STM JOURNALS

Journal of
Open Source Developments

ISSN: 2395-6704
Volume 13, Issue 1, 2026
January–April
DOI (Journal): 10.37591/JoOSD

https://journals.stmjournals.com/joosd

Research JoOSD

NexCart—The Ultimate Shopping Platform: A Full-Stack E-Commerce Web Application

S.N.V.D.L. Priyanka[2,*], Nakka Pranay[1], Moka Akhilesh[1], Mukalla Naveen[1], Katadi Sai Asrith[1]

Abstract

The rapid growth of internet technology has significantly transformed traditional shopping methods into digital commerce platforms. This paper presents NexCart, a web-based e-commerce application designed to provide a simple, scalable, and secure online shopping experience. The system allows users to browse products, manage shopping carts, place orders, and perform secure transactions efficiently. The platform is built with HTML, CSS, and JavaScript on the frontend, while the backend and server-side functionality are managed using Python Flask. MySQL is utilized as a database to ensure secure and structured storage of user, product, and order information. NexCart incorporates key modules such as product browsing, cart management, order processing, and an administrative dashboard for managing system activities effectively. The system is structured using a three-layer architecture that includes presentation, application, and database tiers, helping to improve scalability, simplify maintenance, and boost overall performance. Testing results demonstrate that the system operates efficiently with reliable performance, accurate data handling, and secure transaction management. Overall, the project provides a practical implementation of full-stack web development and serves as a useful model for building modern e-commerce platforms.

Keywords: E-commerce, web application, Flask, MySQL, online shopping, full-stack development

INTRODUCTION

The rapid growth of the internet has greatly changed how businesses function and how consumers buy products and services. E-commerce systems have become an essential component of modern digital economies by enabling seamless online transactions between businesses and customers. These systems remove location-based limitations, allowing businesses to connect with customers worldwide while providing users with the flexibility to shop whenever and wherever they prefer.

Online shopping platforms offer several advantages, including convenience, global accessibility, a wide variety of products, and time efficiency. With the rapid increase in internet penetration and widespread smartphone usage, the demand for reliable, secure, and user-friendly e-commerce platforms continues to grow. Companies are progressively embracing digital technologies to remain competitive and effectively respond to changing customer demands.

This paper presents NexCart, The Ultimate Shopping Platform, a web-based application developed to demonstrate the implementation of a complete e-commerce system [1]. The system allows users to explore available products, add selected items to their carts, complete purchases, and carry out secure payments. It also provides an admin panel for effectively handling product listings, user accounts, and order processing [2].

***Author For Correspondence**
S.N.V.D.L. Priyanka
E-mail: snvdlpriyanka.bvce@bvcgroup.com

[1]Student. Department of Electronics and Communication Engineering, Bonam Venkata Chalamayya Engineering College (Autonomous), Odalarevu, Andhra Pradesh, India
[2]Assistant Professor, Department of Electronics and Communication Engineering, Bonam Venkata Chalamayya Engineering College (Autonomous), Odalarevu, Andhra Pradesh, India

Received Date: April 02, 2026
Accepted Date: April 12, 2026
Published Date: April 30, 2026

Citation: S.N.V.D.L. Priyanka, Nakka Pranay, Moka Akhilesh, Mukalla Naveen, Katadi Sai Asrith. NexCart—The Ultimate Shopping Platform: A Full-Stack E-Commerce Web Application. Journal of Open Source Developments. 2026; 13(1): 16–23p.

The NexCart platform integrates frontend technologies with backend processing and database management to create a scalable and user-friendly online shopping environment. Additionally, the system follows a modular design approach, making it flexible for customization and simplifying the addition of future features, such as payment gateway support, recommendation engines, and improved security mechanisms. This project serves as a practical example of full-stack web development and highlights the core functionalities required to build a modern e-commerce platform [3].

RELATED WORK

Several large-scale e-commerce platforms, such as Amazon, Flipkart, and eBay, have revolutionized the online shopping experience by offering advanced features, such as personalized product recommendations, secure payment gateways, real-time order tracking, and efficient logistics management systems. These platforms utilize modern technologies, including cloud computing, artificial intelligence, and big data analytics, to handle millions of users and transactions simultaneously.

Despite their success, these systems are highly complex and require significant infrastructure, development expertise, and financial investment. Their architecture is often tightly coupled with enterprise-level services, making them less suitable for academic projects or small-scale business implementations. Additionally, the integration of multiple services, such as inventory management, payment processing, and delivery tracking, increases system complexity and maintenance overhead.

Several studies and small-scale e-commerce solutions have attempted to simplify these systems by focusing on core functionalities such as user authentication, product catalog management, and basic checkout processes. However, many of these implementations lack scalability, modularity, and user-friendly interfaces, which limit their practical applicability [4].

The proposed NexCart platform addresses these challenges by providing a lightweight, modular, and easy-to-understand e-commerce architecture. It emphasizes core functionalities such as user sign-up, exploring products, managing the shopping cart, and handling orders, while maintaining a simple and easy-to-implement design. The system was developed using standard web technologies, making it highly customizable and accessible to students and small business owners [5].

Furthermore, NexCart emphasizes scalability and flexibility by adopting a modular design approach, allowing developers to easily extend functionalities, such as recommendation systems, payment gateways, and AI-based features, such as virtual try-on. This makes the platform suitable for learning purposes and adaptable to real-world applications with minimal enhancements.

SYSTEM ARCHITECTURE

The NexCart system follows a three-tier architecture consisting of the NexCart system is designed using a three-tier architecture that separates the application into distinct layers to improve scalability, maintainability, and performance. This architecture ensures that each layer handles specific responsibilities, making the system more organized and easier to develop and manage than a monolithic architecture. The three layers are the presentation, application, and data layers [6].

Presentation Layer

The presentation layer is the frontend of the application and acts as an interface between the user and the system. It is responsible for displaying content, collecting user input, and providing an interactive experience. This layer allows users to smoothly interact with the platform, explore available products, select items, add them to their carts, and finalize their purchases with ease.

The design of this layer focuses on user experience (UX) and responsive design, ensuring compatibility across different devices, such as desktops, tablets, and smartphones. It also handles client-side validation to reduce errors before sending the data to the backend [7].

Technologies Used

- *HTML*: Defines the structure of web pages.
- *CSS*: Provides styling, layout, and responsive design.
- *JavaScript*: Enables dynamic content, user interaction, and real-time updates.

JavaScript is also used to implement features such as form validation, dynamic product rendering, cart updates, and API communication with backend services.

Application Layer

The application layer (also known as the business logic layer) processes user requests and controls the overall functionality of the system. It acts as a bridge between the presentation and data layers [8].

This layer handles core operations such as:

- User authentication and authorization
- Product management
- Cart operations and order processing
- Business rules and validations

This ensures that all operations follow the defined logic before interacting with the database. The modular structure of this layer allows for the easy integration of additional features, such as recommendation systems, payment gateways, and AI-based enhancements.

Data Layer

The data layer handles storage, retrieval, and organization of application data, ensuring that it remains consistent, secure, and easily accessible when needed.

This layer includes:

- User information (login credentials, profiles)
- Product details (name, price, description, images)
- Order and transaction records

The system can use databases such as MySQL, Firebase, or JSON-based storage, depending on the implementation requirements. Proper data-handling techniques were applied to ensure data integrity and reliability.

Overall Architecture Benefits

The three-tier architecture provides several advantages (Figure 1)

- *Scalability*: Each layer can be scaled independently.
- *Maintainability*: Individual layers can be updated or modified more easily without affecting the rest of the system.
- *Reusability*: Components can be utilized repeatedly throughout the application.
- *Security*: Sensitive operations are handled in controlled layers.

WORKING PRINCIPLE

The NexCart e-commerce platform operates by providing an interactive online interface, where users can browse products, select desired items, and complete purchases through a well-structured web-based system. The platform's working principle is based on the coordinated interaction between the user interface, backend processing, and database management, ensuring smooth and efficient system functionality [9].

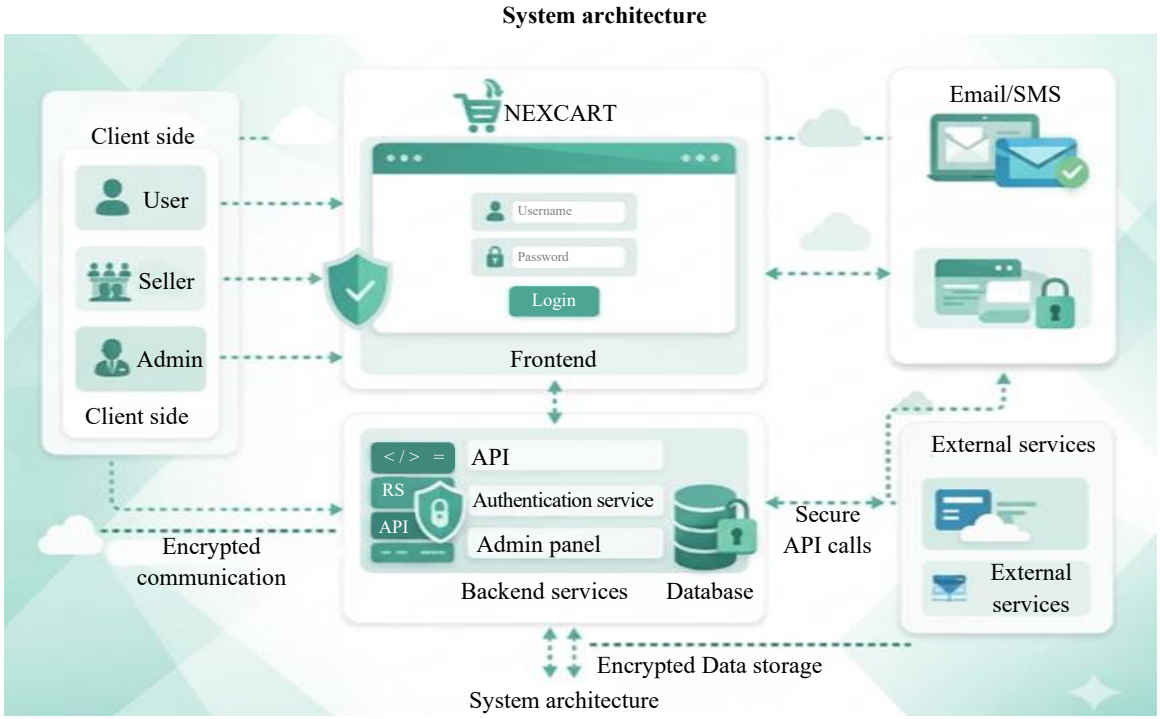

Figure 1. System architecture.

When a user visits the NexCart website, the presentation layer, developed using HTML, CSS, and JavaScript, displays the available products and categories in an organized and user-friendly manner. Users can register a new account or log in to access personalized services, such as saved preferences and order history. After successful authentication, users can browse products, search for specific items, and view detailed information, including images, prices, and descriptions [10].

When a user selects a product and adds it to their shopping cart, the request is forwarded to the application layer, which is implemented using Python. This layer handles user sessions, processes cart updates, and validates the user actions. During checkout, the system verifies the order details and processes the transaction. Finally, the data layer stores all relevant information, including user details, product data, and order records, ensuring data consistency, security, and efficient retrieval for future operations [11].

IMPLEMENTATION AND RESULTS

The NexCart e-commerce platform was implemented using HTML, CSS, and JavaScript for the frontend, and Python for the backend processing. The system allows users to register and log in securely to access the platform [12]. Users can browse products, view product details, add items to the cart, and place orders easily (Figure 2).

System Implementation Overview

The NexCart system was implemented using a full-stack web development approach, integrating frontend, backend, and database technologies to ensure a seamless and efficient e-commerce experience. The frontend was developed using HTML, CSS, and JavaScript to provide an interactive and responsive user interface. The backend was built using Python with the Flask framework, which handles routing, business logic, and API communication. MySQL was used as the database to securely store user data, product details, and transaction records.

The implementation follows a modular architecture, where each component, such as authentication, product management, cart operations, and order processing, is developed independently and integrated through RESTful APIs.

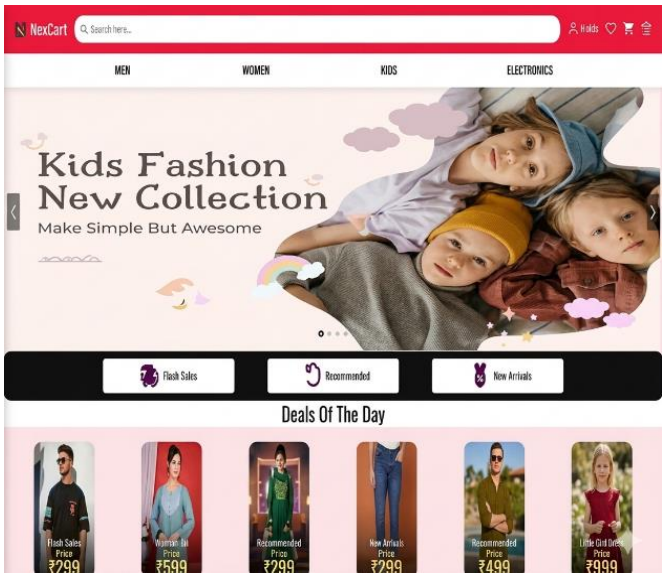

Figure 2. Main home page.

Figure 3. Product page.

User Authentication Implementation

The user authentication module was implemented using secure techniques to ensure data protection and system integrity. User registration includes input validation and password encryption using the Bcrypt hashing algorithm. The login functionality verifies user credentials and maintains session management.

Additionally, an OTP-based password recovery system is implemented using email services, where a time-limited OTP is generated and verified before allowing a password reset. This enhances the security of the user accounts (Figure 3).

Product Management Implementation

The product module retrieves product data from the MySQL database and dynamically displays it on the frontend. Sellers can add, update, and delete products through dedicated APIs, whereas the administrator verifies and approves products before they are visible to users.

Cart and Order Management Implementation

The cart module allows users to add products, update quantities, and dynamically remove items. Cart data is managed using JavaScript on the client side and synchronized with the backend during checkout (Figure 4).

Payment Processing Implementation

The payment module simulates secure transaction handling by validating the payment details and storing the transaction records in the database. After a successful payment confirmation, the system updates the order status and generates an order confirmation response, as shown in Figure 5.

Admin and Seller Module Implementation

The admin panel is implemented to provide complete control over the system operations. The admin functionalities include the following:

- Managing users and sellers
- Approving or rejecting products
- Monitoring orders and transactions

The seller module allows vendors to register, manage their products, and track their earnings through a wallet system. Commission logic is implemented, where a percentage of each product sale is deducted and stored.

Backend API Implementation

The backend was implemented using Flask REST APIs that handled all system operations, such as authentication, product handling, cart operations, and order processing. These APIs ensure smooth communication between the frontend and the database.

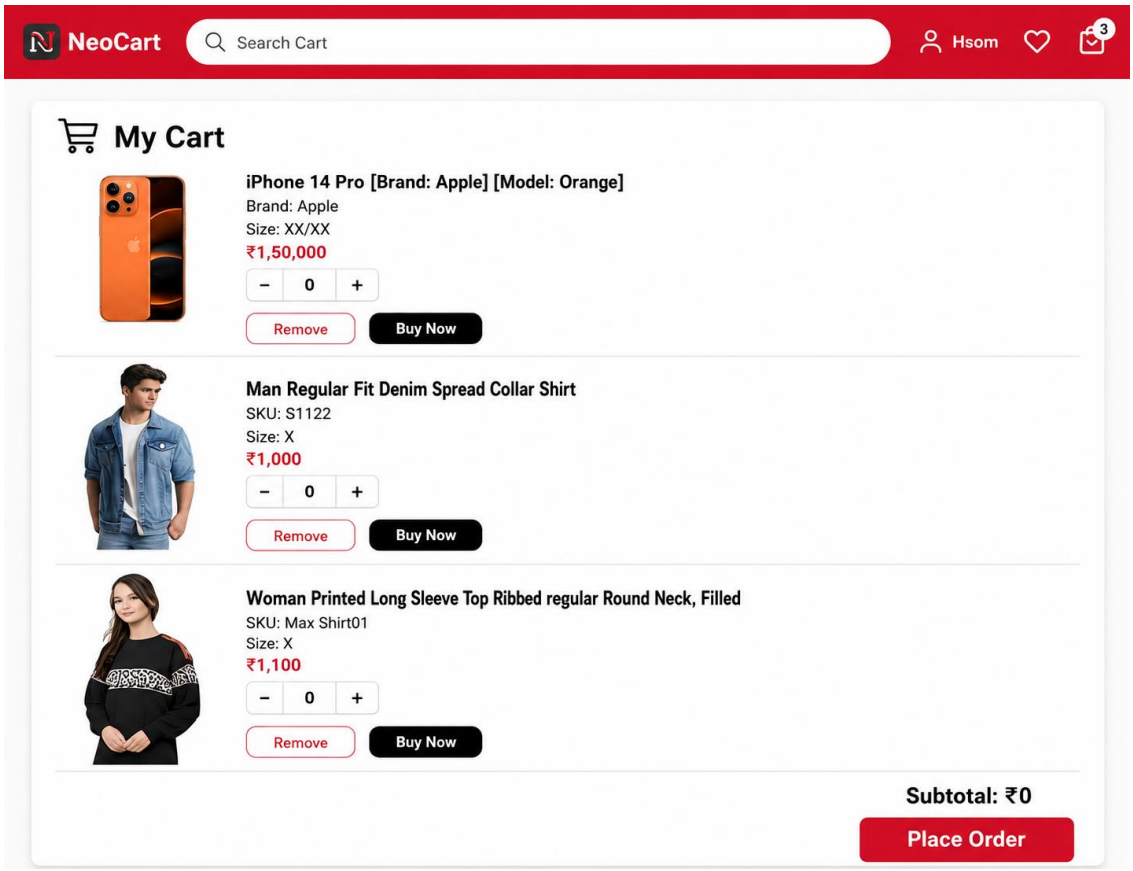

Figure 4. Cart page.

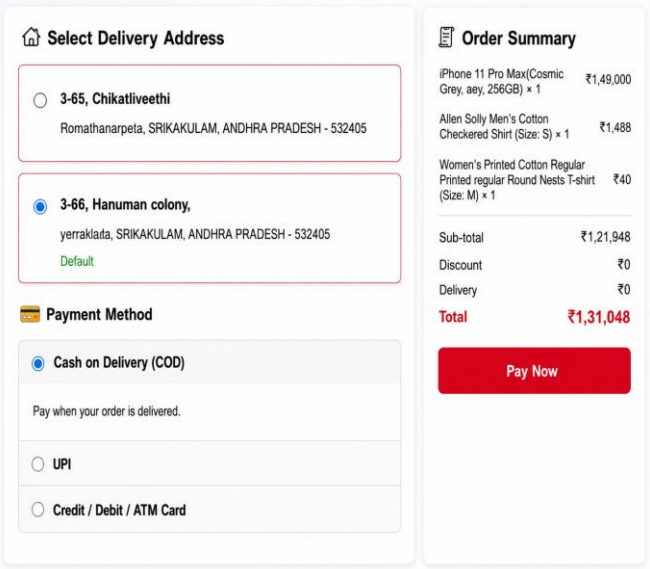

Figure 5. Payment page.

The backend code provided demonstrates:

- Secure database connectivity
- Structured routing system
- JSON-based data exchange
- Error handling and validation

This implementation ensures scalability, maintainability, and efficient system performance.

The platform also includes an admin dashboard that allows administrators to add, update, and delete products and monitor customer orders. The database stores information related to users, products, and transactions in the system. The system was tested to ensure the proper functionality of all modules.

The results show that the platform successfully performs operations such as user authentication, product management, and order processing. Overall, the NexCart system provides a reliable and efficient online shopping experience.

APPLICATIONS

The NexCart e-commerce platform can be used in various fields to support the online buying and selling of products. It is useful for online retail businesses to sell products over the internet. Small and medium enterprises can use this system to expand their businesses digitally. It can also be used in online marketplaces where multiple products are available to customers. The platform helps businesses efficiently manage products, orders, and customer information. Additionally, it supports secure and convenient online shopping for users from any location.

The NexCart e-commerce platform can be used in various fields to support the efficient online buying and selling of products. It is highly beneficial for online retail businesses, enabling them to showcase and sell their products to a wider audience through the internet without the limitations of physical stores. Small and medium enterprises (SMEs) can leverage this system to expand their business digitally, reduce operational costs, and reach customers beyond their geographical boundaries.

In addition, NexCart can be implemented in multi-vendor online marketplaces, where different sellers can list and manage their products on a single platform, providing customers with a wide variety of choices. The system simplifies business operations by offering features such as product management, order tracking, and customer data handling, which improve overall efficiency and decision-making.

The platform is also suitable for startups and educational projects because it provides a clear understanding of e-commerce workflows and system design. NexCart also supports secure and convenient online transactions, ensuring data protection and user trust. Its scalable and modular architecture allows for future enhancements, such as integration with payment gateways, AI-based product recommendations, and mobile applications, making it adaptable to evolving business needs and technological advancements.

CONCLUSION AND FUTURE SCOPE

The NexCart e-commerce platform provides an efficient, reliable, and user-friendly solution for modern online shopping needs. The system enables users to easily browse products, add items to their shopping carts, and place orders through a responsive and intuitive web interface. Its design ensures smooth navigation and a seamless user experience across different devices. Additionally, the platform includes a comprehensive admin dashboard that allows administrators to manage products, monitor user activity, and handle orders efficiently, thereby improving the overall system control and business operations.

The platform enhances convenience for customers by providing quick access to a wide range of products while simplifying inventory and order management for businesses. Its modular and scalable architecture makes it suitable for small-scale applications and growing enterprises. By using standard web technologies, the system ensures flexibility, ease of maintenance, and cost-effective deployment.

In the future, the NexCart system can be further enhanced by integrating secure online payment gateways, enabling real-time transaction processing, and expanding support for mobile applications for better accessibility. Advanced features, such as AI-based product recommendations, personalized user experiences, and chatbots for customer support, can be incorporated to improve engagement. Additionally, implementing stronger security mechanisms, such as data encryption and multi-factor authentication, enhances user trust and system reliability. These improvements will make NexCart a more robust, intelligent, and scalable e-commerce solution capable of adapting to evolving technological trends and user demands.

REFERENCES

1. Ahi AA, Sinkovics N, Sinkovics RR. E-commerce policy and the global economy: a path to more inclusive development? Manag Int Rev. 2023;63:27–56. doi:10.1007/s11575-022-00490-1. PubMed PMID: 36345510.
2. Abar H, Alkan Ö. What factors influence the use of electronic commerce? A case in Turkey. In: Information Resources Management Association, editor. Research Anthology on E-Commerce Adoption, Models, and Applications for Modern Business. Hershey (PA): IGI Global Scientific Publishing; 2021. p. 1893–1910. doi:10.4018/978-1-7998-8957-1.ch094.
3. Duckett J. HTML and CSS: Design and Build Websites. Indianapolis (IN): John Wiley & Sons; 2011.
4. Flanagan D. JavaScript: The Definitive Guide. Sebastopol (CA): O'Reilly Media; 2011.
5. Lutz M. Learning Python: Powerful Object-Oriented Programming. 6th ed. Sebastopol (CA): O'Reilly Media; 2025.
6. Horstmann C. Object-Oriented Design and Patterns. 2nd ed. Hoboken (NJ): John Wiley & Sons; 2009.
7. Montulli L, inventor; Netscape Communications Corp, assignee. Persistent client state in a hypertext transfer protocol based client-server system. United States patent US 6592. 2000.
8. Sommerville I, Sawyer P. Viewpoints: principles, problems and a practical approach to requirements engineering. Ann Softw Eng. 1997;3(1):101–130. doi:10.1023/A:1018946223345.
9. Pressman RS. Web Engineering: A Practitioner's Approach. New York (NY): McGraw-Hill; 2008.
10. Coronel C, Morris S, Rob P. Database Systems: Design, Implementation, and Management. 9th ed. Boston (MA): Cengage Learning; 2011.
11. Appleton B. Patterns and software: essential concepts and terminology. Object Mag Online. 1997;3(5):20–25.
12. Gourley D, Totty B. HTTP: The Definitive Guide. Sebastopol (CA): O'Reilly Media; 2002.

