IJARETY

International Journal of Advanced Research in Education and TechnologY (IJARETY)

Volume 12, Issue 2, March-April 2025

Impact Factor: 8.152

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

www.ijarety.in editor.ijarety@gmail.com

DOI:10.15680/IJARETY.2025.1202011

Automated Inventory Management System for Small Warehouse using Django

P.G. Gayathri, Dr. M. Senthilkumar, R. Girirajan, R.S.Mukilan, R.Rishinath

Assistant Professor, Dept. of AI&DS, Kongunadu College of Engineering and Technology, Trichy, Tamil Nadu, India

Professor, Dept. of AI&DS, Kongunadu College of Engineering and Technology, Trichy, Tamil Nadu, India

UG Student, Dept. of AI&DS, Kongunadu College of Engineering and Technology, Trichy, Tamil Nadu, India

UG Student, Dept. of AI&DS, Kongunadu College of Engineering and Technology, Trichy, Tamil Nadu, India

UG Student, Dept. of AI&DS, Kongunadu College of Engineering and Technology, Trichy, Tamil Nadu, India

ABSTRACT: The Inventory Management System introduces an efficient and user-friendly platform for managing warehouse operations, focusing on stock tracking, sales processing, and purchase management. This web-based system, developed using Python and Django, provides a centralized Admin Panel that empowers administrators to oversee inventory, suppliers, and transactions with precision and ease. The system ensures real-time stock updates and generates automated invoices for both sales and purchases, significantly reducing manual workload and human error. The project features a dynamic interface, including functionalities for adding, updating, and deleting inventory records, managing supplier profiles, and tracking purchase histories. Additionally, it integrates data visualization tools, presenting stock levels and recent transaction trends through graphical representations. This ensures administrators can access actionable insights into inventory and transaction performance at a glance. The system further enhances operational efficiency by offering features like tax calculations, invoice printing, and recent transaction summaries. By combining automation with intuitive design, the Inventory Management System simplifies complex warehouse management tasks, ensuring accuracy and consistency in operations. Its focus on usability and transparency makes it a robust tool for streamlining small- scale inventory management processes while enabling effective decision-making.

KEYWORDS: Inventory Management, Python, Django Web Application, Data Visualization

I.INTRODUCTION

The Inventory Management System (IMS) is a robust and user-friendly web-based application designed to streamline stock management, sales tracking, and purchase monitoring using Python and the Django Web Framework. As businesses evolve in the digital era, efficient inventory tracking has become a necessity for operational success. Traditional methods of inventory management often lead to inaccuracies, mismanagement, and inefficiencies, which can result in financial losses and operational bottlenecks. This project aims to address these challenges by providing a centralized and automated system that simplifies inventory control and enhances business operations. By leveraging Django's powerful backend capabilities and SQLite for database management, this system ensures real-time tracking of stock levels, supplier details, purchase orders, and sales transactions. The system includes key functionalities such as automated invoice generation, tax calculations, supplier management, and real-time graphical representation of inventory data. These features provide valuable insights that help administrators make informed decisions, ensuring optimal stock levels and preventing shortages or overstocking.

DOI:10.15680/IJARETY.2025.1202011

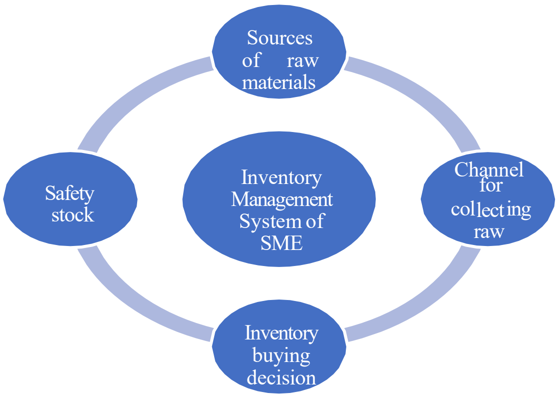
Sources of raw materials
Channel for collecting raw
Inventory buying decision
Safety stock
Inventory Management System of SME

Fig. 1 Introduction of Inventory Management

Beyond its practical applications in businesses and retail, this project serves as an educational resource for first-year IT students, enabling them to understand full-stack web development, backend database management, and real-world application deployment. The system's clean and interactive user interface, built using Bootstrap and Vanilla CSS, ensures a seamless user experience while maintaining ease of access for non-technical users. The IMS Project is particularly relevant in today's digital landscape, where automation plays a crucial role in optimizing workflows. By eliminating the need for manual record-keeping, the system reduces human error, enhances productivity, and improves business efficiency. Additionally, the system's ability to track supplier transactions, generate purchase and sales reports, and maintain supplier profiles makes it a comprehensive solution for small and medium-sized businesses. Furthermore, this project is designed with scalability in mind, allowing future enhancements such as integration with external APIs, barcode scanning, and AI-driven predictive analytics for stock forecasting. The IMS not only digitizes inventory tracking but also serves as a foundation for businesses to adopt advanced technologies in supply chain management. Ultimately, the Inventory Management System in Django stands as a testament to the power of technology in business operations, providing a real-world application of web development, database management, and data visualization. It bridges the gap between theoretical learning and industry requirements, making it a valuable resource for both academic and professional domains. Moreover, this system is highly adaptable and can be extended to include multi-user authentication, role-based access control, and cloud storage integration for greater flexibility and scalability. Future enhancements may incorporate machine learning algorithms to predict stock demands based on past trends, enabling businesses to optimize purchases and reduce costs.

II. LITERATURE SURVEY

Inventory management is a critical aspect of business operations, ensuring efficient stock control, reduced wastage, and improved profitability. Traditional inventory management relied heavily on manual record-keeping, leading to inefficiencies such as overstocking, understocking, and data entry errors. With advancements in web technologies, modern inventory management systems have transitioned towards automated solutions, leveraging frameworks like Django, a Python-based web framework known for its robustness, scalability, and security. Django provides a structured approach to inventory management by integrating a well-defined Model-View-Template (MVT) architecture. The backend database, often powered by PostgreSQL or MySQL, enables seamless storage and retrieval of inventory records, including stock levels, sales transactions, and supplier details. Django's built-in ORM (Object-Relational Mapping) simplifies database operations, allowing developers to manage complex queries with minimal SQL knowledge. Furthermore, Django's admin panel provides a pre-built interface for inventory managers to oversee stock levels, update records, and generate reports without requiring extensive technical expertise. A key feature of Django-based inventory management systems is their ability to handle real-time stock updates. Businesses can track product availability and automate reorder processes through threshold-based alerts, reducing the chances of stockouts or excess inventory. The integration of Django Rest Framework (DRF) enables API-driven solutions, allowing businesses to connect their inventory system with third-party applications such as e-commerce platforms, accounting software, and warehouse management systems. Additionally, security features like user authentication and role-based access control ensure that sensitive inventory data remains protected from unauthorized modifications.

The use of Django for inventory management also enhances user experience through responsive web design. With frontend technologies like HTML, CSS, and JavaScript, businesses can provide intuitive dashboards, real-time analytics, and graphical representations of stock data. By combining Django with JavaScript libraries such as Chart.js or D3.js, businesses can visualize stock trends and make data- driven decisions. The integration of **automated inventory tracking** systems further strengthens operational efficiencies in businesses by eliminating manual errors and ensuring real- time inventory control. Traditional inventory methods, which often involved manual counting or periodic stock audits, resulted in delayed updates and inconsistencies in inventory records. In contrast, automated systems continuously monitor stock movement, updating data instantaneously whenever inventory is added or removed. This ability to update inventory data in real time not only minimizes discrepancies but also ensures that businesses have an accurate picture of their stock levels, fostering more efficient supply chain management. Another significant advancement in inventory management systems is the incorporation of barcode scanning and Radio Frequency Identification (RFID) technologies. These technologies enable quicker and more accurate product identification and tracking. Barcodes and RFID tags can be attached to products, allowing employees to easily scan items during receipt, sales, or stocktaking, significantly reducing the time spent on manual inventory counts and enhancing data accuracy. RFID systems, in particular, can automatically track inventory across various locations without requiring line-of-sight scanning, improving the visibility of stock levels at multiple points in the supply chain.

Integration with Enterprise Resource Planning (ERP) systems has also become a crucial feature of modern inventory management systems. ERP systems connect various business processes, including procurement, manufacturing, finance, and sales, providing a unified platform for managing all aspects of business operations Additionally, cloud-based systems provide reliable data backups and enhanced security, reducing the risks associated with data loss and unauthorized access. By leveraging cloud technologies, businesses can maintain a more flexible and responsive inventory management system, better suited to dynamic market conditions. Furthermore, inventory management systems are increasingly being designed with sustainability in mind. As sustainability becomes a key focus for businesses globally, inventory systems help in reducing waste through better demand forecasting, improved stock rotation, and the reduction of overproduction. By more accurately aligning stock levels with demand, businesses can minimize excess inventory and, in turn, reduce waste. Additionally, these systems can track the lifecycle of products, providing businesses with the ability to manage product end-of-life effectively and ensure that unsold or perishable goods are either returned, repurposed, or disposed of responsibly.

The shift from manual inventory management systems to automated, technology-driven solutions has greatly impacted how businesses manage their stock. These systems help improve operational efficiency, reduce errors, and provide more accurate and up-to-date information about inventory levels. Looking ahead, the future of inventory management will continue to evolve, with businesses increasingly adopting more advanced methods to further optimize their operations. One important trend in the future of inventory management is the continued focus on **streamlining processes**. As businesses grow, the need for more efficient methods to track inventory will become even more critical. Automation will continue to simplify stock tracking, reducing the need for manual intervention, and increasing the speed at which inventory data is updated. This means that businesses will be able to keep better track of their stock levels and make quicker decisions about when to reorder or restock items. Another area that will see growth is the **integration of inventory systems across various business functions**. The days of managing inventory as a standalone process are disappearing. Today, businesses are seeking ways to connect inventory systems with other parts of their operations like sales, procurement, and customer service.

III. PROPOSED METHODOLOGY

The proposed Inventory Management System integrates automation, real-time data processing, and intelligent analytics to revolutionize how businesses manage and optimize their stock. This system redefines traditional inventory practices by ensuring accuracy, efficiency, and proactive decision-making, fostering seamless operations across all levels of supply chain management. The system will dynamically track inventory levels, sales, and supplier interactions in real time, ensuring that businesses have up-to- date insights into their stock status. By continuously monitoring product movement and automating stock adjustments, the system eliminates manual errors and reduces the risk of overstocking or stockouts. This real-time capability ensures that businesses can respond swiftly to fluctuations in demand, enabling more agile and efficient inventory management. Furthermore, by providing accurate forecasting based on historical data and sales patterns, the system empowers managers to make informed decisions regarding procurement and stock replenishment. Automated processes will further transform inventory management by streamlining repetitive tasks such as order generation, stock reconciliation, and reporting. The system will automatically trigger purchase orders when stock levels reach predefined thresholds, reducing manual intervention and ensuring uninterrupted supply. This proactive approach minimizes downtime, optimizes storage space, and reduces operational costs. Additionally, automated auditing features will regularly verify stock consistency, ensuring that inventory records match physical stock and identifying discrepancies before they escalate into significant issues. The system also prioritizes usability and accessibility, ensuring that users across different roles and technical backgrounds can easily navigate and operate the platform. A responsive and intuitive user interface will provide clear dashboards and visual analytics, allowing managers to quickly assess stock health, identify trends, and make data-driven decisions. Customizable user permissions and role-based access controls will ensure that sensitive inventory data is protected, while enabling employees to access the specific functionalities they require.

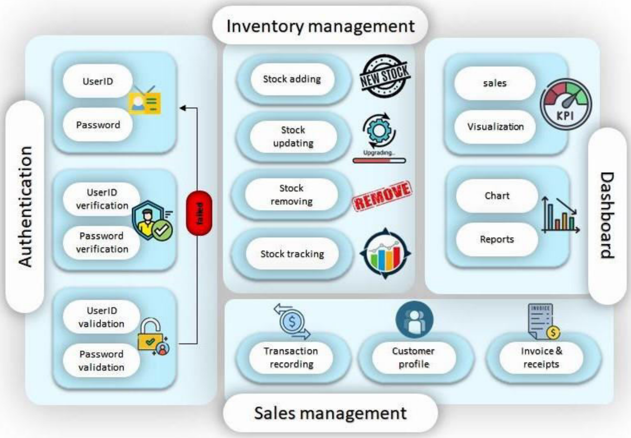

Fig. 2 Technological Architecture

These visualizations support quick, data-driven actions that optimize stock flow and reduce waste. Supplier relationship management is another critical component of the system. By maintaining a comprehensive database of suppliers, including lead times, pricing, and performance history, businesses can streamline procurement processes and negotiate better terms. Automated alerts for delayed shipments or discrepancies in orders ensure that suppliers remain accountable, while fostering transparent and efficient communication channels.

IV. TECHNOLOGIES USED

1. PYTHON DJANGO:

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It follows the Model-View-Template (MVT) architecture, which helps in maintaining a structured codebase. Django's built-in ORM (Object-Relational Mapping) simplifies database management, allowing developers to interact with the database using Python code instead of SQL. The framework also comes with a powerful admin interface, enabling administrators to manage inventory data efficiently without writing additional code.

2. SQLITE:

SQLite is a lightweight, serverless, and self-contained SQL database engine, making it an ideal choice for small to medium-sized applications. It stores the entire database in a single file on disk, simplifying setup and maintenance. In this project, SQLite is used to manage inventory data such as product details, stock levels, supplier information, and transaction histories. Its zero-configuration architecture ensures quick integration with Django, providing reliable data storage for streamlined inventory operations.

3. BOOTSTRAP:

Bootstrap is a popular front-end framework that simplifies the development of responsive and mobile-first web applications. It provides pre-designed components and templates, allowing for a consistent and visually appealing interface. In the inventory management system, Bootstrap ensures that the application is accessible and functional across various devices and screen sizes.

4. CHART.JS:

Chart.js is a JavaScript library used to create visually appealing and interactive charts. It helps in visualizing inventory data, such as stock trends, sales performance, and product movement. By integrating Chart.js, the system provides graphical insights that aid in quick decision-making and inventory optimization.

WHY DJANGO?

Django is a high-level Python web framework that promotes rapid development and clean, pragmatic design, making it an ideal choice for building an inventory management system. Its robust Model-View-Template (MVT) architecture ensures a clear separation of concerns, allowing developers to efficiently manage the data structure, user interface, and business logic. Django's built-in Object- Relational Mapping (ORM) simplifies database interactions, enabling seamless handling of inventory records, stock levels, and sales transactions without extensive SQL knowledge. One of the standout features of Django is its comprehensive admin interface, which allows inventory managers to oversee stock levels, update records, and generate reports with ease.2. Interference-based detection.

The framework's emphasis on reusable code and modular design further enhances maintainability and future-proofing of the application. Django's scalability and flexibility make it suitable for businesses of all sizes. It supports the integration of third-party applications through APIs, allowing for the seamless connection of the inventory management system with e-commerce platforms, accounting software, and warehouse management tools. The framework's emphasis on reusable code and modular design further enhances maintainability and future-proofing of the application.

V. RESULT AND DISCUSSION

The proposed inventory management system, developed using Django and SQLite, offers a streamlined, efficient, and user-friendly solution for managing small warehouse operations. The system enhances stock management, sales tracking, and purchase operations while providing graphical data representation to aid decision-making. By leveraging Django's robust framework, the system ensures rapid development, secure operations, and easy scalability. SQLite, as a lightweight and efficient database, enables quick data retrieval and storage without the overhead of complex configurations. The system's intuitive user interface simplifies inventory tracking, allowing administrators to add, update, and delete inventory records with ease. Real-time stock updates ensure that warehouse managers are always informed about current stock levels, minimizing the risk of overstocking or stockouts. Sales and purchase modules are seamlessly integrated, enabling comprehensive transaction management and facilitating accurate financial reporting.In the fig 1, it shows the graph of time Vs throughput of receiving packet. Throughput is the average rate of successful message delivery over a communication channel.Graphical data representation, powered by Django's integration with visualization libraries, provides clear insights into sales trends, inventory turnover, and supplier performance. This feature empowers warehouse managers to make data-driven decisions, optimize stock levels, and improve overall operational efficiency. The system's automation capabilities reduce human error, streamline repetitive tasks, and enhance productivity. The system is designed with scalability in mind, allowing for easy adaptation to larger warehouse operations or additional functionalities as needed. The modular architecture ensures that new features can be integrated without disrupting existing operations. The use of Django's built-in security features, such as user authentication and permission management, guarantees data integrity and protects sensitive information. Accessibility and user experience are prioritized, with a responsive design that ensures seamless operation across various devices, including desktops, tablets, and smartphones. The system supports multi-user access, enabling collaborative management of inventory by different team members. Role-based access control ensures that users have appropriate permissions based on their responsibilities.

An additional feature of the system is its ability to generate comprehensive reports on inventory status, sales performance, and supplier metrics. These reports can be exported in various formats such as PDF and Excel, facilitating easy sharing and integration with other business tools. The system also includes alert mechanisms that notify managers of low stock levels, pending orders, or discrepancies in inventory, ensuring proactive management and timely decision-making. The system's integration capabilities allow it to connect with other enterprise systems, such as accounting software and customer relationship management (CRM) tools, providing a holistic view of the business operations.
This integration helps streamline workflows and ensures consistency across different business functions.

By automating inventory management processes, the system significantly reduces the time and effort required for manual tracking and reporting. This leads to increased operational efficiency, reduced costs, and improved accuracy. The system also facilitates better supplier management by tracking supplier performance and ensuring timely reordering of stock. In conclusion, the proposed inventory management system offers a comprehensive, efficient, and scalable solution for small warehouse operations. Its integration of Django and SQLite ensures a robust, secure, and user-friendly platform that meets the diverse needs of warehouse managers. The system's ability to provide real-time insights, automate processes, and enhance decision-making makes it an invaluable tool for optimizing inventory management and improving overall business performance.

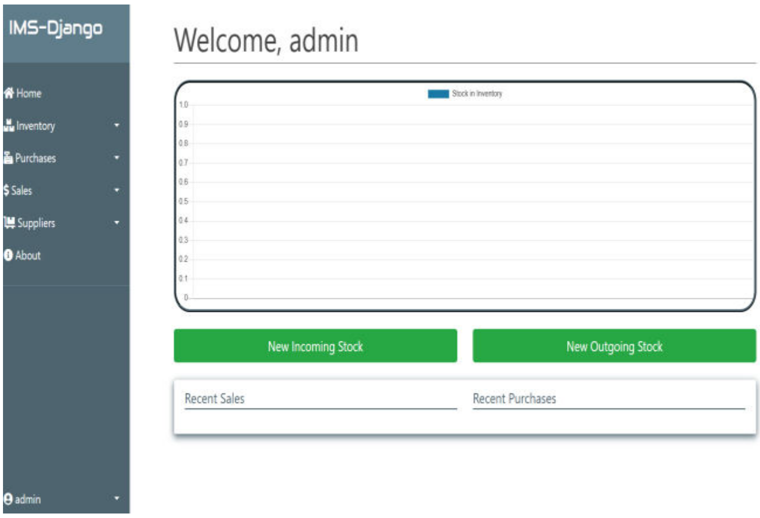

Fig.3. Output Of Inventory Management Dashboard

VI. CONCLUSION

The development and implementation of the Django- based inventory management system represent a transformative step forward in streamlining warehouse operations and optimizing inventory control. By leveraging Django's robust framework and SQLite's lightweight database, this system delivers an efficient, secure, and scalable solution tailored to the needs of small warehouse operations. The integration of real-time stock updates, automated processes, and graphical data representation provides warehouse managers with comprehensive insights, enabling informed decision-making and operational efficiency.A key advantage of this system is its ability to automate routine inventory management tasks, thereby reducing human error, minimizing manual effort, and enhancing productivity. The intuitive user interface and responsive design ensure that users can easily navigate the system across various devices, fostering accessibility and ease of use. Additionally, the system's role-based access control and robust security features safeguard sensitive information, maintaining data integrity and ensuring that users have the appropriate permissions based on their roles.

This project not only improves operational efficiency and reduces costs but also enhances decision-making capabilities, contributing to overall business growth and success. By integrating advanced technologies and focusing on user-centric design, the system represents a significant advancement in modern inventory management solutions, ensuring sustained performance and adaptability in a dynamic business environment.

REFERENCES

[1] Analysis of Inventory Management Using Methodology ROP (Reorder Point) to Minimize DOI (Days of Inventory)W. Abimanyu; H. Wawan R.Erry 2019 International Journal of Innovative Science and Research Technology.

[2] Inventory Management Practices Among Malaysian Micro Retailing Enterprises. K. Ahmad; S.M. Zabri 2024 A Journal of the Academy of Business and Retail Management (ABRM).

[3] Inventory Management Practices Among Small and Medium-Sized Retail Enterprises. K. Ahmad; S. Mohamed Zabri; M.I.P. Mohamed 2024 Proceeding of the 3rd International Conference on Technology Management, Business and Entrepreneurship, Melaka, Malaysia.

[4] Dynamic Discrete Inventory Control Model with Deterministic and Stochastic Demand in Pharmaceutical

Distribution. S. Antic; M.L. Djordjevic; A. Lisec 2024 Applied Science..

[5] Optimal Inventory Policy for a Perishable Item with Demand Function Sensitive to Price and Time. T. Avinadav, A. Herbon, U. Spiegel 2024 International Journal of Production Economics.

[6] Multi-Level Production-Inventory Control: Material Requirements Planning or Reorder Point Policies, S. Axsäter, K. Rosling, 2024 European Journal of Operational Research.

[7] Optimizing Inventory and Store Results in Big Box Retail Environment, A. Ayad, 2024 International Journal of Retail & Distribution Management.

[8] Procurement: Principles and Management, P. Baily; D. Farmer; B. Crocker; D. Jessop; D. Jones, 2024.H. Khalife, N. Malouch, S. Fdida, “Multihop cognitive radio networks: to route or not to route,” IEEE Network, vol. 23, no. 4, pp. 20-25, 2009.

[9] Improving Inventory Performance with Clustering- Based Demand Forecasts, P.K. Bala, 2023 Journal of Modelling in Management.

[10] 1Continuous Review Inventory Models for Perishable Items Ordered in Batches, O. Baron O. Berman; D. Perry, 2023 Mathematical Methods of Operations Research.

[11] A Systematic Literature Review About Dimensioning Safety Stock Under Uncertainties and Risks in the Procurement Process, J. Barros; P. Cortez; M.S. Carvalho,2023 Operations Research Perspectives.

[12] On the Constrained Multi-Item Single-Period Inventory Problem, M. Ben-Daya. A. Raouf, 2023 International Journal of Operation and Productions Management.

[13] K Baskar, GKDP Venkatesan, S Sangeetha, “ A Survey of Workload Management Difficulties in the Public Cloud”, Intelligent Computing in Engineering: Select Proceedings of RICE 2019, Springer Singapore.

[14] Dr Prasanna Venkatesan GKD Baskar K, “A decentralized workload management system for a four- dimensional hyper cubic structure in the public cloud”, International journal of Recent Technology and Engineering, 2018.

[15] Enterprise Resource Planning (ERP) System in Higher Education: A Literature Review, B. Bhamangol; V. Nandavadekar; S. Khilari, 2022 International Journal of Management Research and Development.

[16] Production, Manufacturing, and Logistics on Multi- Item Inventory, D.K. Bhattacharya,2022 European Journal of Operational Research.

[17] An Investigation into the Use of ERP Systems in the Service Sector, V. Botta-Genoulaz, P.A. Millet,2022 International Journal of Production Economics.

[18] Improved Base-Stock Approximations for Independent Stochastic Lead Times with Order Crossover, J.R. Bradley; L.W. Robinson, 2022 Manufacturing & Service Operations Management.

[19] Logística Gestão da Cadeia de Abastecimento, J.C. Carvalho; A.P. Guedes; A.J.M. Arantes; A.L. Martins; A.P.B. Póvoa, C.A. Luís et al 2021.

[20] An Inventory Model with Ramp Type Demand, Time Varying Holding Cost, and Price Discount on Backorders, S. Chandra, 2021 Uncertain Supply Chain Management.

IJARETY

INNO SPACE
SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर
NISCAIR

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975 Impact Factor: 8.152

www.ijarety.in editor.ijarety@gmail.com