Time-Based Priority Dispatch System for Warehouses Using Low-Cost Embedded Hardware

Munagapati Karthik
Electrical and Electronics Engineering
Koneru Lakshmaiah Educational Foundation Vijayawada, India
munagapatikarthik09@gmail.com

***Abstract*— Efficient dispatch and storage operations are essential for maintaining competitiveness in modern supply chains. Yet, many small and medium-scale warehouses still rely on manual inventory handling, which results in operational delays, mismanagement, and higher labour dependency. This paper presents a smart and affordable solution: a Time-Based Priority Dispatch System using low-cost embedded hardware and cloud integration. The system uses ESP32 microcontrollers and RFID technology to automatically scan and identify incoming products. Once scanned, product data—including time and type—is transmitted to Google Firebase, a real-time cloud database. Based on this data, the system intelligently determines zone-based storage locations and sets dispatch priorities depending on the urgency of the item.**

A real-time clock module ensures accurate time tracking, while automation logic categorizes and moves goods to appropriate zones. Through Wi-Fi connectivity, data is continuously updated and reflected on a web-based dashboard that provides live monitoring, product analytics, and dispatch status. The interface enables warehouse managers to track each item's status, analyse movement patterns, and ensure high-priority products are processed on time. The integration of cloud storage, IoT, and automation provides a cost-effective, scalable solution for improving warehouse efficiency. This system reduces human error, enhances transparency, and supports real-time decision-making. Overall, it offers a practical model for smart warehouse management—democratizing access to advanced logistics technologies for resource-constrained businesses and paving the way for smarter, data-driven supply chain solutions.

Keywords—*RFID Tracking, ESP32 Microcontroller Google Firebase , IoT-based Automation , Time-Based Dispatch ,PriorityScheduling,EmbeddedSystems,Warehouse Management ,Real-Time Monitoring , Cloud Integration ,Zone Allocation , Delivery Optimization , Low-Cost Automation , Web Dashboard , RFID Inventory System , Real-Time Clock (RTC),Dynamic Zone Reallocation , Autonomous Dispatch System Logistics Efficiency*

I. INTRODUCTION

The rapid expansion of global trade and e-commerce has transformed warehouse management into a critical pillar of supply chain efficiency. With increasing consumer expectations for timely deliveries and real-time inventory updates, traditional warehouse systems—often reliant on manual processes—struggle to meet modern logistical demands. These outdated systems are particularly common in small and medium-sized facilities where budget constraints prevent the adoption of advanced warehouse management systems (WMS).

In response to this gap, recent advances in embedded systems, wireless communication, and cloud computing offer promising avenues for developing smart, cost-effective warehouse solutions. Leveraging technologies such as microcontrollers, RFID, and real-time cloud databases can enable automation and improve accuracy, even in low-resource environments.

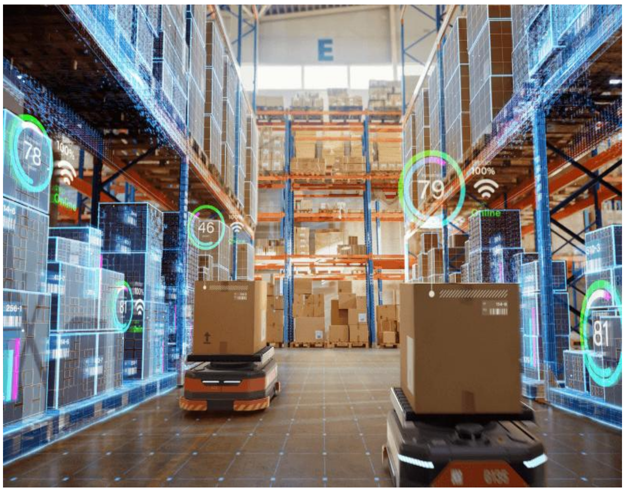
This study introduces a novel approach: a time-based priority dispatch system using low-cost hardware and open-source platforms. By combining RFID-enabled inventory tracking, real-time cloud data handling via Google Firebase, and intelligent automation using ESP32 modules, the system offers an efficient alternative to expensive industrial solutions. A web-based interface provides live monitoring and analytical insights, enhancing operational transparency and decision-making.

The aim is not just to automate dispatch operations, but to empower smaller warehouses with scalable, intelligent tools that bridge the gap between traditional methods and smart logistics.

II. CONCEPT OF THE PROPOSED SYSTEM

A. *Description of how the system works*

The proposed Time-Based Priority Dispatch System for warehouses leverages RFID technology and low-cost embedded hardware to automate and optimize the inventory flow from arrival to dispatch. The process begins when a product enters the warehouse and is scanned using an RFID reader connected to an ESP32 microcontroller. Each product is tagged with essential metadata, including arrival time,



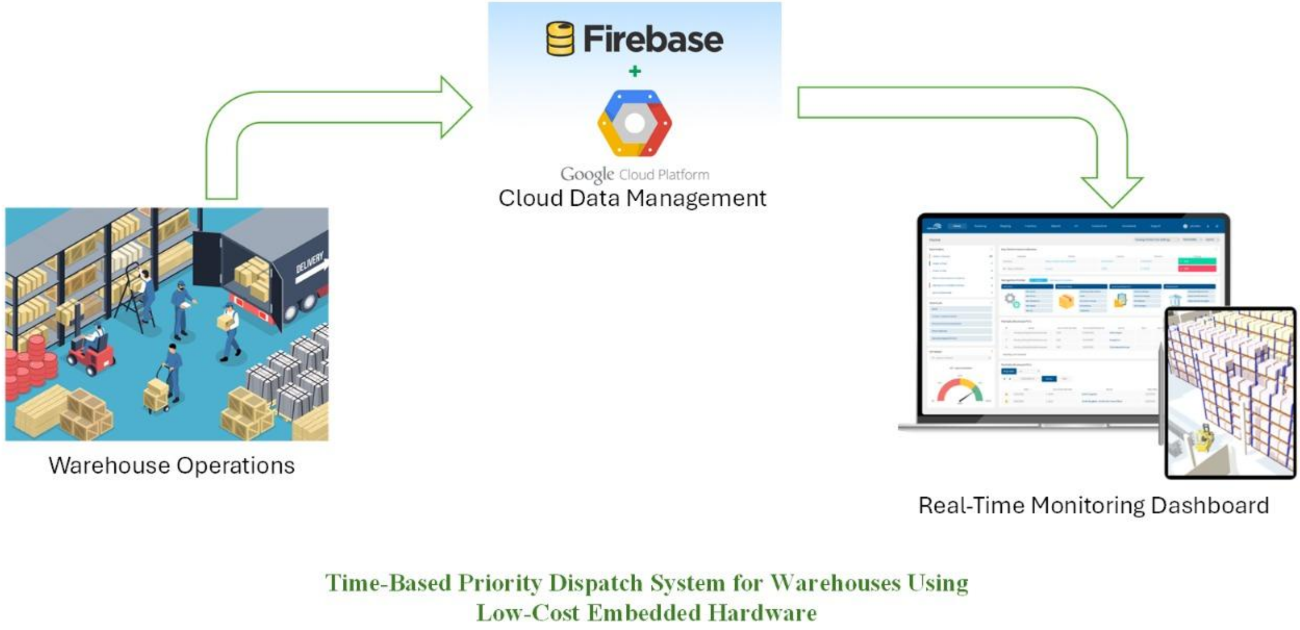


Figure 1 BLOCK DIGRAM OF THE TIME BASED PRIORITY

expected delivery date, and customer details, which are immediately sent to a cloud database via Google Firebase.

Once the product is logged into the system, an intelligent scheduling algorithm evaluates the time left for delivery and dynamically assigns the product to an appropriate storage zone. The zones within the warehouse are prioritized based on their proximity to the dispatch point and urgency of delivery. For instance, if a product has a shorter delivery window, it is automatically assigned to a closer zone like Zone 3 to minimize handling time and ensure rapid dispatch.

As time progresses, the system continuously monitors and updates the product's status. When the dispatch time approaches, the system triggers movement toward the outbound zone. If the product is re-scanned in its designated dispatch zone, it is marked as dispatched in the system, and the Firebase database is updated accordingly.

A real-time dashboard, built using web technologies, visualizes the entire process—displaying current product details, live counts, dispatch history, and zone assignments. This enables warehouse staff to monitor operations seamlessly and make informed decisions without manual intervention, reducing human error and improving delivery efficiency in a cost-effective manner.

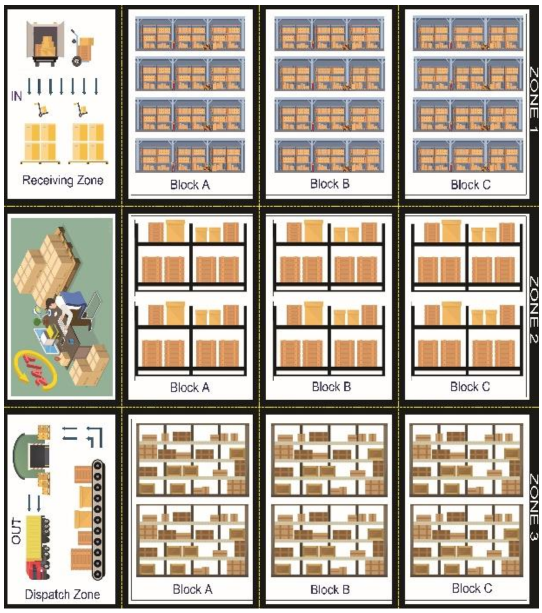
B. *Zones and blocks*

In the proposed Time-Based Priority Dispatch System, the warehouse is logically divided into multiple **zones and blocks** to streamline product storage and dispatch operations based on delivery urgency. Each zone represents a section of the warehouse, and blocks within the zones act as designated storage areas for tagged products.

Upon arrival and RFID scanning, the system evaluates each product's **arrival time**, **delivery deadline**, and **priority level**. Based on this time-sensitive data, the system assigns the

product to a specific **zone and block** using a dynamic scheduling algorithm.

- **Zone Allocation**: Zones are prioritized based on proximity to the dispatch area. Products with **shorter delivery time windows** are allocated to zones nearer to the outbound gate (e.g., Zone 3), ensuring minimal travel time during dispatch.
- **Block Assignment**: Within each zone, individual blocks are used to categorize products by delivery date or customer, allowing faster retrieval and better organization.



This intelligent zone-block mapping minimizes manual handling, reduces delays, and ensures that high-priority items are always closer to the dispatch point. For instance, a product scheduled for delivery within 24 hours is immediately placed in a higher-priority zone with fewer intermediate steps, whereas low-priority items are moved to zones farther from the exit.
As the product's dispatch time approaches, the system cross-verifies its location and triggers alerts if movement is needed. The final scan in the dispatch zone updates the cloud database and marks the product as shipped. This structured layout allows for **automated, efficient, and scalable warehouse management** using low-cost hardware.

C. *Classification.*

In this system, classification is done based on the arrival and delivery times of each product. When a product enters the warehouse and is scanned using RFID, it is classified into three priority levels:

- **High Priority (Delivery within 24 hours):** Moved to **Zone-3**
- **Medium Priority (Delivery in 1–3 days):** Moved to **Zone-2**
- **Low Priority (Delivery after 3 days):** Moved to **Zone-1**

This classification ensures time-based scheduling and optimized storage. Each product's type (e.g., perishable, fragile) can also be considered for special handling. Once classified, all data is synced to **Google Firebase**, and the status is shown in real-time on the web dashboard.

D. *Time-based dispatch logic*

Once a product is classified and placed in its respective zone, the dispatch logic activates to manage **scheduled movement and timely delivery**. This logic continuously tracks the **remaining time to dispatch** using a real-time clock (RTC) module.
The system monitors each product's delivery deadline and initiates action as follows:

- **Pre-Dispatch Alerts** are generated as the deadline approaches.
- **High-priority items in Zone-3** are triggered for immediate movement once they enter the final hour of their delivery window.
- **Medium and low-priority items** are regularly evaluated. If their time window shifts closer, the system reclassifies them dynamically and can **reallocate zones** if necessary.

The logic not only governs storage but also automates **route selection and dispatch preparation**. When dispatch time is reached, the system validates the product using RFID and updates its status to **"Ready for Dispatch"**. The dispatch sequence is finalized and logged to **Google Firebase**, updating the live dashboard with real-time status.

This dynamic and autonomous scheduling process **reduces human error**, optimizes flow, and ensures **on-time delivery**, even in a low-cost warehouse environment.

E. *Use of ESP32 and RFID in automation*

The automation backbone of the proposed warehouse system relies on the integration of **ESP32 microcontrollers** and **RFID technology**. Each product entering the warehouse is tagged with an **RFID label** that stores its unique identification and delivery data.

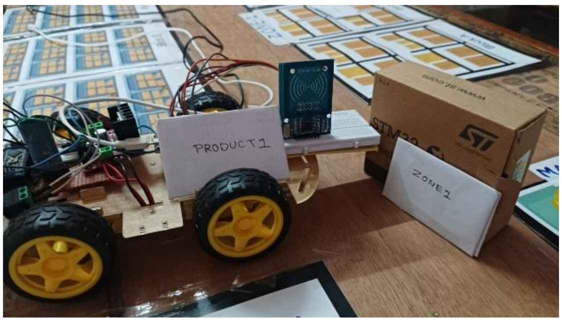
The **ESP32**, a low-cost microcontroller with built-in Wi-Fi, reads this RFID data upon entry using an RFID reader module. It then sends the data to **Google Firebase Cloud** for processing and live tracking. Based on the delivery deadline and arrival timestamp, the system automatically determines the storage zone.

Throughout the process, the ESP32 continues to monitor RFID tags placed at various zone entry and exit points. This allows it to detect when a product enters or leaves a zone and triggers updates in the Firebase database and web dashboard.

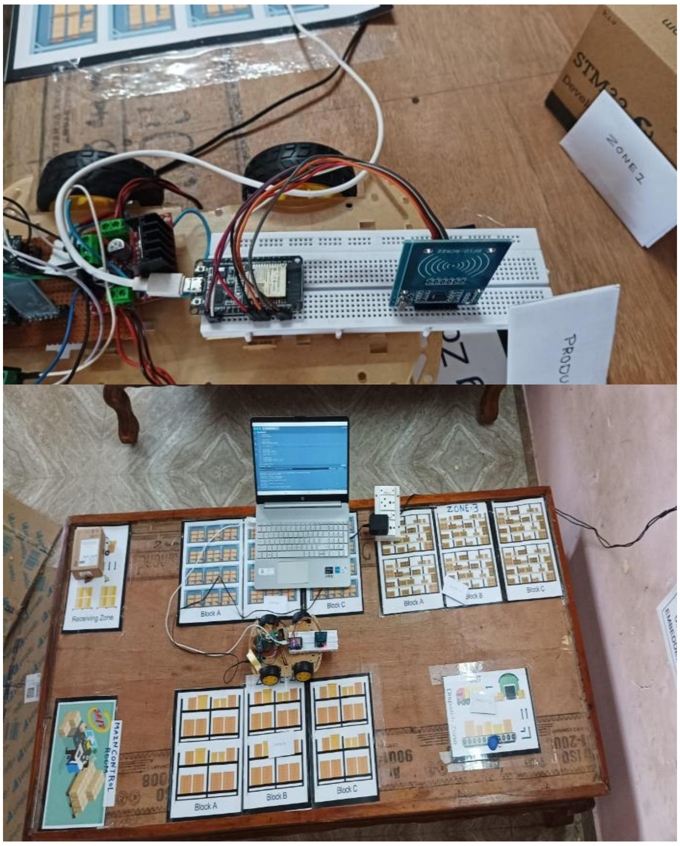
With its wireless capabilities, the ESP32 enables **real-time data flow**, decision-making, and system alerts without the need for expensive industrial hardware. This integration ensures accurate tracking, smooth zone transitions, and timely dispatch—all done autonomously with **minimal human intervention**.

This cost-effective solution highlights the potential of **IoT-based automation** in transforming traditional warehouses into **smart, time-efficient dispatch systems**.

F. *Role of priority zone*



Priority zones are essential for organizing and automating warehouse dispatch based on delivery urgency. The system assigns incoming products to specific zones using RFID and time-based rules: **Zone-3** for items needing dispatch within 24 hours, **Zone-2** for 1–3 days, and **Zone-1** for over 3 days. These zones are arranged to optimize movement, with high-priority zones placed closer to the dispatch area. As deadlines approach, items can be reclassified and shifted between zones. This dynamic zoning improves efficiency, ensures timely deliveries, and reduces manual intervention, creating a smart, responsive dispatch environment.



III. TECHNICAL REQUIREMENTS

The successful implementation of the Time-Based Priority Dispatch System requires a combination of low-cost hardware, simple software tools, and reliable network connectivity. The system uses ESP32 microcontrollers for data processing and communication, RFID readers and passive tags for product identification, and a Real-Time Clock (RTC) module to enable accurate time tracking. All product data is transmitted to Google Firebase, a real-time cloud database, via Wi-Fi. The control logic is developed using the Arduino IDE, while the web dashboard is built using basic web technologies integrated with Firebase. A stable 5V DC power supply is required for each ESP32 node, and the warehouse should maintain consistent Wi-Fi coverage to ensure seamless data synchronization.

A. *Hardware(ESP32, RFID reader/writer, tags, motors (if robotic pickers), sensors, relays)*

The core hardware components of the system include the ESP32 microcontroller, which offers built-in Wi-Fi and low power consumption, making it ideal for IoT applications. Each incoming product is identified using an RFID reader (RC522 or PN532) that scans passive RFID tags attached to the goods.

These tags store a unique ID along with product and delivery metadata. To enable future scalability, the system can also be extended with motors and robotic pickers for automated material movement, along with IR or ultrasonic sensors for presence detection and relays for controlling actuators or conveyors. This hardware combination supports a modular, affordable, and robust automation solution tailored for real-time warehouse operations.

B. *Software*

The software framework of the proposed system integrates both IoT development platforms and web technologies to ensure real-time, automated warehouse operations. The control and communication logic for the ESP32 microcontrollers is developed using the **Arduino IDE** and can be extended with **ESP-IDF** for advanced features. These

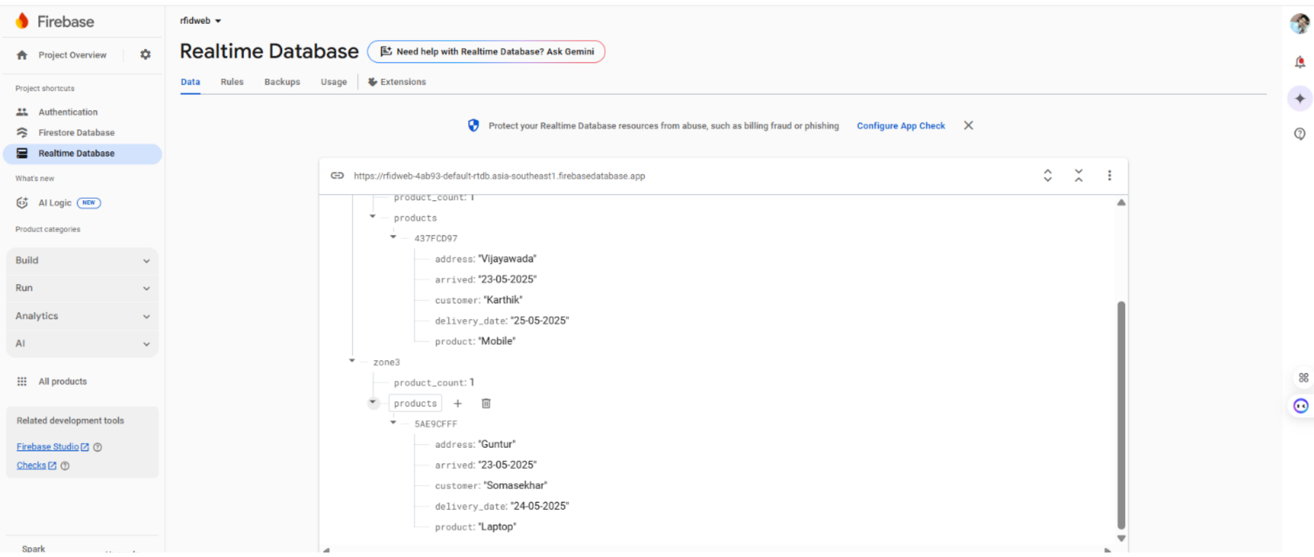
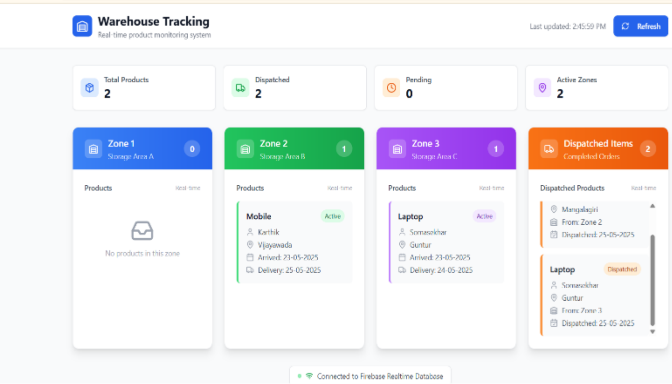


Figure 2 GOOGLE FIRE BASE LINK UP

platforms provide the necessary libraries for integrating **RFID modules**, **RTC**, and **Wi-Fi communication**.

For cloud connectivity, **Google Firebase Realtime Database** is used to store and sync product data, such as arrival time, delivery deadlines, and zone assignments. The **communication protocols** employed include **Wi-Fi** for cloud interaction and **UART** for local serial communication between components (e.g., RFID reader and ESP32).The dashboard interface is designed using **React.js** for the frontend and **Node.js** for backend logic, providing a highly responsive and user-friendly monitoring platform. The system architecture is compatible with **Warehouse Management Software (WMS)** models and can be integrated with open-source or custom WMS platforms. Real-time product tracking, zone status, and dispatch logs are continuously displayed through this web dashboard, offering warehouse managers actionable insights and complete operational visibility.



C. *Power Supply and Connectivity*

The system is designed to operate efficiently on low power while maintaining constant connectivity for real-time operations. Each ESP32 microcontroller requires a stable 5V DC power supply, which can be provided via USB adapters, battery packs, or regulated power modules. For scalability and mobile use, the system can be adapted to run on Li-ion battery packs with solar charging as a potential future upgrade for off-grid environments.

For connectivity, Wi-Fi is the primary communication channel, enabling the ESP32 modules to interact with Google Firebase for real-time data synchronization. A reliable local Wi-Fi network with minimal latency is crucial for seamless system performance. Within the hardware setup, UART (Universal Asynchronous Receiver/Transmitter) is used for serial communication between the ESP32 and peripheral devices such as RFID readers, RTC modules, and sensors. The combined use of efficient power delivery and robust wireless communication ensures consistent, real-time operation of the smart warehouse dispatch system.

IV. Available Technologies & Cost Analysis

To develop a cost-effective and scalable warehouse automation system, various technologies and components were evaluated based on functionality, cost, and ease of integration. The aim was to minimize overall expenses while ensuring high performance and reliability.

A. *Justification for "low-cost" hardware choices.*

The hardware components selected for this system were chosen with a strong emphasis on affordability, availability, and functional efficiency. The **ESP32 microcontroller** offers built-in Wi-Fi, low power consumption, and enough GPIO pins for sensor and RFID integration—all at a significantly lower cost compared to alternatives like Raspberry Pi or industrial controllers. The **RC522 RFID reader** is compact, cost-effective, and reliable for short-range scanning

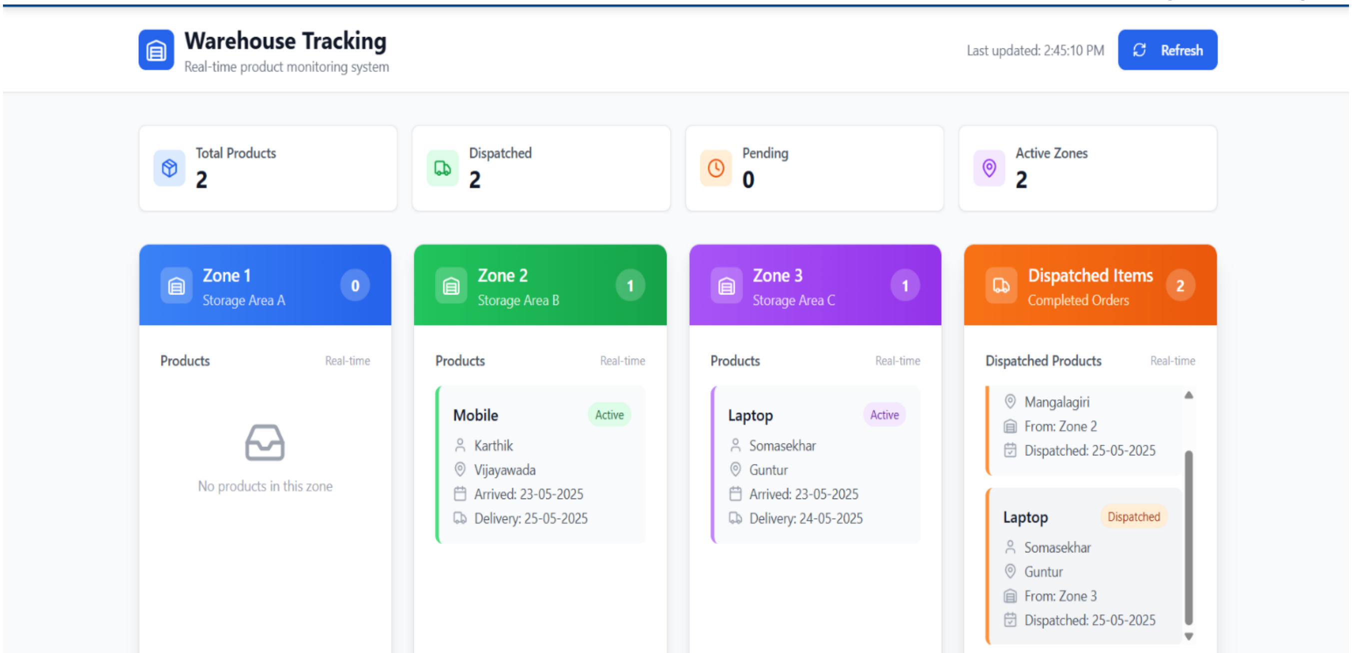


Figure 3 the web-based dashboard provides a clear and dynamic visualization of active zones, product movement, and dispatch history in real-time.

applications, making it suitable for warehouse entry and exit tracking.

Component	**Estimated Cost (INR)**	**Quantity**	**Total Cost (INR)**
ESP32 Dev Board	₹300	1	₹300
RFID Reader (RC522)	₹150	1	₹150
Passive RFID Tags	₹20	10	₹200
Power Supply & Adapter	₹150	1	₹150
Misc. (Cables, PCB, Breadboard)	₹200	1	₹200
Wi-Fi (shared mobile hotspot / existing)	₹0	1	₹0
Total Estimated Cost			**₹1,000**

Instead of investing in commercial Warehouse Management Systems (WMS) or industrial-grade automation kits, the system utilizes a **custom-built dashboard using open-source tools (React.js and Firebase)**. This removes licensing costs and allows full control over customization. Additionally, components like passive RFID tags, USB-based power supplies, and readily available jumper cables or PCBs help keep the total prototype cost within ₹1,000—demonstrating a scalable and practical solution for small to mid-sized warehouses with limited budgets.

V. Chosen Technology & Justification

A. *Benefits (e.g., Wi-Fi support, cost, ease of integration)*

The proposed system offers several key benefits that make it suitable for low-cost warehouse automation. The ESP32 microcontroller includes built-in Wi-Fi, enabling real-time communication with Firebase without additional hardware. This allows continuous data synchronization and live dashboard updates. The components used—ESP32, RC522 RFID, RTC module, and passive tags—are inexpensive and easily available, keeping the total system cost under ₹1,000.

Ease of integration is ensured through modular hardware, allowing future upgrades such as robotic pickers or additional sensors. Open-source software platforms like Arduino IDE and Firebase further reduce costs. The system supports dynamic scheduling using real-time clocks, reducing human error. Its scalability and remote monitoring features make it a practical and flexible solution for modernizing small to medium-sized warehouses.

B. *Why not other methods (e.g., QR code, manual barcode scanners, image processing)*

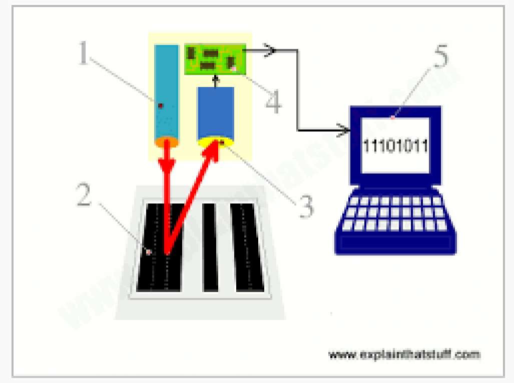


Figure 4 OR CODE METHOD

Alternative methods such as QR codes, manual barcode scanners, and image processing were considered but ultimately not selected due to their limitations in real-time automation and cost-efficiency. QR codes and barcodes, while inexpensive, require manual scanning and line-of-sight visibility, making them inefficient for high-volume, time-sensitive operations. Image processing techniques, though capable of automation, demand high-performance hardware, controlled lighting conditions, and complex software, significantly increasing the system's cost and complexity. In contrast, RFID technology with ESP32 microcontrollers enables fast, contactless, and autonomous tracking without manual effort, making it a more suitable choice for low-cost, scalable warehouse automation.

C. *Comparison of alternatives (brief pros & cons)*

Several alternative technologies were evaluated for inventory tracking and dispatch automation. QR codes and manual barcode systems are low-cost and simple to implement but require line-of-sight and human intervention, leading to slower operations and increased chances of error. Image processing offers advanced automation and accuracy but is expensive and computationally intensive, making it unsuitable for low-budget applications. In contrast, RFID-based systems provide a balance between cost and functionality. They offer contactless, fast scanning without the need for visual alignment, support real-time tracking, and are easily integrated with microcontrollers like ESP32. This makes RFID the most practical and scalable option for the proposed warehouse dispatch system.

VI. Implementation Feasibility & Success Factors

A. How you plan to implement it

The implementation of the proposed dispatch system is highly feasible due to its use of low-cost, readily available components and straightforward integration. ESP32 microcontrollers, RFID readers, RTC modules, and Firebase cloud services form a lightweight yet powerful infrastructure that can be rapidly deployed in existing warehouse environments. The control logic, developed using the Arduino IDE, and the web dashboard, built with open-source technologies, ensure ease of programming and customization. Key success factors for reliable operation include consistent Wi-Fi connectivity for real-time data synchronization, accurate RFID tag placement for proper identification, and a well-structured storage layout with clearly defined zones. Additionally, the system's priority-based scheduling logic enables efficient space utilization and timely dispatch, ensuring improved workflow, reduced human error, and scalability for future expansion.

B. Factors for successful operation (network reliability, tag accuracy, shelf arrangement, etc.)

The successful operation of the proposed system depends on several critical factors. Reliable and uninterrupted Wi-Fi connectivity is essential for maintaining real-time communication between the ESP32 nodes and the Firebase cloud database. Accurate placement and scanning of RFID tags are crucial to ensure correct identification and tracking of products throughout the warehouse. Additionally, the warehouse layout must support a clear and logical zoning structure, with shelves and blocks arranged to reflect the priority-based dispatch logic. Proper labeling, zone proximity to dispatch points, and minimal obstruction in product flow further enhance efficiency. Power stability for ESP32 modules and regular maintenance of RFID readers also contribute to uninterrupted operation and long-term system reliability.

C. Storage algorithm logic (FIFO or priority-based).

The proposed system employs a **priority-based storage algorithm** rather than the conventional First-In-First-Out (FIFO) approach to enhance delivery efficiency. Each product is classified based on its delivery deadline at the time of entry and assigned to a corresponding priority zone. High-priority items (delivery within 24 hours) are stored closest to the dispatch point, while medium and low-priority items are placed in zones progressively farther away. This dynamic approach ensures that time-sensitive goods are dispatched promptly, minimizing delays. Unlike FIFO, which operates solely on arrival order, the priority-based method adapts to real-time changes in urgency, allowing reclassification and reallocation of storage zones as needed. This logic enables optimal space utilization and supports timely, intelligent dispatch operations.

VII. Alternative Methods Considered

A. Barcode scanning systems

Barcode scanning is one of the most widely used inventory tracking systems in conventional warehouses. It uses printed labels with data encoded in bar patterns, which are scanned using handheld or fixed barcode readers. This method is relatively low-cost and easy to implement.

However, barcode systems rely on **manual handling** and **line-of-sight visibility**, which limits their effectiveness in fast-paced or high-volume environments. Additionally, human error in scanning or label damage can lead to inaccuracies in inventory tracking.

B. Vision-based inventory tracking.

Vision-based systems employ cameras and computer vision algorithms to identify and track products. This method enables non-contact, high-throughput inventory analysis and can handle complex tasks such as object detection, damage identification, and product counting. However, such systems demand high-performance processors, controlled lighting environments, and robust software, significantly increasing development and operational costs. Moreover, integration with low-cost microcontrollers like ESP32 is non-trivial, limiting its practicality for small-scale or budget-constrained warehouses.

C. Manual entry systems.

Manual data entry involves workers recording product information using paper logs, spreadsheets, or simple software tools. While this method requires minimal technological investment, it is highly error-prone, time-consuming, and inefficient. Manual systems offer no real-time visibility, are difficult to scale, and heavily depend on human accuracy and diligence, making them unsuitable for modern supply chain demands.

D. Why these were not selected.

The primary reason for not selecting barcode, vision-based, or manual systems is the lack of autonomous operation and scalability within a low-cost environment. Barcode and manual systems require continuous human interaction, which limits throughput and introduces delays. Vision systems, although powerful, are not viable due to high costs and complexity. The proposed RFID-ESP32-based system overcomes these limitations by offering low-cost, autonomous, and real-time tracking capabilities without sacrificing performance.

E. Pros and cons of each method

Method	Pros	Cons
Barcode Scanning	Low cost, well established, simple to implement	Requires manual scanning, line-of-sight, not autonomous

Method	Pros	Cons
Vision-Based Tracking	High accuracy, automation possible, handles complex tasks	Expensive, computationally heavy, lighting dependent
Manual Entry	No hardware cost, simple setup	Error-prone, slow, no real-time tracking, non-scalable
Proposed RFID-ESP32	Autonomous, low cost, real-time data sync, scalable	Limited range, requires initial tag setup

VIII. CONCLUSION AND FUTURE SCOPE

In this paper, a Time-Based Priority Dispatch System for warehouses has been proposed and implemented using low-cost embedded hardware, specifically the ESP32 microcontroller, RFID technology, and cloud integration via Google Firebase. The system offers real-time inventory tracking, automated classification based on delivery deadlines, and intelligent zone allocation to optimize dispatch operations. By automating manual tasks and utilizing dynamic priority logic, the proposed solution significantly enhances efficiency, reduces human error, and lowers operational costs for small and medium-sized warehouses.

The success of the system lies in its simplicity, affordability, and scalability. With a total prototype cost of under ₹1,000, it demonstrates that modern warehouse automation does not require expensive industrial setups. The integration of open-source platforms like Arduino IDE and Firebase also contributes to flexibility and ease of customization.

REFERENCES

[1] [1] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of Things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, Sep. 2012.

[2] [2] A. S. Dobkin and R. Weigand, "Using RFID for warehouse management and logistics," *IEEE Potentials*, vol. 30, no. 4, pp. 36–39, Jul.–Aug. 2011.

[3] [3] J. W. Yoon, Y. J. Lee, and H. S. Kim, "Design and implementation of an IoT-based warehouse monitoring system," *IEEE Access*, vol. 7, pp. 61700–61710, 2019.

[4] [4] A. Ghosh, R. Bandyopadhyay, and M. Roy, "A low-cost IoT-based smart inventory system using RFID," in *Proc. IEEE Int. Conf. Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2112–2116, 2019.

[5] [5] S. Patel, K. Patel, and V. Mehta, "Warehouse automation using ESP32 and Firebase," in *Proc. IEEE Int. Conf. Smart Technologies for Smart Nation (SmartTechCon)*, pp. 1000–1005, 2017.

[6] [6] M. A. Matin and M. M. Islam, "Overview of RFID technology," in *Proc. IEEE Int. Conf. Telecommunication Networks and Applications Conference*, pp. 1–6, 2006.

[7] [7] J. Zhang and Y. Zhang, "Real-time cloud-based RFID system for smart logistics," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1493–1502, May 2014.

[8] [8] B. Bhattacharyya, T. Ghosh, and S. Dey, "Design and implementation of smart inventory system using Firebase and RFID," in *Proc. IEEE Int. Conf. Emerging Trends in Engineering and Technology (ICETET)*, pp. 56–60, 2020.

[9] [9] M. Aazam, S. Zeadally, and K. A. Harras, "Deploying fog computing in industrial internet of things and smart manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4674–4682, Oct. 2018.

[10] [10] Y. Lu and K. Cecil, "RFID-enabled real-time manufacturing execution system for intelligent warehouse management," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 10, pp. 1340–1353, Oct. 2015.