QUICKCART – Retail Checkout Application

Bankuru Vamsi[1], Basa Sai Durga Harshith[2], N. Srinivasan[3]

[1,2,3] Department of Computer Science and Engineering Sathyabama Institute of Science and Technology, Chennai, India.

vamsikrishnabankuru2005@gmail.com*

Abstract. In today's fast-paced and competitive digital economy, small and medium retail businesses urgently need a reliable, scalable, and efficient platform to connect seamlessly with wholesalers and manufacturers. Traditional B2B commerce is often manual, inefficient, and lacks operational transparency. To fill this critical gap, we introduce QuickKart, a comprehensive and innovative B2B wholesale e-commerce platform that digitalizes, automates, and streamlines the wholesale ordering process. QuickKart enables vendors to upload, update, and manage their product inventories while allowing buyers (retailers) to conveniently place bulk orders with greater efficiency. The platform includes key features such as user authentication, role-based access (Admin, Vendor, Buyer), product management, a shopping cart with minimum order validation, invoice generation, and order tracking. Vendors can upload product data via forms or Excel files, and buyers can download auto-generated invoice PDFs after confirming their orders. The backend is built with Spring Boot, providing secure RESTful APIs to handle business logic, database operations, and file management. The frontend, developed with React.js, offers a responsive, user-friendly interface with protected routes, dynamic dashboards, and interactive analytics charts. Additional modules, like an admin control panel, sales trend analysis, and real-time notifications (via WebSocket or polling), enhance the platform's robustness and scalability. Optional integrations, such as payment gateway sandbox, product ratings, and multi-vendor chat, further improve its features. Through this project, we aim to simulate a real-world wholesale commerce system and gain practical experience with full-stack development, REST API design, authentication, file handling, charting, and DevOps practices.

Keywords: QuickCart, B2B e-commerce, Retail checkout application, Wholesale management, Inventory management, Bulk order processing, Invoice generation, Role-based access control, Spring Boot

1 Introduction

The retail sector is undergoing a major transformation with the adoption of digital platforms that are reshaping the way wholesale and retail businesses operate. Ecommerce has emerged as the backbone of this transformation by enabling efficiency, transparency, and scalability across business operations [1]. While Business-to Consumer (B2C) platforms have advanced rapidly, the Business-to-Business (B2B) wholesale segment still faces significant challenges, especially for small and medium enterprises that continue to rely on traditional methods such as phone calls, paper catalogs, and direct store visits for ordering. These processes are time-consuming, prone to errors, and lack

R. Vasanth Kumar Mehta et al. (eds.), *Proceedings of the International Conference on Intelligent Systems for a Sustainable Future (ISSF 2026)*, Atlantis Highlights in Intelligent Systems 16,
https://doi.org/10.2991/978-94-6239-693-7_113

real-time visibility, making it difficult for retailers to compete in the current digital economy.

QuickCart is developed to address these challenges by providing a comprehensive wholesale e-commerce solution that digitizes the ordering process. The platform allows vendors to upload and manage their product inventories while enabling buyers to place bulk orders quickly and efficiently. It includes features such as secure authentication, role-based access, shopping cart management with validation, automated invoice generation, order tracking, and real-time updates. The system also supports vendor uploads through forms or Excel sheets and provides invoice downloads in PDF format for retailers after confirming their orders.

The platform leverages modern technologies such as React.js for building a responsive and interactive user interface [8], Node.js with Express or optionally Spring Boot for backend API development, and MongoDB or Firebase for database operations and real-time synchronization. To ensure secure financial transactions, the system integrates payment gateways like Razorpay or Stripe [6]. Additional modules such as admin dashboards, sales trend analysis, and real-time notifications [9] enhance the scalability and usability of the platform.The motivation behind QuickCart is to create a reliable, secure, and scalable B2B ecommerce application [10] that simplifies wholesale transactions for small and medium businesses. By bridging the gap between wholesalers and retailers, the platform not only provides business benefits but also serves as a learning opportunity to apply fullstack development concepts in a real-world scenario. The system demonstrates the 2 practical applications of technologies such as authentication, database integration, REST API design, file handling, payment processing, and DevOps practices.

2 Literature Survey

The expansion of e-commerce has been widely examined, with research showing that digital platforms enhance efficiency, reduce human errors, and provide scalability for businesses. Most studies, however, emphasize Business-to Consumer platforms, highlighting personalization, secure payments, and delivery tracking, while Business-to-Business wholesale systems remain less explored despite their importance in supply chains. Literature indicates that small and medium enterprises often struggle with existing digital platforms due to high costs and technical complexity. Researchers propose cloud computing, distributed databases, and modular architectures, but these solutions are often difficult for SMEs to adopt. Studies on supply chain management stress the value of real-time inventory synchronization, while user experience research emphasizes responsive design and mobile accessibility for successful adoption [2]. Security challenges such as fraud and data breaches are another recurring theme. Scholars recommend multi-factor authentication, token-based APIs, and secure payment gateways to build trust [3]. Research on databases highlights the efficiency of NoSQL systems like MongoDB [4] and real-time solutions like Firebase [7] in managing concurrent users and large inventories. Recent works also highlight analytics and artificial intelligence, including predictive sales forecasting, product recommendations, and automated invoicing. These features improve decision making and system

performance but are rarely implemented in SME-focused wholesale platforms. Overall, literature points to a gap in affordable, secure, and automation-driven systems, which provides the foundation for QuickCart to serve as a reliable and scalable solution tailored to SMEs.Overall, QuickCart represents a robust and practical solution for digitizing wholesale commerce. It ensures transparency, efficiency, and ease of use for vendors, retailers, and administrators, while providing a secure, scalable, and user-friendly platform that meets the demands of the digital economy. This project also highlights the role of technology in solving real-world business challenges and contributes to building industry-ready skills in full-stack software engineeringIn addition to addressing current wholesale challenges, QuickCart has been designed with scalability in mind so that it can be extended to support multi-vendor systems, product rating features, AI-based product recommendations, and predictive sales forecasting in the future. This ensures that the platform is not only a solution for present requirements but also adaptable to future advancements in technology and business needs. The system also focuses on providing a responsive design to support multiple devices, ensuring accessibility for users across desktops, tablets, and smartphones. From an academic perspective, the project provides an opportunity to apply theoretical knowledge in practical software development, covering important areas such as requirement analysis, system design, database modeling, and testing methodologies. It encourages problem-solving skills, teamwork, and exposure to modern software engineering practices that are highly valued in industry

By developing QuickCart, the project team aims to demonstrate how a well-designed digital platform can transform traditional business operations into efficient, reliable, and technology-driven solutions.

2.1 Inferences From Literature Survey

The review of literature highlights that although e-commerce has made remarkable progress in the retail sector, the wholesale domain still lacks effective digital solutions. Small and medium enterprises face major barriers such as high costs, limited technical expertise, and complex system requirements. This shows the need for platforms that are affordable, lightweight, and tailored to the operational realities of SMEs. Unlike consumer platforms that emphasize personalization, wholesale systems must prioritize efficiency in bulk ordering, accurate stock management, and automated invoice generation, yet these requirements are often neglected in existing research. Security emerges as another dominant theme. Scholars recommend multi-factor authentication, encrypted APIs, and trusted gateways to ensure transaction reliability.

These findings emphasize that any wholesale solution, including QuickCart, must embed security deeply into its architecture. Literature also stresses the relevance of cloud computing and distributed databases, which provide scalability and fault tolerance, allowing systems to handle heavy user traffic while remaining stable and efficient.User experience is identified as a decisive factor for adoption. If platforms are not mobile-friendly, intuitive, and responsive, SMEs are reluctant to embrace them. For this reason, QuickCart must integrate seamless interfaces supported by modern frameworks like React.js. Another inference is the rising role of analytics and artificial intelligence.

Features such as predictive sales forecasting and recommendation engines are increasingly valuable but are rarely integrated into SME-oriented wholesale platforms.

This gap provides scope for QuickCart to introduce AI-driven features in the future. Finally, the absence of cost-effective platforms, limited automation in invoicing, and inadequate real-time update mechanisms are recurring gaps in literature. These inferences strongly support the need for QuickCart as a secure, scalable, and userfriendly solution capable of addressing the shortcomings of existing systems and meeting the evolving demands of wholesale commerce.

2.2 Open problems in existing system

The survey of existing wholesale e-commerce platforms highlights several critical limitations that restrict their adoption and effectiveness for SMEs:

1. High Cost and Complexity Most current platforms are designed for large enterprises, requiring high investment and technical expertise. SMEs struggle to adopt such systems due to resource constraints and complicated setup processes.
2. Lack of Bulk Order and Invoice Automation Existing systems focus mainly on B2C needs rather than wholesale requirements. Limited support for bulk ordering, automated invoice generation, and streamlined billing causes inefficiency [5].
3. Absence of Real-Time Inventory Synchronization Many platforms fail to provide accurate, real-time stock updates, leading to order mismatches and delays. Vendor uploads are often manual, slow, and prone to errors without proper validation mechanisms.
4. Security and Transaction Risks Weak authentication and poor payment gateway integration expose retailers and vendors to risks of fraud and data breaches. Without secure systems, user trust and adoption remain low.
5. Poor User Experience and Limited Adoption Interfaces are often complex, non-intuitive, and not optimized for mobile devices. Retailers hesitate to shift from traditional methods when digital platforms are not user-friendly

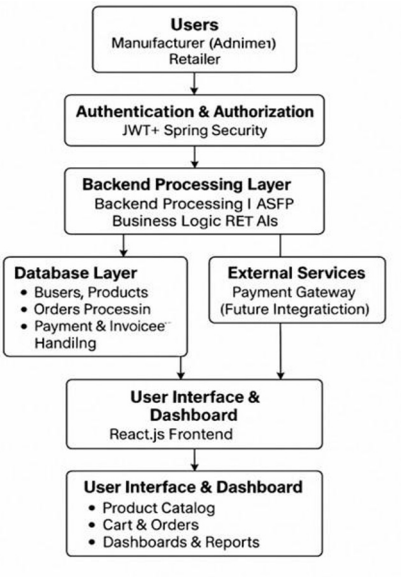
3 System Architecture

The proposed QuickCart – Retail Checkout Application requires a clear definition of functional and non-functional requirements to ensure that it serves as a secure, efficient, and scalable solution for wholesale trade. Functionally, the system must enable vendors to register, upload, and manage their product catalogs, while providing retailers with the ability to place bulk orders through a validated shopping cart system. Automated invoice generation in PDF format after order confirmation, real-time inventory synchronization to prevent stock mismatches, and order tracking are essential capabilities. Role-based authentication is required to differentiate access for vendors, retailers, and administrators, with the admin panel responsible for sales monitoring, user management, and reporting.

Performance requirements emphasize minimal latency, ensuring that order processing and system responses occur in under one second, while also supporting scalability to handle large user volumes and transaction loads. Security remains a primary concern, requiring token-based authentication, encrypted APIs, and trusted payment gateway integration to protect sensitive financial and user data.

From a non-functional perspective, the system must be designed with usability and reliability as key priorities. The platform should be lightweight enough for small and medium enterprises to adopt easily, while also supporting future expansion through a modular architecture. The design will incorporate React.js for the front end, Node.js with Express or Spring Boot for backend APIs, and MongoDB or Firebase for database operations, with real-time updates enabled through WebSockets or Firebase synchronization. These choices ensure maintainability and flexibility, allowing the integration of additional features such as predictive analytics, recommendation engines, and sales forecasting in later versions.

Fig. 1. System Architecture of the Proposed Retail System



A. System Architecture Description

The User Interface (UI) module serves as the primary interaction point between the user and the system. A web-based interface is implemented with separate pages for home, user registration, login, and emotion prediction. After successful authentication, the user is granted access to the speech emotion recognition panel. The interface allows the user to either record speech using a microphone in real time or upload a pre- recorded audio file. The UI module also displays the predicted emotion, class probabilities, and additional signal information such as recording duration and loudness.

B. User Interface Module

The User Interface (UI) module acts as the primary interaction point between users and the QuickCart system. A web-based interface is developed using React.js, providing separate views for user registration, login, dashboards, product catalog, cart, and order tracking. Based on the authenticated user role (Manufacturer or Retailer), the interface dynamically displays role-specific features. Retailers can browse products, add items to the cart, place bulk orders, and track order status, while manufacturers can manage products, inventory, orders, and invoices. The UI module communicates with the backend through secure REST APIs and displays real-time updates, notifications, and validation messages to enhance user experience.

C. Access Control and Authorization Module

The Authentication and Role Management module ensures secure access to the system. This module is implemented using JWT (JSON Web Token) authentication and Spring Security. It handles user registration, login, token generation, and role-based authorization. Each request from the frontend is validated using JWT tokens to ensure that only authenticated and authorized users can access protected APIs. This module enforces strict access control, preventing unauthorized operations and safeguarding sensitive business data.

D. Backend Processing Module

The Backend Processing module forms the core of the QuickCart application and is implemented using Spring Boot. This module handles all REST API requests received from the frontend and applies the required business logic. It manages product creation and updates, validates inventory availability before approving orders, processes order status transitions (Pending, Approved, Shipped, Delivered), generates invoices, and records payment information. Proper validation, exception handling, and logging mechanisms are implemented to ensure reliability, scalability, and maintainability of the system.

E. Order & Stock Management Module

This module is responsible for managing bulk orders and maintaining accurate inventory levels. When a retailer places an order, the module verifies product availability and reserves stock upon approval by the manufacturer. Inventory levels are automatically updated to prevent over-ordering. The module also tracks the complete order lifecycle and ensures consistency between orders and stock data. Any discrepancies or insufficient stock scenarios are handled through appropriate validations and error responses.

F. Third-Party Integration Layer (Future Enhancements)

This optional module is designed to support integration with third-party services such as payment gateways (Razorpay or Stripe), email notification systems, and analytics tools. These services enhance the overall functionality of the application by enabling automated payments, notifications, and reporting features. The modular design allows easy integration of these services in future enhancements without affecting existing system components.

G. Secure Communication Framework

This module ensures secure communication between the frontend and backend layers. All API interactions occur over secure channels and are protected using JWT-based authentication. Role-based access control and input validation mechanisms are enforced to prevent unauthorized access, data tampering, and common security vulnerabilities.

4 Result and Discussion

The QuickCart application was successfully designed, implemented, and tested as a B2B retail platform connecting manufacturers and retailers. The system effectively supports user authentication, role-based access control, product management, bulk order processing, inventory tracking, and order status monitoring. The application demonstrated stable performance during functional testing and ensured secure data handling through JWT-based authentication and Spring Security.

The React-based user interface provided a responsive and intuitive experience for both manufacturers and retailers. Retailers were able to browse the product catalog, place bulk orders, and track order status without difficulty. Manufacturers could efficiently manage products, monitor inventory levels, approve or reject incoming orders, and generate invoices. All backend operations were processed through well-defined REST APIs developed using Spring Boot, ensuring smooth communication between the frontend and backend layers.

The MySQL database effectively stored and managed all application data, including users, products, orders, order items, payments, and invoices. Referential integrity was maintained through proper relational mappings, and inventory levels were accurately updated upon order approval. Error handling and validation mechanisms ensured that invalid operations, such as placing orders with insufficient stock, were correctly prevented.

The results indicate that the QuickCart system successfully addresses the challenges associated with traditional manual B2B order management processes. By digitizing product listings, bulk ordering, and inventory management, the application significantly reduces human errors and improves operational efficiency. The role-based access control mechanism ensures that manufacturers and retailers can only perform authorized actions, thereby enhancing system security and data protection. The modular architecture adopted in this project contributed to improved scalability and maintainability. The separation of concerns between the user interface, backend processing, and database layers allows individual components to be enhanced or scaled independently. The use of RESTful APIs enables seamless integration between the frontend and backend, making the system adaptable to future extensions such as mobile applications or third-party service integrations.

Overall, the QuickCart application demonstrates a practical and scalable solution for B2B retail management. The project successfully meets its objectives by providing a secure, efficient, and user-friendly platform that can be further enhanced with advanced analytics, automated notifications, and real-time reporting in future iterations.

5 Conclusion

This project focused on the design and development of QuickCart, a B2B retail management application aimed at streamlining interactions between manufacturers and retailers. The system was built to address common challenges such as manual order handling, lack of inventory visibility, and inefficient communication within the supply chain.

By adopting a structured development approach and a layered system architecture, the application successfully integrates frontend, backend, and database components into a unified platform. The use of React.js enabled the creation of an interactive and responsive user interface, while Spring Boot provided a robust framework for implementing business logic and secure RESTful services. MySQL was used to ensure reliable and structured data storage.

The project also provided valuable practical exposure to full-stack development concepts, including database design, API development, authentication mechanisms, and role-based access control. Emphasis on modularity and clean separation of concerns improved the maintainability and extensibility of the system.

In conclusion, QuickCart serves as a well-structured and practical B2B retail solution that demonstrates the effective application of modern web technologies to solve real-world business problems. The system lays a strong foundation for future enhancements and reflects the successful achievement of the project's design and learning objectives

QuickCart lays a strong foundation for future enhancements such as payment gateway integration, AI-driven demand prediction, analytics dashboards, and real-time notifications. These potential extensions position the platform not only as an academic prototype but as a scalable industry-ready solution for digital wholesale commerce.

In conclusion, the QuickCart system demonstrates how modern web technologies can be effectively applied to transform traditional supply chain interactions into secure, efficient, and technology-driven processes, thereby improving productivity and supporting the digital evolution of small and medium retail businesses.

References

1. ACM Digital Library, "Progressive Web Applications in E-commerce: Enhancing User Retention and Performance", 2019
2. Elsevier, "User Behavior and Interface Design in Online Stores: A Study on Retail Adoption", 2022.
3. IEEE, "A Framework for Secure Online Shopping with Encrypted Transactions", 2020.
4. IEEE Access, "MongoDB vs. SQL Databases for Scalable E-commerce Platforms", 2019
5. International Journal of Computer Theory and Engineering (IJCTE), "Order Management and Workflow Automation in Online Retail Systems", 2021.
6. International Journal of Engineering Research & Technology (IJERT), "Integration of Payment Gateways (Razorpay/Stripe) in Online Applications", 2021.

7. International Journal of Scientific & Engineering Research (IJSER), “Firebase as a Real-Time Backend Solution for E-commerce Platforms”, 2020.
8. International Journal of Scientific Research in Computer Science and Engineering (IJSRCSE), “Single Page Applications using React.js for Retail Platforms”, 2020
9. Springer Proceedings in Business Information Systems, “The Role of Admin Dashboards and Analytics in E-commerce Decision Making”, 2020.
10. Springer, “Design and Development of Scalable E-commerce Web Applications”, 2021