Impact Factor (QJIF): 8.47 **E-ISSN: 2583-1615, P-ISSN: 3049-3498**

International Journal of Research in Academic World

Received: 20/January/2026 **IJRAW:** 2026; 5(3):41-43 **Accepted:** 27/February/2026

Designing and Implementing a Real-Time Inventory Analytics Platform Using Modern Full-Stack Web Architecture

[1]M Akila and [*2]T Rajeshwari

[1]Student of M.Sc., Department of Computer Science, Bon Secours College for Women (Autonomous), Thanjavur, Tamil Nadu, India.

[*2]Assistant Professor, Department of Computer Science, Bon Secours College for Women (Autonomous), Thanjavur, Tamil Nadu, India.

Abstract

Inventory management is a critical operational function for organizations that handle large volumes of products such as mobile phones, books, and electronic devices. Efficient management of stock levels, financial investment, and sales performance is essential for maintaining profitability and operational stability. Traditional inventory management systems typically rely on manual record keeping or static databases that do not provide real-time insights or automated analytics. As a result, businesses often experience inaccurate inventory records, delayed decision-making, and financial miscalculations. This research presents the design and implementation of a Real-Time Inventory Analytics Platform using modern full-stack web architecture. The proposed system integrates React TypeScript for frontend development, NestJS for backend services, and PostgreSQL as the relational database management system. The platform enables real-time stock monitoring, automated financial calculations, and lifecycle-based classification of products into active stock, old stock, and sold-out categories.

The system also includes secure user authentication, dynamic dashboard analytics, and efficient search functionality. Financial indicators such as investment amount, profit percentage, and selling price are automatically calculated using database queries and backend processing. Experimental evaluation demonstrates that the proposed system significantly improves data accuracy, operational efficiency, and response time compared to traditional inventory management systems. The modular architecture ensures scalability, maintainability, and adaptability for future enhancements such as cloud deployment and artificial intelligence-based stock forecasting.

Keywords: Real-Time Inventory Management, Full-Stack Web Architecture, React TypeScript, NestJS Framework, PostgreSQL Database.

1. Introduction

Inventory management systems play a vital role in modern business environments where organizations manage thousands of products and transactions daily. These systems help maintain accurate records of stock levels, product details, pricing, and sales performance. Efficient inventory tracking ensures smooth business operations, prevents stock shortages, and improves financial transparency. In many small and medium-sized businesses, inventory records are still maintained manually or using spreadsheet-based solutions. Such systems are prone to human errors, data inconsistencies, and delayed reporting. Without automated analytics and real-time updates, managers often struggle to monitor stock performance and financial investments effectively.

With the rapid advancement of web technologies, modern full-stack applications provide powerful solutions for developing scalable and efficient business management systems. Full-stack architecture integrates frontend interfaces, backend services, and database management systems into a unified platform that enables real-time communication and data synchronization.

This research proposes a real-time inventory analytics platform designed to address the limitations of traditional inventory systems. The proposed system integrates modern technologies such as React TypeScript, NestJS, and PostgreSQL to create a scalable and modular architecture capable of handling complex inventory operations.

The primary objectives of this study include:

i). Designing a scalable full-stack architecture for inventory management
ii). Implementing real-time analytics for stock monitoring
iii). Automating financial calculations related to investment and profit
iv). Maintaining historical records of old stock and sold-out products
v). Providing a secure authentication mechanism for users

By combining modern web frameworks with relational database management, the proposed system aims to improve operational efficiency and support data-driven decision-making in inventory management.

2. Review of Literature

Inventory management has been widely studied in operations

***Corresponding Author:** T Rajeshwari

research, supply chain management, and information systems. Early research focused primarily on mathematical models for optimizing stock levels and minimizing inventory costs. Silver *et al*. introduced fundamental inventory control principles such as safety stock calculation and economic order quantity models. Their work highlighted the importance of accurate demand forecasting in maintaining optimal inventory levels. Chopra and Meindl emphasized the integration of information systems within supply chain management. They argued that efficient information flow between business operations and inventory systems is essential for improving supply chain performance. With the emergence of web technologies, researchers began exploring the use of distributed systems and service-oriented architectures in inventory management platforms. RESTful web services, introduced by Fielding, provided a scalable approach for communication between frontend and backend systems. Recent studies also focus on microservices architecture for large-scale applications. Microservices enable the separation of system functionalities into independent modules, allowing developers to maintain and deploy system components independently. Database research has also contributed significantly to inventory management systems. Relational database management systems such as PostgreSQL provide advanced features including indexing, foreign key constraints, and transaction management that ensure data consistency and performance.

Swarm Optimization with Neural Networks for Effective Classification Techniques" by K. Kalyani (2021) introduces a hybrid EHBMO-NN model, combining Extended Honey Bee Mating Optimization with Artificial Neural Networks to improve classification accuracy and reduce training time. It uses HBMO to select optimal weights for neural network hidden layers, outperforming conventional methods on benchmark datasets. The accurate cancer classification is very important task for cancer treatment. Recently the informative genes are identified from the thousands of genes for correct cancer classification. The collection of microscopic Deoxyribo Nucleic Acid (DNA) microarray is attached in the solid surface. In this study, DNA microarray data is used for cancer classification. The accurate cancer classification is very important task for cancer treatment. Recently the informative genes are identified from the thousands of genes for correct cancer classification. The collection of microscopic Deoxyribo Nucleic Acid (DNA) microarray is attached in the solid surface. In this study, DNA microarray data is used for cancer classification (6).

3. Existing System

The existing inventory management systems used in many small and medium businesses rely on traditional methods such as manual record keeping or basic database applications. Although these systems provide basic functionality for storing product information, they suffer from several limitations that reduce efficiency and accuracy in inventory operations.

The major limitations of existing systems are listed below:

i). **Manual Inventory Tracking:** Many traditional systems depend on manual entry of product data and stock updates. This increases the chances of human errors and inconsistent records.

ii). **Lack of Real-Time Data Updates:** Existing systems do not provide real-time monitoring of inventory levels. Updates often occur only after manual refresh or batch processing.

iii). **Limited Data Analytics:** Most systems only store product information without providing analytical insights such as total investment, profit percentage, or sales trends.

iv). **No Product Lifecycle Management:** Traditional inventory systems do not categorize products into different lifecycle stages such as active stock, old stock, or sold-out items.

v). **Poor User Interface:** Many older systems have complex or outdated interfaces that make it difficult for users to navigate and manage inventory data efficiently.

vi). **Limited Search and Filtering Features:** Existing systems often lack advanced search capabilities, making it difficult to quickly locate specific products or transactions.

vii). **Weak Security Mechanisms:** Basic authentication methods used in traditional systems may expose user data and business information to unauthorized access.

4. Proposed System

The proposed system introduces a Real-Time Inventory Analytics Platform designed using modern full-stack web architecture. The system improves traditional inventory management by providing real-time monitoring, automated analytics, and scalable system architecture.

The key features of the proposed system are as follows:

i). **Real-Time Inventory Monitoring:** The system continuously updates product stock levels whenever new products are added, edited, sold, or moved to old stock. This ensures that users always view the most recent inventory data.

ii). **Full-Stack Web Architecture:** The platform is developed using a modern technology stack including React TypeScript for the frontend, NestJS for the backend, and PostgreSQL for the database, enabling scalable and efficient system performance.

iii). **Automated Financial Analytics:** The system automatically calculates financial metrics such as total investment, profit amount, and profit percentage, eliminating manual calculations and reducing errors.

iv). **User Authentication and Security:** Secure login and registration features are implemented to protect user data. Input validation and authentication mechanisms ensure that only authorized users can access the system.

v). **Product Management Module:** Users can add, edit, delete, and view product information. Each product contains attributes such as product name, category, price, quantity, and date.

vi). **Dashboard Analytics:** The dashboard provides a comprehensive overview of inventory statistics including:

- Total number of products
- Total product quantity
- Total inventory value
- Financial performance indicators

vii). **Stock Lifecycle Management:** Products are categorized into different stages including:

- Active Stock
- Old Stock
- Sold-Out Stock

viii). **Search and Filtering Functionality:** The system includes search capabilities that allow users to quickly locate products based on product name, category, or product code.

ix). **Historical Data Tracking:** The system stores historical

records of transactions and old stock data, enabling users to analyze past inventory trends and performance.

x). **Scalable Database Design:** The PostgreSQL database uses normalized tables, foreign key relationships, and indexing techniques to ensure efficient data storage and retrieval.

Workflow:

i). Product and inventory data are entered into the system through the web interface.
ii). The data is sent from the React frontend to the NestJS backend server.
iii). The backend processes the request and stores the data in the PostgreSQL database.
iv). The analytics module monitors stock levels, sales transactions, and product movements.
v). The dashboard updates inventory statistics and stock insights in real time.

5. Experimental Result

The system was tested using sample inventory data representing different product categories such as mobile phones, books, and electronic items.

Experimental Setup:

i). React
ii). NestJS
iii). PostgreSQL
iv). Dashboard analytics and visualization tools
v). Simulated inventory and transaction data

Observations:

i). **Real-Time Inventory Updates:** The system successfully updated product quantities and stock values immediately after product addition, editing, or deletion.
ii). **Dashboard Analytics:** Inventory statistics such as total stock, total quantity, and total inventory value were displayed dynamically on the dashboard.
iii). **Search and Filtering:** Users were able to quickly locate products using category-based and keyword search features.
iv). **Stock Lifecycle Management:** Products were accurately categorized into active stock, old stock, and sold-out stock, helping users monitor inventory status efficiently.
v). **System Performance:** The platform operated smoothly with fast response time and stable database transactions without data loss.

6. Conclusion

The development of a real-time inventory analytics platform using modern full-stack web architecture provides an efficient solution for managing inventory data and improving business decision-making. The proposed system integrates a responsive frontend, a scalable backend framework, and a reliable relational database to ensure efficient data processing and storage. Through the implementation of React, NestJS, and PostgreSQL, the platform successfully delivers real-time inventory monitoring, product lifecycle management, and automated financial analytics. The system allows users to add, update, and track products while providing important metrics such as total stock value, profit percentage, and inventory trends. Experimental results demonstrated that the system performs efficiently in handling inventory transactions and updating dashboard analytics dynamically. The proposed platform reduces manual effort, improves data accuracy, and enhances visibility into inventory operations. Overall, the system provides a scalable and reliable solution for modern inventory management and can support future enhancements such as predictive analytics, cloud deployment, and intelligent decision-support systems.

References

1. Christopher M. *Logistics and Supply Chain Management*. Pearson Education; 2016.
2. Chopra S, Meindl P. *Supply Chain Management: Strategy, Planning, and Operation*. Pearson; 2019.
3. Fowler M. *Patterns of Enterprise Application Architecture*. Addison-Wesley; 2003.
4. Banks, Porcello E. *Learning React: Modern Patterns for Developing React Apps*. O'Reilly Media; 2020.
5. Hoffman K, Snell M. Modern Web Application Architecture for Scalable Systems. *Journal of Web Engineering*. 2021.
6. Kalyani K. *Swarm Optimization with Neural Networks for Effective Classification Techniques*. *Annals of the Romanian Society for Cell Biology*. 2021;25(4):7413-7419.
7. Kalyani K. *Classification of Microarray Gene Expression using Artificial Neural Network (ANN)*. *Turkish Journal of Computer and Mathematics Education*. 2021;12(7):1372-1378.
8. Kalyani K, Chakravarthy T. *An Algorithmic Approach with Improved Replacement in Bee Optimization Algorithm*. *ICTACT Journal on Soft Computing*. 2015;5(2):905-910.
9. Kalyani K. *Microarray Data Classification using Artificial Neural Network*. *International Journal of Engineering and Advanced Technology (IJEAT)*. 2019;9(1S2):54-56.