Lack of Automation: Manual operations dominate both user and admin experiences.
- Weak Security and Fraud Prevention: Basic verification systems make platforms vulnerable to fraud.
- Minimal AI Integration: No utilization of machine learning for personalization or optimization.
- Poor User Experience: Generic layouts, non-interactive interfaces, and outdated navigation flows.

PROBLEM STATEMENT

Traditional e-commerce platforms typically offer static interfaces, generic product listings, and rule-based recommendation engines, providing the same experience to every user regardless of their preferences, behaviour, or purchase history. These systems lack the capability to adapt dynamically to individual user needs, resulting in a one-size-fits-all experience that often leads to user dissatisfaction, low engagement, and reduced conversion rates.

Moreover, conventional systems are limited in their ability to provide real-time support and automation. Customer queries are often handled manually or through delayed response systems like FAQs and support tickets, causing friction in the user experience. Security vulnerabilities such as inadequate fraud detection, poor session management, and unreliable transaction monitoring further reduce consumer trust in these platforms.

In today's competitive and rapidly evolving digital market, there is a growing need for an intelligent, adaptive, and secure e-commerce solution. This leads to the necessity of a system like SMARTKART—

a platform designed to integrate AI-driven personalization, intelligent chatbots, and fraud detection to offer a seamless, interactive, and secure online shopping experience**.**

PROPOSED SYSTEM

The proposed system, **SMARTKART**, is an intelligent e-commerce platform designed to enhance online shopping through the integration of artificial intelligence (AI) and machine learning (ML) technologies. Unlike traditional platforms that provide static interfaces and limited interactivity, SMARTKART adapts dynamically to individual user behaviours, delivering a personalized and interactive experience.

SMARTKART employs machine learning algorithms to analyse user preferences, search history, and purchase patterns. This data-driven approach enables the system to generate real-time, highly relevant product recommendations, improving user engagement and satisfaction. The recommendation engine is built using collaborative filtering and content-based filtering methods.

Developed using the Django web framework, SMARTKART ensures scalability, modularity, and security. It incorporates a responsive user interface and a robust backend that supports features such as:

- Dynamic product search and filtering
- Secure user authentication and session management
- AI-driven product recommendation system
- Real-time chatbot support using natural language processing (NLP)
- Administrative dashboard for product and order management

The platform includes separate modules for customers and administrators. Customers can register, browse products, manage their carts, place orders, and receive AI-generated recommendations. Administrators, on the other hand, can manage products, track orders, analyse user behavior, and retrain recommendation models as needed.Additionally, SMARTKART supports real-time customer support via an AI chatbot, which improves the user experience by providing immediate assistance for product-related queries. The chatbot evolves through continuous learning from user interactions.

The system is designed with a modular architecture to support future enhancements such as augmented reality (AR) integration, multilingual support, predictive analytics, and voice-based navigation. Overall, SMARTKART provides a secure, intelligent, and scalable solution for modern e-commerce challenges, delivering value to both users and administrators.

IMPLEMENTED ALGORITHMS

Logistic Regression

Logistic regression is a supervised machine learning algorithm used for binary classification tasks. In the SmartKart chatbot, logistic regression can be effectively used to classify user queries, such as distinguishing between product-related questions and support requests.

The algorithm works by modelling the probability that a given input belongs to a particular class. It applies the logistic (sigmoid) function to a linear combination of input features to produce an output between 0 and 1, representing a probability.

In the context of SmartKart, input features such as keywords, query length, and user interaction history are used to train the model. The predicted probabilities help the chatbot decide how to respond — whether to recommend a product, redirect to customer support, or answer FAQs.

Logistic regression is chosen for its simplicity, interpretability, and efficiency, making it suitable for real-time chatbot response classification.

WORKING OF MACHINE LEARNING ALGORITHMS

Machine learning (ML) algorithms enable systems to learn patterns from historical data and make intelligent decisions without being explicitly programmed. In the SmartKart chatbot, ML plays a crucial role in understanding user intent, classifying queries, and providing relevant product recommendations. The process begins with data collection from user interactions such as chats and browsing behavior. This data undergoes preprocessing to remove noise and extract meaningful features. The cleaned data is then used to train models like logistic regression, which learn to associate specific input patterns with desired outputs. Once trained and validated, the model is integrated into the chatbot to make real-time predictions. As users continue to interact with the system, new data is collected and used to fine-tune the model, allowing the chatbot to become more accurate and responsive over time.

SYSTEM ARCHITECTURE

The system architecture of SMARTKART shows how all parts of the platform work together.
At the top, we have the users, which include both customers and admins.Customers can access the website to register, log in, browse products, add items to the cart, and place orders.Admins can log in to manage products, view orders, and handle user information.All user requests are sent to the Django web server, which controls the flow of data.The Django server handles logic like login checks, showing products, saving orders, and managing chatbot responses.The database (SQLite) stores all the data, including user details, product information, and order history. The chatbot is connected to the system and uses logistic regression to provide replies to user queries.Each part of the architecture is connected smoothly so that the system works without delay. This design makes the system modular, secure, and easy to scale or upgrade in the future. The system architecture of SMARTKART shows how all parts of the platform work together. At the top, we have the users, which include both customers and admins.Customers can access the website to register, log in, browse products, add items to the cart, and place orders.Admins can log in to manage products, view orders, and handle user information. All user requests are sent to the Django web server, which controls the flow of data. The Django server handles logic like login checks, showing products, saving orders, and managing chatbot responses.The database (SQLite) stores all the data, including user details, product information, and order history.The chatbot is connected to the system and uses logistic regression to provide replies to user queries.Each part of the architecture is connected smoothly so that the system works without delay.
This design makes the system modular, secure, and easy to scale or upgrade in the future.

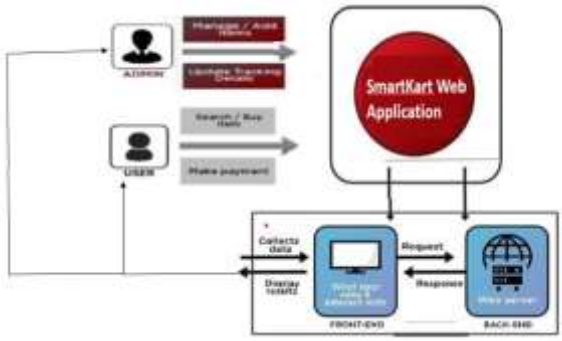

Fig:2 System Architecture Diagram

1. Modules:

SMARTKART is divided into four important modules, each with its own role in the system.

The Customer Module allows users to register, log in, view products, manage their profile, and place orders.It also lets users add items to the cart and check out with payment and shipping details.The Admin Module is used by the admin to log in and manage all backend activities.Admins can view and delete user accounts, manage orders, and control product listings.The Product Management Module lets the admin add new products, update details, or delete old ones.Products can be filtered and displayed based on categories, prices, or availability.The Chatbot Module helps users by answering basic questions and guiding them through the website.This chatbot improves the shopping experience by providing quick help without human support.Together, these modules make SMARTKART a complete and easy-to-manage e-commerce system.

RESULT ANALYSIS

SmartKart:

Fig:3

Fig:4

Fig:5

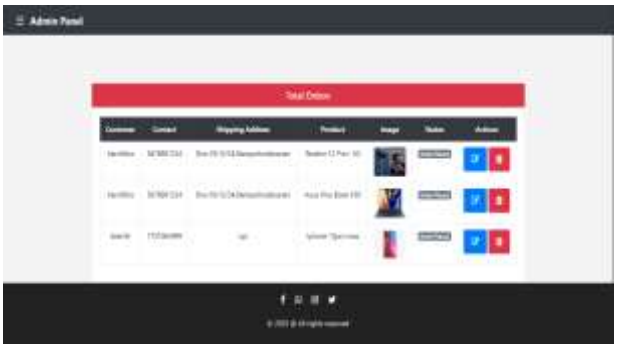

Fig:6

Fig:7

CONCLUSION

SMARTKART is a simple and smart e-commerce platform that improves the shopping experience. It provides product suggestions using AI and supports users through an integrated chatbot. The system is built with Django, making it secure, scalable, and easy to manage. Admins can control products, users, and orders from a clean dashboard. Overall, SMARTKART is a complete, user-friendly, and practical solution for online shopping.

REFERENCES

1. 10.1 " Sheetal Taneja; Pratibha Gupta R, Python as a tool for web server application development, JIMS8I-International Journal of Information Communication and Computing Technology l. (2018)
2. 10.2 " Kavya S.L; Dr.Sarathambekai S, Python Libraries and Packages for Web Development - A Survey, International Journal of Innovative Research in Technology.(2019)
3. 10.3 " Surya Teja N, A Study on Different Framework Architectures, International Journal of

Innovative Research in Science, Engineering and Technology. (2018) – 10.4 " Adamya Shyam; Nitin Mukesh, A Django Based Educational Resource Sharing Website: Shreic, Journal of Scientific Research (2018) –

4. (https://www.bhu.ac.in/research_pub/jsr/Volumes/JSR_64_01_2020/34.pdf)
5. 10.5 Ahmed Yunus; Md Masum, Design and Development of an E - Commerce System in a Rapid Organized Way, International Journal of Science and Research.(2020)
6. 10.6 Busari O.A; Adebisi O.A; AdeagaI.I; Oni A.A, Development of an Online Shop with Python Web Framework (Django), International Journal of Advanced Research in Science,EngineeringandTechnology(2021)System: A Review" by A. A. Patel et al. (2020