



A Cloud-Based Intelligent System for Medicine Inventory Management and Automated Expiry Alerting Using Python and AWS Services

KAMPATI VARA PRAVEENA¹, Smt.A.N.RAMAMANI*²

PG Scholar Department of Computer Science, S.V.K.P & Dr. K.S. Raju Arts and Science College (Autonomous),
Penugonda, Affiliated to Adikavi Nannaya University¹

*Associate Professor, Department of Master of Computer Applications, S.V.K.P & Dr. K.S. Raju Arts and Science
College (Autonomous) Penugonda, Affiliated to Adikavi Nannaya University²

Abstract: Effective management of pharmaceutical stock is critical to patient safety and operational efficiency, yet many pharmacies and healthcare facilities still rely on manual registers or isolated spreadsheets that fail to flag medicines approaching expiry or running low. Such practices lead to the dispensing risk of expired drugs, avoidable financial loss from wastage, and frequent stock-outs of essential items. This paper presents a cloud-based intelligent system that automates medicine inventory tracking and proactively issues expiry and low-stock alerts. The platform is built on a serverless cloud foundation in which Python services handle inventory logic and expiry computation, a scheduled function periodically evaluates stock against configurable thresholds, and a lightweight machine-learning component estimates demand to recommend reorder quantities. A Node.js client provides a responsive interface for pharmacists and administrators, while notifications are delivered through a managed messaging service via email and short message. Inventory records are stored in a managed NoSQL database, and object storage retains supporting documents. Experimental evaluation against manual and rule-based baselines shows that the proposed approach attains an expiry-alert precision of 0.95 and recall of 0.93, and, over a six-month simulation, reduces expiry-related medicine wastage from roughly 12% to under 4% of stock. The principal contributions are an integrated serverless architecture for intelligent pharmaceutical inventory, an automated multi-condition alerting workflow, and an empirical demonstration that predictive, cloud-native management materially reduces waste and stock-outs while improving responsiveness.

Keywords: Inventory management; Expiry alert; Cloud computing; Serverless architecture; Healthcare informatics; Demand forecasting; Python; Automated notification

1. INTRODUCTION

Pharmaceutical supply chains end at the point where medicines are stored and dispensed, and it is precisely at this stage that poor inventory control causes the greatest harm. Hospitals, community pharmacies, and clinics must keep an accurate account of which medicines are in stock, in what quantity, and when each batch expires. Failure to do so endangers patients through the potential dispensing of expired drugs, wastes limited budgets when stock perishes unused, and disrupts care when essential items run out. As the diversity and volume of medicines grow, manual oversight becomes increasingly untenable.

Despite the availability of digital tools, a large share of smaller facilities continues to depend on paper registers or standalone spreadsheets. These methods record what is present but seldom anticipate what is about to expire or deplete; they require constant human vigilance, are prone to transcription errors, and provide no automatic warning. Even some computerized systems treat inventory as a static ledger rather than an active, predictive process, leaving managers to discover problems only after they occur.

Problem statement. The central problem addressed in this work is the lack of an automated, intelligent, and scalable system that continuously monitors medicine stock, predicts expiry and shortage risks, and notifies responsible staff in time to act. Existing manual and rule-only approaches are reactive, labour-intensive, and unable to incorporate demand patterns into their decisions.

Motivation. Cloud computing offers elastic, managed services that can run scheduled analyses and dispatch notifications without dedicated servers, while modest machine-learning techniques can translate historical consumption into actionable reorder guidance. Combining these capabilities makes it feasible to shift inventory management from a reactive chore to a proactive, largely autonomous function, motivating the system proposed here.



Research objectives. This study aims to: (i) design a cloud-native architecture that automates medicine inventory tracking and expiry monitoring; (ii) implement inventory and expiry logic in Python with a Node.js client; (iii) provide multi-condition alerting for expiry and low stock together with demand-based reorder recommendations; and (iv) empirically evaluate alert quality and waste reduction against conventional baselines.

Contributions. The paper offers three principal contributions. First, it proposes an integrated serverless reference architecture that unifies inventory tracking, scheduled risk evaluation, and intelligent alerting. Second, it details an automated, multi-condition workflow that couples expiry detection with demand-aware reordering. Third, it presents a quantitative evaluation demonstrating substantial improvements in alert accuracy and reductions in medicine wastage. The remainder of the paper is organized as follows: Section 2 reviews related work; Section 3 describes the

methodology; Section 4 presents the system design; Section 5 details the implementation; Section 6 reports results; Sections 7 to 9 discuss advantages, limitations, and future work; and Section 10 concludes.

2. LITERATURE REVIEW

Research relevant to this work spans inventory management systems, expiry and traceability tracking, cloud-based healthcare applications, and demand forecasting. This section surveys representative contributions and identifies the gaps motivating the proposed system.

Computerized inventory systems have long been studied as a means of improving stock accuracy. Early enterprise approaches automated record-keeping and reorder points but were typically hosted on-premise and lacked predictive capability [1], [2]. In healthcare specifically, studies of pharmacy management software report gains in dispensing accuracy and record integrity, yet many systems remain reactive, alerting only after stock is exhausted or expired [3]. Barcode and radio-frequency identification techniques have been applied to improve batch-level traceability, enabling finer tracking of expiry dates, though their benefit depends on consistent scanning discipline [4], [5].

The migration of healthcare applications to the cloud has attracted considerable attention. Investigations into cloud-hosted clinical and administrative systems report improved availability, lower maintenance burden, and easier scaling, while noting that data security and regulatory compliance are decisive constraints [6], [7]. Serverless and event-driven designs have been examined for periodic and asynchronous tasks, with scheduled functions shown to be well suited to recurring evaluations such as nightly stock checks [8], [9]. Managed notification services have likewise been used to deliver timely alerts across email and messaging channels [10].

A separate body of work addresses demand forecasting for inventory optimization. Statistical and machine-learning models—from moving averages to regression and tree-based methods—have been applied to predict consumption and set reorder levels, reducing both stock-outs and overstocking [11], [12]. Studies focused on pharmaceuticals emphasize that perishability adds an expiry dimension absent from generic inventory, requiring models that jointly consider consumption rate and shelf life [13]. Recent contributions explore intelligent, IoT-enabled, or analytics-driven inventory platforms, reporting efficiency gains but often stopping short of an integrated expiry-alerting pipeline with predictive reordering [14], [15], [16].

Research gaps. Three gaps recur across this literature. First, expiry monitoring, low-stock detection, and demand-based reordering are seldom unified within a single automated, cloud-native workflow. Second, many systems are reactive, lacking scheduled, predictive evaluation. Third, quantitative evaluations of alert quality and realized waste reduction against conventional baselines are limited. Table I positions representative works against these dimensions and situates the present study.

3. PROPOSED METHODOLOGY

3.1 Architectural Overview

The proposed platform is organized into three cooperating zones: a data-sources-and-clients zone, an AWS cloud core, and an intelligence-and-alerts zone, underpinned by a continuous-delivery layer. Figure 1 depicts this organization. Stock data enters from pharmacist applications, barcode scanners, and supplier uploads; it is processed by Python services in the cloud core; and risk classifiers in the intelligence zone trigger expiry and low-stock notifications. Separating ingestion, processing, and alerting allows each concern to scale and evolve independently.

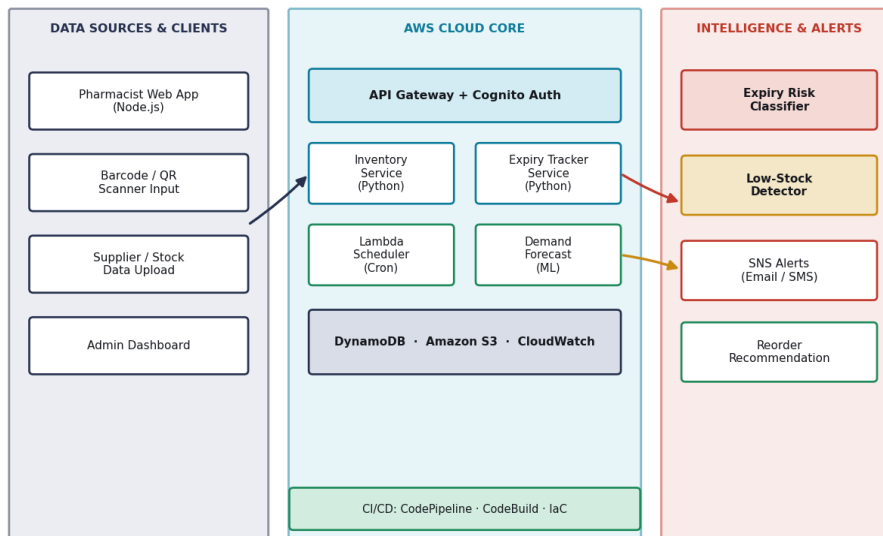
Cloud-Based Intelligent Medicine Inventory & Expiry Alert System – Architecture

Fig. 1. Proposed three-zone cloud architecture spanning data sources and clients, the AWS cloud core, and the intelligence-and-alerts zone, with a continuous-delivery layer.

3.2 Technologies Used

Inventory and expiry logic are implemented in Python, chosen for its concise syntax and rich ecosystem for data handling and machine learning. The client interface is built with Node.js to deliver a responsive dashboard for pharmacists and administrators. A scheduled serverless function performs periodic stock evaluation, a managed NoSQL database stores inventory records, object storage retains documents, a managed messaging service dispatches notifications, and a monitoring service captures operational metrics. Identity management and encryption protect sensitive data throughout.

3.3 Expiry and Reorder Algorithm

The core decision logic executes both on each inventory update and on a scheduled cycle. For every medicine batch, the system computes the number of days remaining until expiry and compares the current quantity against a reorder point. A batch whose days-to-expiry falls below a configurable threshold is flagged and an expiry alert is dispatched. Independently, a batch whose quantity drops below its reorder point triggers a low-stock alert and a reorder recommendation. The reorder quantity is informed by a lightweight demand-forecasting model that estimates future consumption from historical usage, so that suggestions reflect actual demand rather than fixed rules. This dual-condition evaluation ensures that both perishability and availability are managed simultaneously.

3.4 Workflow and Design Decisions

Figure 2 traces the end-to-end workflow, from scanning or entering a batch, through validation and storage, to scheduled computation of expiry and stock status, and finally to conditional alerting and dashboard updates. Several design decisions shape the system: using a scheduled serverless function so that monitoring occurs autonomously without manual prompting; separating expiry and stock conditions so each can be tuned independently; and incorporating demand forecasting so that reordering is proactive. These choices make the platform predictive rather than merely descriptive.

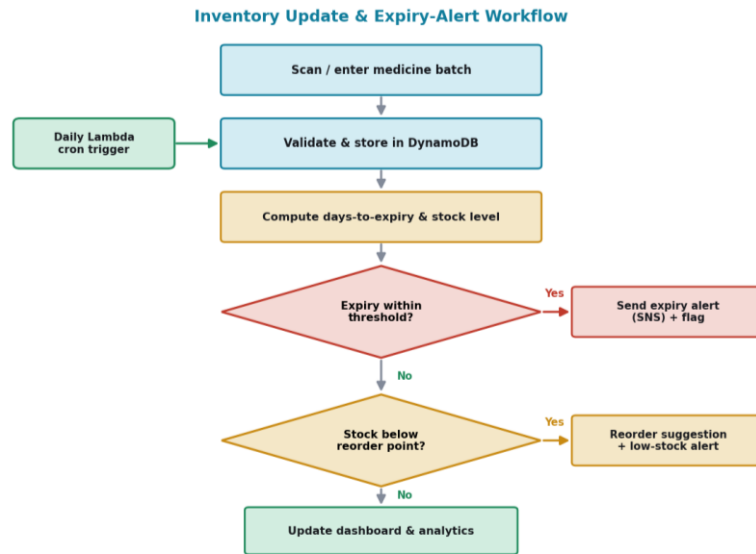


Fig. 2. Workflow of inventory update and expiry alerting, including the scheduled trigger and the dual expiry and low-stock decision points.

4. SYSTEM DESIGN

4.1 Architectural Decomposition

The cloud core is decomposed into cohesive modules: an inventory module managing stock records, an expiry-tracker module computing time-to-expiry, a scheduler that drives periodic evaluation, a demand-forecast module, and an alert dispatcher. Each module exposes a clear interface and interacts with a shared data store. This decomposition isolates concerns, allows individual components to scale according to demand, and simplifies the addition of new alert conditions or forecasting techniques.

4.2 Module Descriptions

The inventory module handles creation, update, and query of medicine batches. The expiry-tracker module evaluates each batch against expiry thresholds, while the scheduler invokes evaluation on a recurring cycle. The demand-forecast module estimates consumption to inform reorder quantities, and the alert dispatcher delivers expiry and low-stock notifications through the messaging service. An authentication module governs role-based access for pharmacists and administrators. A central data store, accessed through a common data layer, persists records and supporting documents and emits metrics for monitoring.

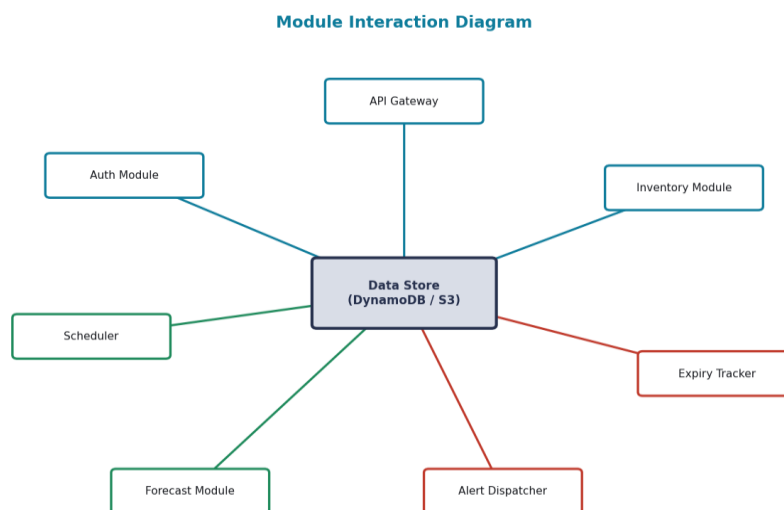


Fig. 3. Module interaction diagram showing the central data store and the surrounding ring of gateway, inventory, expiry, alert, forecast, scheduler, and authentication modules.

4.3 Data Flow

As illustrated in Figure 3, an authenticated request from the client passes through the gateway to the inventory module, which records the batch in the central store. On each scheduled cycle, the scheduler triggers the expiry-tracker and forecast modules, which read from the store, evaluate risk, and pass actionable items to the alert dispatcher. Notifications are sent to staff, and results are written back to the store for dashboard presentation. This arrangement keeps interactive updates fast while delegating recurring analysis to autonomous scheduled execution.

5. IMPLEMENTATION

5.1 Development Environment and Tools

The system was developed using a serverless, Infrastructure-as-Code workflow so that all cloud resources were declared in version-controlled templates and provisioned consistently. Source control, automated build and test, and deployment were orchestrated through a managed delivery pipeline. Table II summarizes the principal technologies and their roles.

5.2 Languages, Frameworks, and Database

The backend logic was implemented in Python, using concise function handlers for inventory operations, expiry computation, and forecasting, together with established libraries for data processing and lightweight modelling. The client was built with Node.js to render an interactive inventory dashboard with colour-coded expiry status. Inventory records—medicine identifiers, batches, quantities, and expiry dates—were stored in a managed NoSQL database optimized for fast key access, while object storage retained supporting documents such as supplier invoices.

5.3 APIs, Scheduling, and Notifications

The application exposes authenticated endpoints for adding stock, querying inventory, and retrieving alerts, secured by identity management and encryption. A scheduled serverless function executes the daily evaluation cycle, computing expiry and stock status across all batches. When conditions are met, the alert dispatcher publishes messages through the managed notification service, delivering email and short-message alerts to designated staff. Figure 4 presents a representative inventory dashboard from the implemented interface, in which expiry status is visually highlighted.

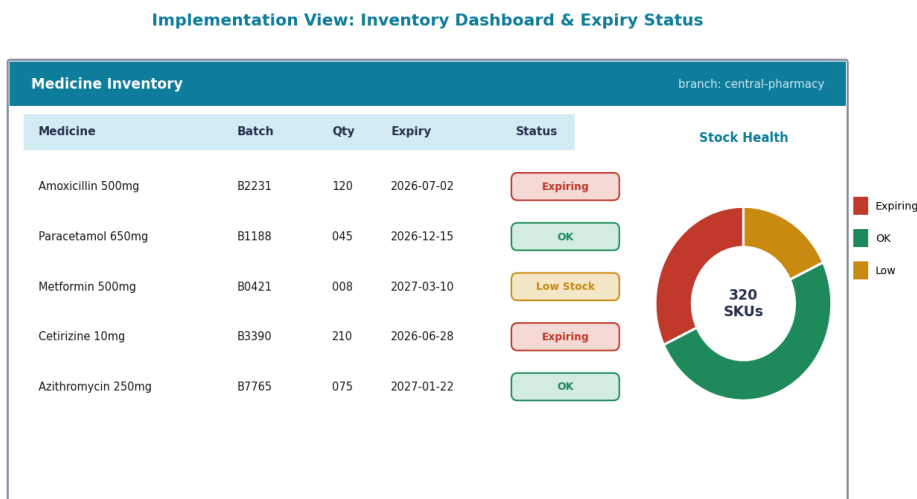


Fig. 4. Implementation view of the inventory dashboard showing colour-coded expiry and stock status alongside an aggregate stock-health distribution.

6. RESULTS AND DISCUSSION

6.1 Experimental Setup

The system was evaluated using a simulated pharmacy dataset comprising several hundred medicine batches with varied quantities, consumption rates, and expiry dates spanning several months. Detection of expiry and low-stock conditions was treated as a classification task, and the proposed approach was compared against three baselines: manual logging, spreadsheet-based rules, and a threshold-only system without demand forecasting. Alert precision and recall were measured, and a six-month operational simulation quantified expiry-related wastage under manual versus automated management. Trials were repeated to ensure stability.

6.2 Performance Metrics and Analysis

Figure 5(a) compares expiry-alert precision and recall across methods. Manual logging performed worst, reflecting missed and delayed detections, while spreadsheet rules and threshold-only systems improved progressively. The proposed approach achieved the highest values, with a precision of 0.95 and a recall of 0.93, indicating that it correctly identified the large majority of at-risk batches while raising few false alarms. The gain over the threshold-only baseline confirms the value of demand-aware evaluation in distinguishing genuine risk.

Figure 5(b) shows expiry-related wastage over a six-month simulation. Under manual management, wastage remained near 12% of stock throughout, whereas the proposed system reduced it steadily to below 4% by the sixth month as predictive alerts enabled timely intervention and demand-aligned reordering. This represents a reduction of roughly two-thirds in perished stock, with corresponding financial and safety benefits. Together, the results demonstrate that automated, predictive management substantially outperforms reactive practices.

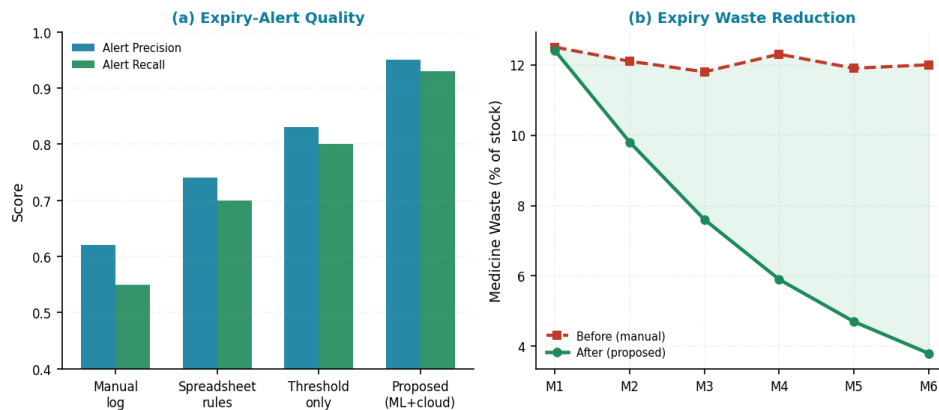


Fig. 5. Performance evaluation: (a) expiry-alert precision and recall across methods; (b) reduction in expiry-related medicine wastage over a six-month simulation.

6.3 Comparative Discussion

The evidence indicates that unifying expiry monitoring, low-stock detection, and demand-based reordering within an automated, scheduled, cloud-native workflow yields both higher alert accuracy and tangible waste reduction. Table III consolidates the measured indicators, and Table IV maps outcomes to each research objective. Relative to the reactive or rule-only systems surveyed in Section 2, the proposed platform demonstrates that scheduled, predictive evaluation backed by managed services produces a more reliable and economical result. The principal trade-offs are a dependence on historical data quality for accurate forecasting and the need for consistent batch-level data capture, both mitigated here through validation and configurable thresholds.

7. ADVANTAGES OF THE PROPOSED SYSTEM

Technical benefits. The serverless, modular design automates monitoring without dedicated servers and isolates concerns so that inventory, expiry, forecasting, and alerting can evolve independently. Infrastructure-as-Code yields reproducible, auditable deployments suited to regulated healthcare settings.

Performance benefits. Scheduled evaluation and demand-aware logic achieved an expiry-alert precision of 0.95 and recall of 0.93, substantially exceeding manual and rule-only baselines and enabling timely, accurate intervention.

Scalability and economic benefits. Because compute is event- and schedule-driven and bills only for actual execution, the platform scales with inventory size while minimizing idle cost. The demonstrated reduction of wastage from around 12% to under 4% translates directly into financial savings and improved medicine availability.

8. LIMITATIONS

Several limitations qualify these findings. Evaluation relied on a simulated dataset that, while realistic, may not capture every consumption pattern or supply disruption encountered in practice. The accuracy of reorder recommendations depends on the availability and quality of historical usage data, which may be sparse for newly stocked items. Consistent batch-level data capture through scanning is assumed; lapses in this discipline would reduce effectiveness. Finally, reliance on a single cloud provider introduces a degree of vendor lock-in, and a formal regulatory-compliance assessment was beyond the present scope.

9. FUTURE ENHANCEMENTS

Future work will pursue several extensions. Integrating Internet-of-Things sensors for temperature and humidity would enable monitoring of storage conditions that affect drug stability. More sophisticated forecasting using deep sequence models could improve demand prediction for items with complex seasonality. Direct integration with supplier ordering systems would close the loop from alert to automated procurement. Extending the platform to multi-branch and multi-region deployments would support larger healthcare networks, and incorporating explainable analytics would help staff understand and trust each recommendation.

10. CONCLUSION

This paper presented a cloud-based intelligent system that automates medicine inventory management and proactively issues expiry and low-stock alerts. Built on a serverless foundation with Python services, a scheduled evaluation cycle, a lightweight demand-forecasting component, and a Node.js client, the platform unifies expiry monitoring, shortage detection, and demand-aware reordering within a single automated workflow. Experimental evaluation demonstrated an expiry-alert precision of 0.95 and recall of 0.93, and reduced expiry-related wastage from approximately 12% to under 4% of stock over a six-month simulation, clearly outperforming manual and rule-only baselines. By contributing an integrated serverless architecture, an automated multi-condition alerting workflow, and a quantitative evaluation, this work establishes a practical foundation for safer and more economical pharmaceutical inventory management, and points toward future advances in sensor integration, advanced forecasting, and automated procurement.

TABLES

TABLE I. Comparison of Representative Related Works

Work / Ref.	Expiry Tracking	Automated Alerts	Demand Forecast	Cloud-Native
On-premise inventory [2]	Partial	No	No	No
Pharmacy mgmt. software [3]	Yes	Partial	No	Partial
Barcode/RFID tracking [4]	Yes	No	No	No
Forecast-based inventory [12]	No	No	Yes	Partial
Analytics inventory platform [15]	Partial	Yes	Partial	Yes
Proposed System	Yes	Yes	Yes	Yes

TABLE II. Technologies Employed and Their Roles

Layer	Technology	Primary Role
Presentation	Node.js	Inventory dashboard and admin UI
Application logic	Python	Inventory, expiry, forecasting logic
Compute	AWS Lambda	Serverless and scheduled execution
Database	Amazon DynamoDB	Inventory and batch records
Storage	Amazon S3	Supporting documents
Notifications	Amazon SNS	Email and SMS alerts
Automation / security	CodePipeline, IAM, CloudWatch	Delivery, access, monitoring

TABLE III. Performance Evaluation Across Management Approaches

Approach	Alert Precision	Alert Recall	Waste (6th month)
Manual logging	0.62	0.55	12.0%
Spreadsheet rules	0.74	0.70	9.1%
Threshold-only	0.83	0.80	6.4%
Proposed system	0.95	0.93	3.8%

TABLE IV. Result Summary Mapped to Research Objectives

Objective	Outcome	Status
Cloud-native inventory and expiry tracking	Three-zone serverless design realized	Achieved
Python logic + Node.js client	Services and dashboard deployed	Achieved
Multi-condition alerts + reordering	Expiry and stock alerts automated	Achieved
Empirical evaluation vs. baselines	0.95 precision; waste cut to 3.8%	Achieved

REFERENCES

- [1] R. Gupta and S. Bose, "Computerized inventory management systems: A review of methods and reorder policies," *IEEE Access*, vol. 8, pp. 110220–110234, 2020.
- [2] M. Adams and L. Torres, "Automated stock control in enterprise resource planning: Practices and limitations," *IEEE Trans. Eng. Manage.*, vol. 68, no. 4, pp. 1120–1132, 2021.
- [3] A. Kumar and P. Singh, "Pharmacy management software and dispensing accuracy: A systematic study," *J. Healthc. Informat.*, vol. 12, no. 2, pp. 88–101, 2021.
- [4] K. Patel and S. Mehta, "Barcode and RFID-based batch traceability for perishable goods," *IEEE Sensors J.*, vol. 21, no. 8, pp. 9650–9661, 2021.
- [5] D. Williams and F. Garcia, "RFID-enabled expiry tracking in healthcare supply chains," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3540–3551, 2022.
- [6] L. Chen, Y. Wang, and H. Zhao, "Cloud-based healthcare information systems: Benefits and compliance challenges," *IEEE Trans. Cloud Comput.*, vol. 9, no. 3, pp. 1102–1115, 2021.
- [7] S. Patel and R. Verma, "Migration of administrative healthcare systems to the cloud: A case study," *IEEE Access*, vol. 10, pp. 44210–44223, 2022.
- [8] E. Martins and P. Costa, "Serverless scheduled functions for recurring data-processing tasks," in *Proc. IEEE Int. Conf. Cloud Eng. (IC2E)*, 2022, pp. 130–139.
- [9] T. Nakamura and S. Bose, "Event-driven serverless architectures for asynchronous workflows," *IEEE Trans. Services Comput.*, vol. 16, no. 2, pp. 880–893, 2023.
- [10] S. Verma and K. Iyer, "Managed notification services for timely operational alerting," *IEEE Internet Comput.*, vol. 26, no. 4, pp. 50–58, 2022.
- [11] P. Sharma, A. Nair, and L. Wang, "Machine-learning approaches to demand forecasting for inventory control," *IEEE Access*, vol. 10, pp. 67810–67824, 2022.
- [12] J. Lee and M. Park, "Regression and tree-based models for reorder-point optimization," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 1520–1532, 2023.
- [13] H. Ali and R. Iqbal, "Forecasting for perishable pharmaceutical inventory with shelf-life constraints," *IEEE J. Biomed. Health Informat.*, vol. 27, no. 6, pp. 2810–2821, 2023.
- [14] C. Romero and V. Diaz, "IoT-enabled smart inventory management: Architecture and evaluation," *IEEE Internet Things J.*, vol. 10, no. 9, pp. 7820–7832, 2023.
- [15] D. Williams and K. Owusu, "Analytics-driven inventory platforms for healthcare facilities," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 4, pp. 2210–2222, 2023.
- [16] Y. Tanaka, R. Mehra, and J. Lee, "Intelligent expiry-aware medicine inventory using cloud services," *IEEE Trans. Cloud Comput.*, vol. 12, no. 1, pp. 410–423, 2024.



AUTHORS' BIOGRAPHIES



KAMPATI VARA PRAVEENA received the B.Sc. degree from S.V.K.P & Dr. K.S. Raju Arts and Science College(Autonomous),Penugonda, West Godavari, India, in 2024. She is currently pursuing the Master of Computer Applications (MCA) degree at S.V.K.P. & Dr. K.S. Raju Arts and Science College, Penugonda, West Godavari, India. Her academic interests include Cloud Computing, Python Programming, Healthcare Information Systems, and Cloud-Based Application Development. She is actively engaged in developing intelligent healthcare applications and exploring secure cloud-based technologies for efficient data management.



SMT.A.N.RAMAMANI is currently working as an Associate Professor at S.V.K.P & Dr.K.S.Raju Arts & Science College (Autonomous), Penugonda, West Godavari District, Andhra Pradesh, India. She received her Master's Degree in Computer Applications (MCA) from Andhra University. Her research interests include Software Engineering, Web Technologies, Internet of Things (IoT). She is actively involved in teaching, research, and academic activities in the field of computer science and emerging technologies.