Internet of Things (IoT) for Smart Inventory Management

Ashutosh
Monika Rethi

Abstract:
A smart inventory system is a computational time efficient system that helps businesses to manage and track their inventory levels, orders, and deliveries. It facilitates companies to have real-time visibility into their inventory and helps them make more informed and quick decisions about restock and how much to order. One key feature of a smart inventory system is its ability to automatically reorder items when they reach a certain threshold, eliminating the need for manual intervention. This helps to ensure that businesses always have the right amount of inventory on hand, reducing the risk of running out of stock or having excess inventory that takes up valuable storage space. This paper includes deployment of this system in real world which benefits to handle smart inventory system with improved accuracy 30% and efficiency in inventory tracking around 10%, reduced lead times for ordering and restocking, and the ability to track inventory across multiple locations. This paper briefly elaborates the implementation of smart inventory system that greatly improves a business's inventory management process, leading to increased profitability by more than 50%, average foot fall increased to 25% and reducing the waiting time of customer by nearly 75% making customer more satisfied.
Keywords: *IoT, Cloud Server, Android application, ESP 32, Smart Inventory, Industrial Automation.*

1. Introduction

Modern warehouses, characterized by multiple picking zones and thousands of item retrievals per hour, demand speed and precision from their workforce to meet high consumer expectations. Technological innovations have played a vital role in enhancing these processes, one such being the "Prototype" methodology, which has significantly evolved to streamline warehouse functionality [1].

Maintaining complete visibility and control over inventory across all touchpoints is indispensable for operational efficiency. With the increasing emphasis on performance and long-term viability, precise inventory oversight has become more important than ever. Inefficiencies in stock record accuracy can lead to substantial financial drawbacks. For instance, research by Chuang and Oliva demonstrated that an Inventory Record Inaccuracy (IRI) level of 29% in a retail setting can slash earnings by up to 10% [2]. Traditional methods of tracking inventory manually often suffer from inaccuracy, slowness, and susceptibility to human oversight. Discrepancies may arise from incorrect entries, unidentified inventory sources, or misplaced goods.

In contrast, light-assisted picking systems allow for the selection, arrangement, and preparation of items without paper-based instructions [3]. These setups use visual indicators, such as lights installed near storage compartments, to guide staff toward the correct product and quantity for each order. Such systems minimize unnecessary movement and reduce the reliance on reading order sheets or semi-automated instructions, which in turn enhances accuracy and speeds up the fulfillment process.

Additionally, an integrated application has been introduced to simplify inventory handling. It supports features like organizational profiling, monitoring sales and procurement activity, and keeping a real-time log of available stock. The platform allows for updating inventory records in accordance with transactions or product returns, providing transparency in stock levels and financial data. New items are registered with a distinct label and timestamp, and the system includes secure login protocols to deter unauthorized access or misuse of resources [4].

As smartphones encompass both Android and iOS ecosystems, the study further suggests the development of warehouse management utilities compatible with these devices. Drawing from a case study involving a university's automated storage system, the initiative emphasizes a mobile-first approach. In tandem with desktop-based WMS platforms, these handheld solutions empower warehouse teams to operate efficiently throughout the facility without relying on stationary systems [5]. Data related to storage and retrieval activities collected via the app is transmitted directly to the backend server for centralized processing and analysis [6].

2. Literature Survey

Chih-Chin Liang [7] introduced a forecasting framework aimed at improving inventory assessment. This model, leveraging domain-specific sequential pattern insights, demonstrated an optimal prediction precision of up to 66.3%. It offers valuable implications for inventory coordination within the food logistics and manufacturing domain, allowing for dependable and streamlined oversight of stock levels using expert-derived data patterns.

In a related development, Souvik Paul [8] elaborated on a system architecture grounded in the Internet of Things (IoT), offering enhanced functionality when compared to conventional stock handling techniques. The system's architecture promises improvements in responsiveness, scalability, and accuracy, addressing the growing demand for efficiency in globalized operations involving extensive component arrays and geographically distributed storage hubs. Additionally, the study addressed the incorporation of robotic mechanisms for sorting and handling materials, grounded in traditional asset tracking foundations.

Rajesh Bose [9] contributed to this field by presenting an innovative framework tailored to optimize material handling, specifically focusing on construction supplies such as shuttering and formwork assets. Although the empirical foundation of this work is rooted in Indian infrastructural firms, the applicability of the model extends to varied geographies, offering potential value for inventory planning across diverse construction environments.

Ali Alwadi [10], on the other hand, provided a detailed survey of contemporary advancements in "Radio-Frequency Identification (RFID) enabled stock systems. The work discusses key obstacles in designing these frameworks, followed by an in-depth comparative analysis of RFID modalities, configurations, and operational protocols. The review also includes state-of-the-art middleware advancements, emphasizing elements such as tag classifications, reader mechanisms, antenna designs, and integration layers. Special attention is given to collision management strategies and throughput optimization in passive RFID environments, highlighting both the strengths and operational constraints of various algorithmic implementations used in such systems.

3. Methodology

The methodology for implementing a smart inventory system involves the integration of automation technologies to streamline traditional stock management processes. The system is designed to replace labor-intensive manual tasks—such as item counting, status tracking, inventory updates, restocking notifications, and report generation—with automated, real-time operations. At the core of this approach is the deployment of IoT sensors, RFID tags, and barcode scanners that work in tandem to ensure seamless data acquisition from every stage of inventory movement. These devices continuously collect and transmit data to a centralized inventory management platform, allowing for instant updates and round-the-clock visibility into stock levels and locations.

The system architecture also includes cloud-based databases and analytical engines that process the incoming data to detect discrepancies, forecast demand, and generate actionable insights. Machine learning algorithms may be embedded to refine restocking schedules and predict future requirements based on historical trends and real-time consumption patterns. The user interface, accessible via mobile or desktop applications, enables stakeholders to monitor inventory status, approve requisitions, and configure alerts for low-stock thresholds. This comprehensive automation eliminates redundant paperwork and minimizes the likelihood of human oversight or error. The overarching methodology thus emphasizes accuracy, efficiency, and responsiveness in stock handling, ensuring uninterrupted operational continuity and informed decision-making [11].

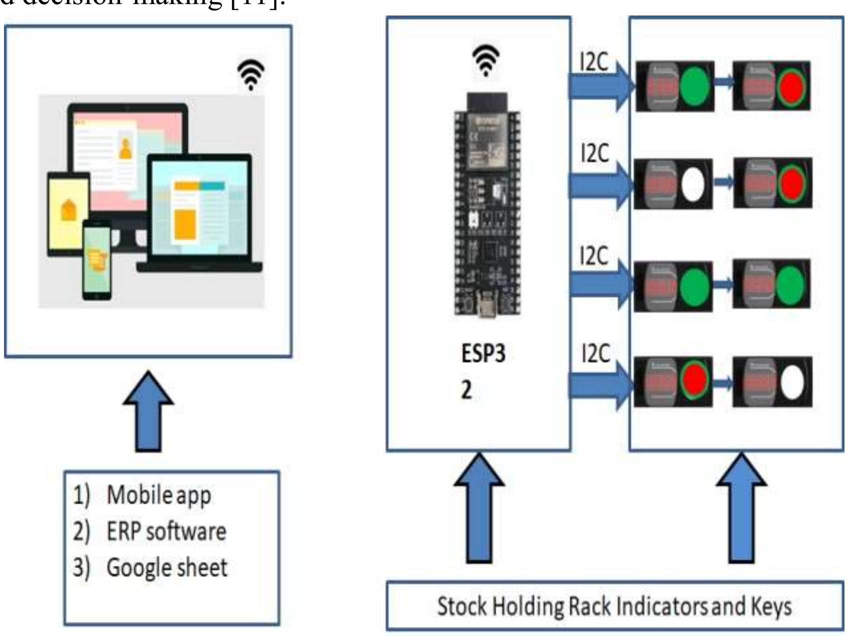

Fig. 1 Smart Ware House System

Fig. 1 depicts the overall implementation of smart warehouse system incorporating mobile app, ERP software and database. Inventory is updated timely whenever changes occur in stock, the same is reflected to on rack display for warehouse rack handling person. The stock can be verified through central software or from individual rack, both reflects similar count. The stock update algorithm runs whenever the stock is modified.

Fig. 2 Smart Warehouse Application Software

Figure 2 illustrates the mobile interface of the smart warehouse management application. This software enables users to access warehouse functionalities by initially connecting to the designated warehouse network using a specific IP address. After establishing a stable network link, users proceed to the authentication interface to log in and oversee various operations.

To ensure seamless wireless communication within the facility, it is essential to maintain uninterrupted Wi-Fi coverage. This can be achieved by strategically deploying a sufficient number of wireless access points and signal boosters across the premises.

Each storage rack comprises several tiers, each supporting multiple bins that hold different types of components or materials. Every bin is identified with a distinct code, which corresponds to a uniquely assigned shelf identifier. This coding system aligns with the actual placement of items, ensuring accurate physical-to-digital mapping within the storage environment [12].

For user convenience, a seven-segment display is mounted adjacent to every bin, allowing personnel to view the quantity of stored items at a glance without needing to consult the central database. Furthermore, each container is equipped with dedicated control buttons that allow operators to manually update the inventory count on the spot. This is particularly useful when materials are added or withdrawn directly from the bins without digital intervention. In such cases, the revised count is automatically transmitted to the central control system for synchronization and confirmation. This integrated feedback mechanism represents a distinctive innovation embedded within the application's design.

3.1. Block Diagram

The racks has unique ID based on the IP address. Fig. 3 depicts the overall configuration of racks and container, where each rack hold multiple shelf which holds single or multiple container depending upon the shlef or material size. The unique ID for rack is through the programmed IP address allocated to WiFi modules, the onboard wifi module based microcontrollers boards are prefeered in this design to maintain simplicity, These Wi-Fi modules allow all the devices to communicate with one another. This design uses ESP 32 Wi-Fi module [13].

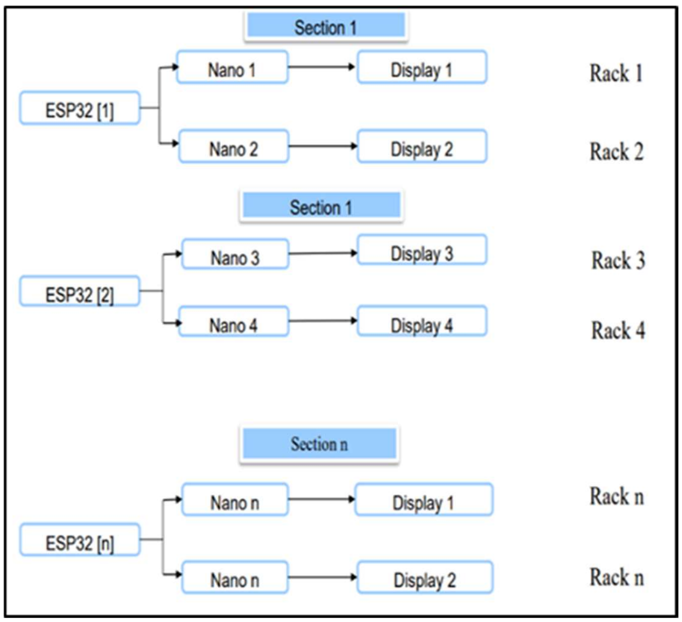

Fig. 3 Block Diagram of Racks, Shelf and Container

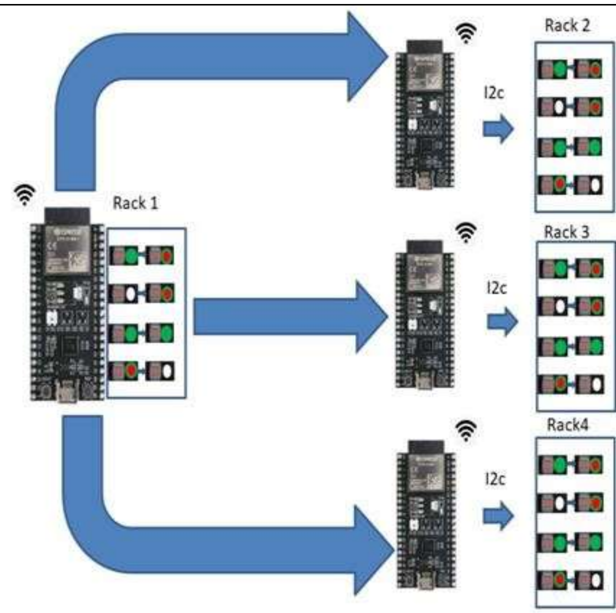

Fig. 4 Connection Diagram of Wi-Fi Modules Container Module

Fig. 4 describes the connection between each rack Wi-Fi modules and its associated container module. Once the WiFi module is network-connected, the data will be retrieved from the cloud and transferred to the Arduino Nano for display on the seven-segment display. The variety of communication buses available over the AVR based board and these can be used to interact with a computer, Arduino board, or other microcontrollers. Each container indicators and associated keys are interfaced to its unique ATmega328P based arduino Nano development board, so as the easy communication established between ESP and different Nano boards over the rack through IIC bus communication.

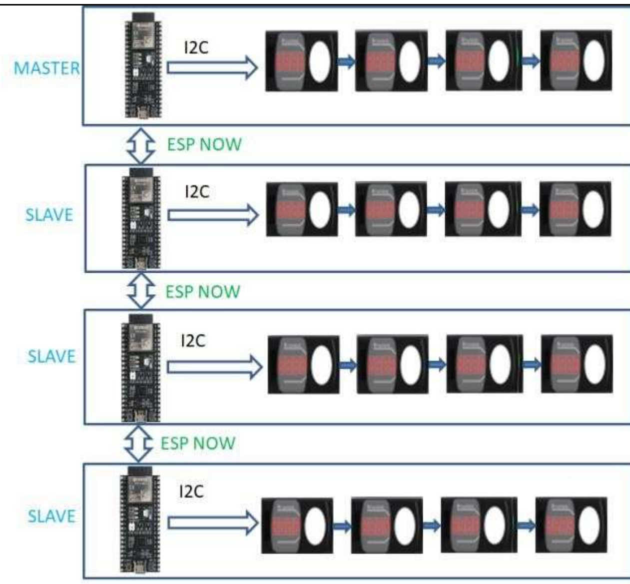

Fig. 5 Shelf and Container

A seven-segment display is a type of electronic display that can be used instead of more complicated dot matrix displays to show decimal numbers. Fig. 5 describes the arrngement of Shelf and Container, where to assist and update the use each container is designed with its unique display to showcase the current inventory count. The system incorporated three switches: one for adding and removing components, one for updating the data on the LCD and cloud database, and one for updating both. When we hit the increment or decrement button, the timer value is modified and updated, and the arduino Nano's Tx pin is utilized to transmit the modified material count value to the Wi-Fi module. The Wi-Fi module will connect to the Cloud in order to update the inventory system's real-time data as it receives data from the Arduino Nano. Thus the data updation is done through the established connection between the cloud database and the android application.

3.2. Flowchart

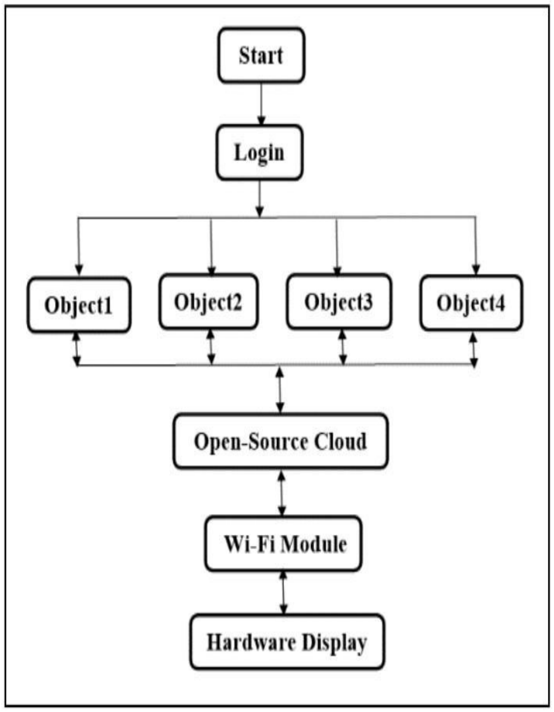

Fig. 6 Flowchart for Application

Fig. 6 depicts the Flowchart for overall application implementation of smart warehouse system incorporating mobile app, ERP software and database. Once User login, then the updated inventry is visible and can do further operations using the application software. Whenevre changes occure in stock, the same is reflected to on rack display for warehouse rack handling person and cloud database for central persons.

3.3. Algorithm

The smart warehouse system is composed of two primary components: the software interface and the physical hardware. The interaction between these two sections is bidirectional, ensuring real-time synchronization and control. The operational algorithm begins with the initiation of the software application. Upon launch, users are prompted to input their login credentials, including a unique ID and password, to gain access. Once authenticated, the user proceeds to input the quantity of components associated with specific containers.

This input data is then stored on a cloud-based, open-source database, enabling seamless access and management. From here, the hardware side of the system comes into play. Using the integrated ESP32 Wi-Fi module, the hardware fetches the stored data directly from the cloud. The retrieved information is then visually represented on a seven-segment digital display positioned near each container, allowing for easy readability. This cycle—from data entry to hardware display—is continuous and can be repeated as many times as necessary, ensuring updated inventory counts and streamlined warehouse operations.

From Hardware to software:

The process begins by waiting for an action to be performed on the physical switch. Once triggered, the hardware switches are used to manually update the data based on the specific operational requirements. These updates are then transmitted wirelessly from the hardware to an open-source cloud platform through the ESP32 Wi-Fi module. After successful synchronization, the updated information is reflected in the corresponding Android application in real time. The system then loops back, ready to detect the next switch interaction, thereby creating a continuous cycle of manual input, cloud synchronization, and application-level updates.

4. Implementation and Results

The real-world implementation and testing of the said system is done in the local electronic components shop selling approximately 10,000 components in a day. This shop open for six day a week and all the months in a year. It is Observed that the net time required to identify the correct component container is drastically reduce for both new employee or experienced. Thus the total delivery per time increased and it saves overall standing time of customer, increasing more footfalls and sale. Thus the store now accommodates more components with very small laydown time. The only constraint is to keep the power up of all wifi modules during the shop opening time. The inventory is tracked lively and orders are placed whenever the component or material shortage occurs. The Android application designed in Android Studio is installed in various devices. This mobile app is connected to the opensource cloud based database. Both the program and the WiFi module provide real-time data. Initially user has to login successfully to connect in to the Android app, then user will be able to modify the app's data and update it on the open source cloud. You can also use the addition and subtraction buttons. Manual processes such as item tracking and counting, inventory database updating, report writing, restocking, and so on can be automated with the assistance of a smart inventory system, saving time and effort while reducing the possibility of human error.

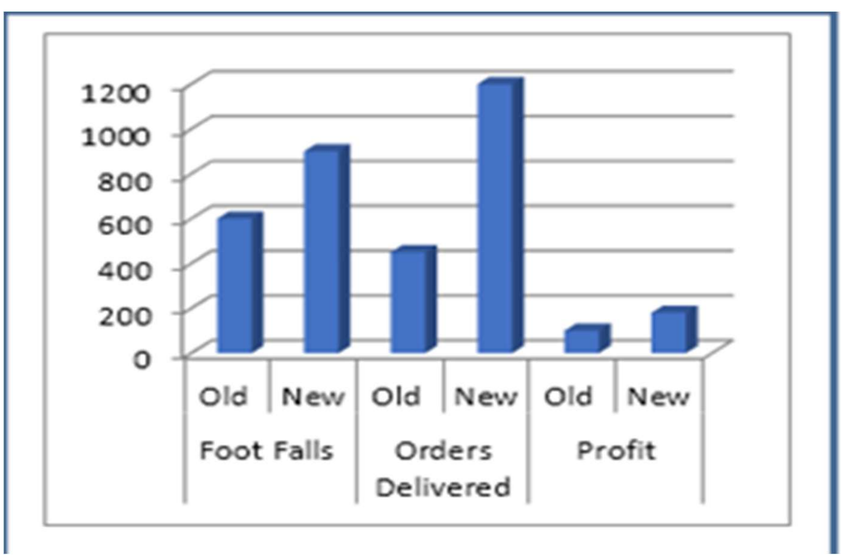

Fig. 7 Footfalls, Average Order Delivered and Profits

Fig. 7 shows the graphical representation of average daily footfall in the stores along with average waiting time and the profit margin. From the diagram it is cleared that the footfall without the smart inventory management is much less than the new footfalls with implementation of smart inventory system. However it's almost increases by 50% with respect to old footfalls. This effect is due to the quick delivery of goods to the customer and reduced waiting time simultaneously. From the diagram clearly states number of order delivered without the smart inventory management is much more than the order delivery time with implementation of smart inventory system. It's almost decrease by 75% with respect to old delivery time. Thus the overall waiting time of customer is reduced due to the quick delivery of goods to the customer. The overall profit ration for the day also increases as the above parameters improve.

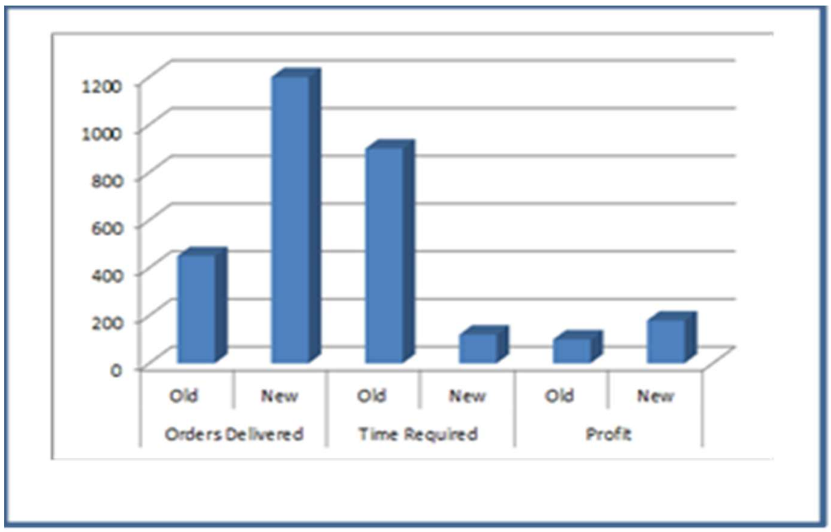

Fig. 8 Order Delivered, Time Required and Profits

Fig. 8 shows the graphical representation of average daily order delivered from the stores along with average waiting time and the profit margin. From the diagram it is cleared that the profit is less when the store not using smart inventory management, however the profit parameter improves once the store is supported by the smart inventory system which improves average waiting time for order, average footfall and delivery time.

5. Conclusion

The 'Smart Warehouse Management System" (SWMS) presented in this paper showcases a comprehensive and practical approach to modernizing traditional inventory and logistics processes within warehouse environments. By leveraging automation, intelligent hardware integration, and seamless connectivity, the system significantly enhances operational efficiency and overall performance. One of the most impactful outcomes observed is the substantial increase in customer footfalls—by nearly 50%—which can be attributed to faster order fulfillment and minimized waiting times. The optimization of order dispatch processes has also resulted in a dramatic reduction in delivery durations, with improvements reaching up to 75% compared to earlier methods. These enhancements directly contribute to increased customer satisfaction and, consequently, improved profit margins for the business.

A key feature of the system is its dual-mode inventory update mechanism. Users can update the inventory count either manually through physical keys located at each component bin or digitally via the centralized computerized system. This dual approach ensures flexibility and real-time accuracy in stock monitoring. Furthermore, the system integrates intelligent displays and controls at the container level, allowing on-site personnel to manage inventory adjustments instantly, which are then reflected in the central database. This feedback loop ensures that the digital inventory aligns closely with physical stock conditions, minimizing discrepancies.
Beyond inventory updates, the SWMS also automates several routine tasks such as tracking item movement, updating stock records, generating reports, and initiating restocking procedures. These functions, which traditionally relied on manual labor, are streamlined through the smart system, reducing both time consumption and the likelihood of human errors. Additionally, the system supports a completely paperless workflow, further enhancing sustainability and operational transparency. Overall, the Smart Warehouse Management System not only revolutionizes warehouse operations through technological integration but also sets a foundation for scalable, efficient, and user-centric inventory management in future industrial applications.

References

Singh, K. (2019). Smart warehouse management using electronic sensor-based computational intelligence. *SCI*, 823.

Vamsi, A. M. (2019, October). IoT based autonomous warehouse management for warehouses. *EAISICC*, 371–376.

Tejesh, B. S. S. (2018). Warehouse management system using IoT and open source framework. *Alexandria Engineering Journal*, *57*(4), 3817–3823.

Chen, M.-C., Cheng, Y.-T., & Siang, C.-Y. (2022). Development of inventory management system based on radio frequency identification technology. *Sensors and Materials*, *34*(3), 1163–1177.

Olanrewaju, R. F. (2021). Cloud-based warehouse system for effective management of under and over-stock hazards. *ICCCE*, 274–278.

Korkmaz, I. (2015). A cloud-based and Android-supported scalable home automation system. *compeleceng*, *43*, 112–128.

Liang, C.-C. (2013). Smart inventory management system of food-processing-and-distribution industry. In *Procedia Computer Science*, *17*, 373–378. Elsevier.

Paul, S. (2019). Study of smart inventory management system based on the Internet of Things. *International Journal on Recent Trends in Business and Tourism*, *3*(3).

Bose, R., Mondal, H., Sarkar, I., & Roy, S. (2022). Design of smart inventory management system for construction sector based on IoT and cloud computing. *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, *2*.

Gawande, S. P., & Tambe, S. B. (2019, June). Inventory management system for warehouse. *International Research Journal of Engineering and Technology (IRJET)*, *6*(6), 3454–3458.

Tan, W. C., & Sidhu, M. S. (2022). Review of RFID and IoT integration in supply chain management. *Operations Research Perspectives*, *9*, 1342–1348.

Zope, A. S., Jambhale, M. S., Korde, N. M., Khan, H. A., & Bhargav, A. (2017). IoT based industrial automation. *International Journal of Engineering Research and Technology*, *5*(1), 821–827.

Mane, P. B., Pethakar, M. S., & Pawar, R. R. (2017). Watermarking and cryptography-based image authentication on reconfigurable platform. *Bulletin of Electrical Engineering and Informatics*, *6*(2), 181–187.

Mandwale, A. J., Purohit, H. G., & Gawande, M. B. (2015, January). Different approaches for implementation of Viterbi decoder on reconfigurable platform. In *2015 International Conference on Pervasive Computing (ICPC)* (pp. 1–4). IEEE.

Mane, P. B., & Mulani, A. O. (2018). High-speed area-efficient FPGA implementation of AES algorithm. *International Journal of Reconfigurable and Embedded Systems*, *7*(3), 157–165.

Kashid, M. M., Shaikh, S. A., Kadam, M. S., & Mujawar, A. N. (2022). IoT-based environmental parameter monitoring using machine learning approach. In *Proceedings of the International Conference on Cognitive and Intelligent Computing: ICCIC 2021* (Vol. 1, pp. 43–51). Springer Nature Singapore.

Yadav, D., & Gangwar, S. (2023). Generic home automation system using IoT gateway based on WiFi and ant+ sensor network. *International Journal on Recent and Innovation Trends in Computing and Communication*, *11*(3), 153–165. https://doi.org/10.17762/ijritcc.v11i3.6332

Rossi, G., Nowak, K., Nielsen, M., García, A., & Silva, J. (n.d.). Enhancing collaborative learning in engineering education with machine learning. *Kuwait Journal of Machine Learning*, *1*(2). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/119

Kumar, A., Dhabliya, D., Agarwal, P., Aneja, N., Dadheech, P., Jamal, S. S., & Antwi, O. A. (2022). Cyber-internet security framework to conquer energy-related attacks on the internet of things with machine learning techniques. *Computational Intelligence and Neuroscience*, 2022. https://doi.org/10.1155/2022/8803586