# PHARMA DEAD STOCK MANAGEMENT

**A MINI PROJECT REPORT**

***Submitted by***


**DEVIKA C (221801009)**

**KEERTHIKA P (221801027)**

**LAVANYA S (221801028)**


***in partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY IN ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**


**RAJALAKSHMI ENGINEERING COLLEGE**
**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**ANNA UNIVERSITY, CHENNAI**


**MAY 2025**

# ANNA UNIVERSITY, CHENNAI

## BONAFIDE CERTIFICATE

Certified that this Report titled "**PHARMA DEAD STOCK MANAGEMENT SYSTEM**" is the bonafide work **DEVIKA C (221801009) , KEERTHIKA P (221801027), LAVANYA S (221801028)** who carried out the work under my supervision .

| SIGNATURE | SIGNATURE |
|---|---|
| **Dr. J. M. GNANASEKAR, M.E., Ph.D.,**<br>Professor and Head,<br>Department of Artificial Intelligence and Data Science,<br>Rajalakshmi Engineering College,<br>Chennai – 602 105. | **Dr. S.SURESH KUMAR, M.E., Ph.D.,**<br>Professor,<br>Department of Artificial Intelligence and Data Science,<br>Rajalakshmi Engineering College,<br>Chennai – 602 105. |

Submitted for the project viva-voce examination held on______________________.

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

# ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E**., our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. J. M. GNANASEKAR., M.E., Ph.D.,** Head of the Department, Professor and Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. We are glad to express our sincere thanks to our supervisor **Dr. S.SURESH KUMAR., M.E., Ph.D.,** Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

Finally we express our thanks for all teaching, non-teaching, faculty and our parents for helping us with the necessary guidance during the time of our project.

# ABSTRACT

Effective pharmaceutical inventory management is crucial for minimizing waste, ensuring medicine availability, and maintaining regulatory compliance. Traditional inventory systems often lack predictive analytics, real-time tracking, and automated alerts, leading to expired stock accumulation, financial losses, and inefficient stock movement. This paper proposes an AI-driven dead stock management system that integrates logistic regression-based expiry prediction, barcode-enabled stock tracking, and e-commerce synchronization to enhance warehouse efficiency and reduce wastage. The system classifies medicines into high, medium, and low expiry risk categories based on multiple parameters, such as expiry dates, stock movement trends, and demand patterns. By leveraging machine learning, the system generates automated alerts and maintains an expiry calendar, allowing warehouse managers to take timely corrective actions and prevent unnecessary losses. Additionally, Shopify integration facilitates the clearance of near-expiry medicines through discounted online sales, maximizing revenue recovery and improving inventory turnover. The proposed system not only optimizes pharmaceutical inventory management through predictive analytics and real-time tracking but also ensures compliance with industry regulations by preventing expired medicines from entering circulation. By combining AI-driven analytics, automated notifications, and seamless e-commerce integration, this approach revolutionizes pharmaceutical supply chain management, making it more data-driven, cost-effective, and operationally efficient while significantly reducing medicine wastage.

## TABLE OF CONTENTS


| CHAPTER NO. | TITLE | PAGE NO. |
|---|---|---|
| | **ABSTRACT** | |
| | **LIST OF FIGURES** | **VI** |
| 1. | **INTRODUCTION** | |
| | 1.1 GENERAL | 1 |
| | 1.2 NEED FOR THE STUDY | 1 |
| | 1.3 OBJECTIVE OF THE STUDY | 2 |
| | 1.4 OVERVIEW | 3 |
| **2.** | **REVIEW OF LITERATURE** | |
| | 2.1 INTRODUCTION | 4 |
| | 2.2 LITERATURE REVIEW | 5 |
| **3.** | **SYSTEM OVERVIEW** | |
| | 3.1 EXISTING SYSTEM | 6 |
| | 3.2 PROPOSED SYSTEM | 8 |
| | 3.3 FEASIBILITY STUDY | 11 |
| **4.** | **SYSTEM REQUIREMENTS** | |
| | 4.1 HARDWARE REQUIREMENTS | 14 |
| | 4.2 SOFTWARE REQUIREMENTS | 15 |
| **5.** | **SYSTEM DESIGN** | |
| | 5.1 SYSTEM ARCHITECTURE | 18 |

| | | |
|---|---|---|
| | 5.2 MODULE DESCRIPTION | 19 |
| | 5.2.1 ORDER MANAGEMENT MODULE | 19 |
| | 5.2.2 WAREHOUSE MANGEMENT MODULE | 20 |
| | 5.2.3 BARCODE SCANNING MODULE | 23 |
| | 5.2.4 SHOPIFY INTEGRATION MODULE | 25 |
| | 5.2.5 REPORTING SYSTEM | 26 |
| **6.** | **RESULT AND DISCUSSION** | 28 |
| **7.** | **CONCLUSION** | |
| | 7.1 CONCLUSION | 30 |
| **8.** | **REFERENCES** | **42** |

# CHAPTER I

# INTRODUCTION

## 1.1 GENERAL

Inventory mismanagement—especially of non-moving or obsolete items—is a major concern across industries like pharmaceuticals, where it leads to resource wastage, compliance risks, and operational inefficiencies. Traditional systems often lack features like predictive analytics, real-time tracking, and automated notifications, making them ineffective in handling dead stock. To address this, our project proposes an **AI-driven Dead Stock Management System** that uses machine learning (logistic regression) to predict expiry, classify inventory by risk levels, and provide intelligent alerts. The system also supports **barcode-enabled stock tracking**, **centralized item classification**, and **automated alerts**, enabling warehouse managers to act promptly and reduce losses.

In addition, the system integrates with **e-commerce platforms like Shopify** to facilitate the discounted sale of near-expiry medicines, improving inventory turnover and maximizing revenue recovery. With features like an expiry calendar, supplier management, and detailed reporting, the solution promotes transparency and supports regulatory compliance. By combining AI, automation, and seamless e-commerce integration, the system enhances pharmaceutical supply chain efficiency, reduces wastage, and paves the way for a **data-driven and sustainable inventory management approach**.

## 1.2 NEED FOR THE STUDY

Inventory management is a critical component of supply chain operations, particularly in the pharmaceutical industry, where the consequences of holding expired or obsolete stock can be severe. Inadequate control over inventory often results in expired medicines, overstocking, understocking, and financial loss, while also violating health and safety regulations. Traditional inventory systems are typically manual, static, and reactive, lacking the necessary intelligence to forecast expiry trends or detect dead stock before it becomes a liability. These limitations emphasize the urgent need for a more intelligent and automated inventory management solution.

The study is driven by the necessity to implement a data-driven, predictive approach to inventory control. An AI-driven dead stock management system offers proactive features such as expiry date prediction, stock movement analysis, automated alerts, and barcode-enabled tracking, helping managers identify and act on non-moving inventory well in advance. Furthermore, e-commerce platform integration allows organizations to recover value from near-expiry items through timely clearance sales, thus reducing wastage and improving inventory turnover.

In the context of the pharmaceutical supply chain, where accuracy, efficiency, and compliance are non-negotiable, the adoption of an intelligent inventory system becomes even more crucial. This study aims to develop a solution that not only enhances visibility and decision-making but also ensures regulatory compliance, sustainability, and operational cost-efficiency. By leveraging machine learning algorithms and smart technologies, the proposed system transforms conventional inventory practices into a modern, automated, and optimized process that addresses both present and future inventory challenges.

## 1.3 OBJECTIVES OF THE STUDY

The primary goal of this study is to design and implement an AI-powered Dead Stock Management System that improves the efficiency, accuracy, and sustainability of pharmaceutical inventory handling. By leveraging automation, predictive analytics, and e-commerce integration, the system aims to address existing challenges related to expired stock, wastage, and poor inventory visibility.

Key Objectives:

1. To develop a logistic regression-based model for predicting the expiry risk of inventory items based on historical trends and usage patterns.

2. To implement real-time stock tracking using barcode integration for accurate and efficient item monitoring.
3. To automate alert notifications and maintain an expiry calendar for proactive decision-making and timely stock clearance.
4. To classify inventory into high, medium, and low expiry risk categories for better prioritization and management.

## 1.4 OVERVIEW OF THE PROJECT

Inventory mismanagement, especially in sectors like pharmaceuticals, can lead to serious consequences such as expired stock, increased wastage, regulatory non-compliance, and financial losses. Traditional inventory systems are often reactive and lack features such as predictive analytics, intelligent alerts, and e-commerce integration. To address these limitations, this project presents an **AI-driven Dead Stock Management System** that leverages machine learning, real-time tracking, and automated workflows to effectively manage and reduce obsolete or non-moving inventory.

The system uses a **logistic regression model** to predict the expiry risk of pharmaceutical products based on historical stock movement, shelf life, and demand trends. Each item is categorized into high, medium, or low-risk groups, allowing warehouse managers to prioritize actions. The system maintains an **expiry calendar**, which serves as a timeline for identifying stock approaching its expiration date, and sends **automated notifications** to prompt timely interventions such as internal redistribution, clearance sales, or disposal.

A key feature of this system is its **barcode-enabled tracking**, which ensures real-time inventory updates, reduces manual errors, and streamlines stock monitoring. In addition, the project integrates with **e-commerce platforms like Shopify**, allowing near-expiry items to be listed for discounted sales online. This not only helps reduce waste but also enables organizations to recover some of the cost associated with products that would otherwise go unused.

# CHAPTER II

# REVIEW OF LITERATURE

## 2.1 INTRODUCTION

The effective management of inventory, particularly in the pharmaceutical sector, has long been a subject of research due to its direct impact on organizational efficiency, regulatory compliance, and public health. As the demand for better stock control and minimization of wastage grows, researchers have explored various technological and analytical methods to improve inventory systems. The literature reveals that traditional inventory practices often struggle with issues like manual tracking, lack of real-time visibility, and inability to forecast demand and expiry trends accurately. These gaps have led to significant interest in applying artificial intelligence (AI), machine learning, and automation technologies to improve inventory performance.

Several studies highlight the need for predictive models that can assess expiry risk and track stock movement trends to reduce dead stock. Others emphasize the importance of barcode systems and RFID for real-time inventory visibility, and the role of e-commerce integration in clearing obsolete stock efficiently. The literature also supports the use of data-driven approaches and cloud-based solutions for centralized inventory control, enhanced decision-making, and improved transparency across supply chains. This review summarizes existing research and technological advancements that form the foundation for the proposed AI-based dead stock management system, identifying both the challenges in current practices and the potential benefits of an intelligent, automated solution.

## 2.2 FRAMEWORK OF LCA (LITERATURE CRITICAL ANALYSIS)

The **Literature Critical Analysis (LCA)** framework serves as a structured approach to evaluate, compare, and synthesize previous research studies that are relevant to the development of a Dead Stock Management System. This framework is designed to critically analyze existing methodologies, technologies, findings, and limitations in order to identify knowledge gaps and justify the proposed AI-driven approach.The framework encompasses five key dimensions:

**1. Technological Perspective**

This dimension investigates the technologies adopted in past inventory management systems, such as **barcode scanning, RFID tracking, AI/ML algorithms**, and their integration with **cloud infrastructure and e-commerce platforms**. It assesses how these technologies have influenced inventory visibility, automation levels, and predictive accuracy—and where they have fallen short.

**2. Functional Gaps in Existing Systems**

This component identifies the shortcomings in traditional inventory systems, especially in handling **non-moving or dead stock**. Issues such as the **lack of real-time data**, **ineffective expiry tracking**, and **limited automation** are explored. The analysis pinpoints how these limitations hinder timely decision-making and contribute to inventory wastage.

**3. AI and Predictive Analytics Usage**

Here, the use of **AI techniques**—such as **logistic regression, decision trees, and neural networks**—in prior studies is evaluated. The framework examines how predictive analytics has been leveraged to forecast expiry dates, classify stock by risk, and enhance inventory planning. It also considers the **accuracy, scalability, and real-world applicability** of such models.

**4. Integration and Automation Capabilities**

This dimension evaluates the level of **system integration** with platforms like **ERP software or e-commerce tools (e.g., Shopify)**, and the extent of **automation** in functions like alerts, expiry notifications, and stock clearance. It focuses on how integration and automation contribute to **proactive inventory control** and **operational efficiency**.

# CHAPTER III

# SYSTEM OVERVIEW

## 3.1 EXISTING SYSTEM

In the current pharmaceutical industry, inventory management primarily relies on manual stock tracking or traditional Enterprise Resource Planning (ERP) systems that are often outdated, inefficient, and reactive rather than predictive. Most pharmacy warehouses and distributors maintain inventory records manually using spreadsheets, ledger books, or basic database management systems. These systems focus primarily on tracking available stock quantities and reordering thresholds, but they lack critical features such as expiry prediction, dead stock identification, and real-time analytics, which are essential for modern pharmaceutical supply chain management.

The traditional methods heavily depend on human intervention for tasks such as stock counting, expiry monitoring, and restocking. This manual handling often leads to inaccuracies, errors, and inconsistencies in stock data. Warehouse managers typically rely on physical audits at fixed intervals to check for expired or soon-to-expire medicines. These audits are time-consuming and prone to human error, resulting in the accumulation of dead stock that goes unnoticed until the stock is either expired or rendered unsellable. Consequently, pharmacies and distributors incur significant financial losses due to wastage, non-compliance with regulatory standards, and unnecessary overstocking.

Another critical limitation of the existing system is the lack of proactive notification mechanisms. There are no automated alerts generated for near-expiry products, leading to a delayed response in selling or disposing of these medicines. This delay increases the amount of waste and contributes to inefficient stock rotation practices. Without real-time expiry data or movement trends, warehouse managers struggle to prioritize inventory clearance, which further aggravates the dead stock problem.

Furthermore, most traditional systems do not integrate with online platforms or sales channels. This lack of integration limits the opportunity to quickly move excess or near-expiry stock through discounted sales or targeted promotions. Manual listing and sales processes introduce delays, reducing the chances of revenue recovery before the products become obsolete.

In addition to these inefficiencies, the existing systems generally lack detailed reporting and analytics capabilities. Reports, if generated, are basic and limited to stock availability rather than providing insights into expiry trends, high-risk products, or warehouse performance metrics. Managers often have to manually analyze scattered data to understand which products are stagnant and where action is needed, consuming valuable time and resources.

Moreover, regulatory compliance is another area where traditional inventory systems fall short. Regulatory bodies require stringent monitoring of expiry dates and stock movement records to ensure that expired or unsafe medicines do not reach consumers. Without robust tracking and expiry monitoring mechanisms, pharmacies and warehouses risk non-compliance, which can result in severe penalties, reputational damage, or loss of license.

In terms of technological adaptation, barcode usage is minimal or nonexistent in many traditional systems. Most warehouses still rely on manual data entry of batch numbers and medicine details, which is tedious and error-prone. The absence of barcode-based tracking hampers operational efficiency and traceability, making it harder to quickly identify products, track their lifecycle, and ensure first-expiry-first-out (FEFO) inventory movement.

To summarize, the existing system for pharmaceutical stock management is characterized by high manual dependency, poor expiry monitoring, minimal automation, lack of integration with e-commerce, weak analytics, and significant risks of regulatory non-compliance. These inefficiencies collectively contribute to increased operational costs, higher wastage rates, poor stock visibility, and lost revenue opportunities. With growing demands for efficiency, compliance, and sustainability in the healthcare sector, it is evident that traditional dead stock management practices are insufficient to meet modern needs, and a more intelligent, automated, and predictive solution is required.

## 3.2 PROPOSED SYSTEM

The proposed system presents an integrated, intelligent platform for managing dead stock in pharmaceutical inventories. The system is composed of several interrelated modules — Order Management, Warehouse Management, Reporting, and Alert/Calendar features — that work cohesively to optimize inventory handling, minimize wastage, and maximize operational efficiency. By leveraging real-time tracking, predictive analytics, and automated notifications, the system ensures proactive management of pharmaceutical products, particularly those nearing expiry.

### 1. Order Management

Order Management is the foundation of efficient inventory control. In the proposed system, a centralized Order Management module is developed to handle the complete life cycle of pharmaceutical orders. The key functionalities include:

- **Order Creation:** New purchase orders can be created by entering essential details such as customer information, product name, batch number, quantity, and expected delivery dates.
- **Order Tracking:** All orders are assigned a unique Order ID and their status is updated in real-time — from "Pending" to "Delivered" or "Cancelled."
- **Editing and Updating Orders:** Users can modify order details dynamically in case of stock adjustments, customer updates, or order fulfillment changes.
- **Deletion and Archiving:** Unfulfilled or obsolete orders can be deleted or archived without impacting active inventory records.

The system uses **MongoDB** to persistently store order information and ensures that all updates reflect immediately across all related modules. Barcode scanning functionality is integrated during the order placement to minimize human errors and quickly validate product batches.

**2. Warehouse Management**

Warehouse Management plays a critical role in ensuring that medicines are stored correctly and moved efficiently to prevent stock expiration and wastage. The warehouse module is designed with the following features:

- **Stock Entry and Exit:** Products are added into the warehouse system through barcode scanning, ensuring that the batch numbers, quantities, and expiry dates are accurately recorded.
- **Batch Tracking:** Every batch of medicine is tracked individually based on its expiry risk. Products are organized using **First Expiry First Out (FEFO)** principles.
- **Location Mapping:** Each product batch is assigned a storage location (e.g., Rack A, Shelf 3), enabling quick retrieval and reducing search times.
- **Expiry Risk Classification:** Based on movement history and expiry dates, items are categorized into **High Risk**, **Medium Risk**, and **Low Risk** for optimized handling.
- **Warehouse Transfer Management:** Products approaching expiry can be automatically suggested for transfers to different pharmacies or branches where demand is higher.

**3. Reporting Module**

The Reporting module provides detailed, actionable insights into inventory health, operational efficiency, and expiry risk trends.
Key report types generated by the system include:

- **Dead Stock Analysis Report:** Identifies products that have not moved for a specified period and categorizes them by expiry proximity.
- **Expiry Risk Reports:** Lists products grouped by High, Medium, and Low expiry risk, enabling warehouse managers to prioritize actions.

- **Stock Movement Reports:** Tracks inflow and outflow of products across the warehouse over customizable time periods.
- **Revenue Recovery Reports:** Highlights financial gains achieved by clearing near-expiry products through discounted sales or redistribution.

Reports are presented through dynamic **visualizations** — including **bar charts**, **pie charts**, and **line graphs** — for easy interpretation. Managers can export reports in multiple formats (CSV, PDF) for audits, performance reviews, and operational planning.

**4. Alerts and Calendar Integration**

One of the innovative aspects of the proposed system is the integration of an intelligent **Alert and Calendar system** that proactively notifies managers before problems occur.

**4.1. Automated Alerts**

- **Expiry Alerts:** Products nearing expiry (e.g., within 90, 60, 30 days) trigger automated alerts to warehouse staff.

Alerts are visible on the dashboard and can optionally be sent via email or SMS to responsible personnel. This ensures that no critical product movement or expiry event is overlooked.

**4.2. Inventory Expiry Calendar**

An interactive expiry calendar is maintained, where each day lists products approaching expiry within the next 3 months.
Features of the calendar include:

- **Color Coding:** Red for High-risk products, Yellow for Medium-risk, Green for Low-risk.
- **Clickable Events:** Clicking a calendar entry shows product details, quantity, and suggested actions.

The calendar provides a **predictive view** of upcoming inventory risks, allowing the pharmacy to plan clearance sales, supplier returns, or stock transfers in advance.

## 3.3 FEASIBILITY STUDY

The AI-driven Dead Stock Management System proposed for pharmaceutical inventory management aims to integrate predictive analytics, real-time stock tracking, and e-commerce synchronization. The feasibility study below assesses the project's technical, operational, economic, and legal and ethical feasibility.

### 1. Technical Feasibility

**Technology Stack:**

- **MERN Stack**: The system will utilize **MongoDB** for flexible data storage, **Express.js** for backend API management, **React** for a dynamic frontend UI, and **Node.js** for scalable server-side operations.
- **Logistic Regression**: This machine learning model will predict expiry risk based on factors like expiry date, stock movement, and demand patterns. Python will handle model training and inference, integrated into the backend.
- **Barcode Tracking**: Integration with barcode scanners will allow real-time tracking of stock levels, ensuring timely identification of nearing-expiry medicines.
- **Shopify Integration**: The system will synchronize with **Shopify** to sell near-expiry stock at discounted prices, maximizing revenue recovery and improving inventory turnover.

**System Integration:**

- The backend (Node.js/Express.js) will interact with the **logistic regression** model and external services (like Shopify and Twilio for SMS/email alerts).
- **Real-Time Alerts**: Automated notifications will alert warehouse managers about high-risk medicines or overstock situations, facilitating timely actions.

**2. Operational Feasibility**

**System Operations:**

- The Web UI will be designed for ease of use, allowing warehouse managers to efficiently track and manage stock levels, expiry predictions, and alerts.
- **Training**: Warehouse staff will receive training on interpreting alerts, using the expiry calendar, and making decisions based on the system's recommendations.
- **Maintenance**: Regular updates to the logistic regression model will be necessary for continuous accuracy. Maintenance schedules can minimize disruptions.

**Scalability:**

- The MERN stack ensures that the system can scale for larger inventories, multiple warehouses, or different regions.
- Future improvements could include adding more predictive models or advanced analytics.

**3. Economic Feasibility**

**Cost Considerations:**

- **Development Costs**: The primary expenses involve development of the MERN stack application, Python integration for logistic regression, and the API integrations for barcode scanning and Shopify.
- **Operational Costs**: Hosting services (e.g., AWS, Heroku), API services (Twilio, SendGrid), and cloud infrastructure for model deployment will incur ongoing costs.
- **Revenue Potential**: By selling near-expiry medicines via **Shopify**, the system can recoup revenue that would otherwise be lost due to dead stock. The reduction in wastage also translates to higher profitability.

**Return on Investment (ROI):**

- The system reduces wastage, improves stock turnover, and optimizes stock procurement, leading to significant cost savings and better financial performance. The investment in technology will lead to a positive ROI through improved efficiency and reduced losses.

**4. Legal and Ethical Feasibility**

**Regulatory Compliance:**

- **Pharmaceutical Regulations**: The system complies with **FDA**, **GMP**, and **EU pharmaceutical standards** by ensuring that expired medicines are flagged and properly disposed of or redirected before reaching consumers.
- **Data Privacy**: The system will adhere to **GDPR** for European users and **HIPAA** in case of patient-related data, ensuring proper handling of sensitive information.

**Ethical Considerations:**

- **Transparency**: The expiry prediction model will be explainable, enabling warehouse managers to understand and trust the AI-generated insights.
- **Social Responsibility**: By ensuring expired medicines are not circulated, the system contributes to public health safety.

# CHAPTER – IV

# SYSTEM REQUIREMENTS

## 4.1 HARDWARE REQUIREMENT

In the development and deployment of a **Pharma Dead Stock Management System** (including barcode scanning through image uploads), appropriate hardware support is essential to ensure optimal system performance, reliability, and scalability. This section outlines the minimum and recommended hardware specifications necessary for both the **server** and **client** machines involved in the project.

### 1. Server/Hosting Infrastructure

- **CPU**: A multi-core processor, such as an **Intel Xeon** or **AMD EPYC** with at least 8 cores, is recommended to handle the computational load of machine learning algorithms and real-time data processing.
- **RAM**: A minimum of **32 GB DDR4 RAM** is recommended for smooth handling of large datasets and to support efficient caching mechanisms. More RAM (64 GB or higher) may be required for high-traffic environments or larger datasets.
- **Networking**: High-speed Ethernet connection (minimum 1 Gbps) to ensure fast data transfer between servers and users, ensuring that real-time recommendations are delivered without delay.

### 2. GPU (Graphical Processing Unit)

- For machine learning tasks, especially training recommendation models like collaborative filtering and content-based filtering algorithms, a **dedicated GPU** is highly recommended. A **NVIDIA Tesla** or **A100 GPU** with at least **16 GB of VRAM** is ideal for accelerating complex computations like matrix factorization and neural network training.

### 3. Cooling and Power Supply

- **Cooling Systems**: Servers should be equipped with adequate cooling systems (air or liquid cooling) to maintain optimal performance and avoid overheating during heavy computational loads.
- **Power Supply**: A **redundant power supply** (UPS) to ensure the system remains operational during power outages and avoid data loss or corruption.

## 4.2 SOFTWARE REQUIREMENTS

### 1. Server-Side Software

To build the backend part of the Pharma Dead Stock Management System, the following server-side software components are required:

- **Node.js**:
  Node.js serves as the runtime environment for executing server-side JavaScript code. It is essential for building a lightweight, scalable, and high-performing backend service.
- **Express.js**:
  Express.js is a web application framework for Node.js used to create the API endpoints. It simplifies server creation and routing functionalities.
- **MongoDB**:
  MongoDB is a NoSQL database used to store and manage all the critical data such as warehouse stock information, barcode details, and expiry logs.
- **CORS**
  CORS (Cross-Origin Resource Sharing) middleware ensures that the frontend running on a different domain or port can securely interact with the backend APIs.

- **Body-Parser**:
  Integrated into Express.js, body-parser parses incoming JSON requests so that the server can easily process the submitted data.
- **Nodemon (for development)**:
  Nodemon is a development utility that automatically restarts the server when file changes are detected, helping in rapid backend development.
- **Postman**:
  Postman is used during development for API testing, ensuring that all endpoints behave as expected.
- **MongoDB Compass**:
  MongoDB Compass is a GUI tool to view and interact with the database visually without using command-line queries.
- **Git**:
  Git is essential for source code version control, allowing easy collaboration and project management.

**2. Client-Side Software**

The frontend part of the system provides an interactive dashboard for users to manage stock, track expiry risk, and decode barcodes.

- **Vite**:
  Vite is a modern frontend build tool that significantly speeds up development and bundling processes compared to older tools like Webpack.
- **Axios**:
  Axios is a promise-based HTTP client for making API requests from the frontend to the backend server.
- **React Router DOM**:
  This library is used to implement navigation and routing between different pages within the application, such as Warehouse Dashboard, Reports Page, and Barcode Scanner.

- **@zxing/browser**:
  This library is used for decoding barcode images directly in the browser, enabling the image upload barcode scanning functionality.
- **VS Code (Visual Studio Code)**:
  A popular and lightweight code editor highly recommended for both backend and frontend development.
- **Google Chrome Developer Tools**:
  Chrome’s built-in developer tools help in inspecting elements, debugging JavaScript, and monitoring network activity during frontend development.

**3. Browser Compatibility**

To ensure the best performance and user experience, the application should support all modern browsers:

- Google Chrome (latest version)
- Safari (latest version)

The web application should also be responsive and functional on tablets and smartphones for users who may need mobile access.

**4. Dependencies and Libraries**

The project requires the installation of several important libraries for smooth functioning:

- Mongoose for structured database interactions.
- Axios for communication between frontend and backend.
- React Router DOM for frontend navigation.
- @zxing/browser for barcode image decoding functionality.

The software requirements detailed above provide a strong foundation for developing the Pharma Dead Stock Management System.
Using modern backend, frontend, and database technologies ensures that the platform will be efficient, secure, scalable, and easy to maintain.

# CHAPTER V

# SYSTEM DESIGN

## 5.1 SYSTEM ARCHITECTURE

The proposed Dead Stock Management System follows a modular, interconnected architecture centered around a centralized database. The system begins with the **Order Management** and **Barcode Scanner** modules, where orders and batch codes are created, scanned, and stored into the database. The **Warehouse Management** module retrieves and updates inventory details based on stock movement and expiry information. **Logistic Regression** algorithms are integrated to predict expiry risks for each product, and generate **Expiry Prediction Alerts** proactively. All inventory and expiry data is visualized through the **Report UI**, which connects to both the **Expiry Date Calendar** and the user-facing **Web UI** for real-time monitoring. Simultaneously, products nearing expiry are synchronized with **Shopify**, enabling clearance sales and reducing wastage. Data flows between the modules ensure seamless inventory tracking, expiry prediction, and external e-commerce integration, thereby making pharmaceutical operations highly efficient, predictive, and data-driven.

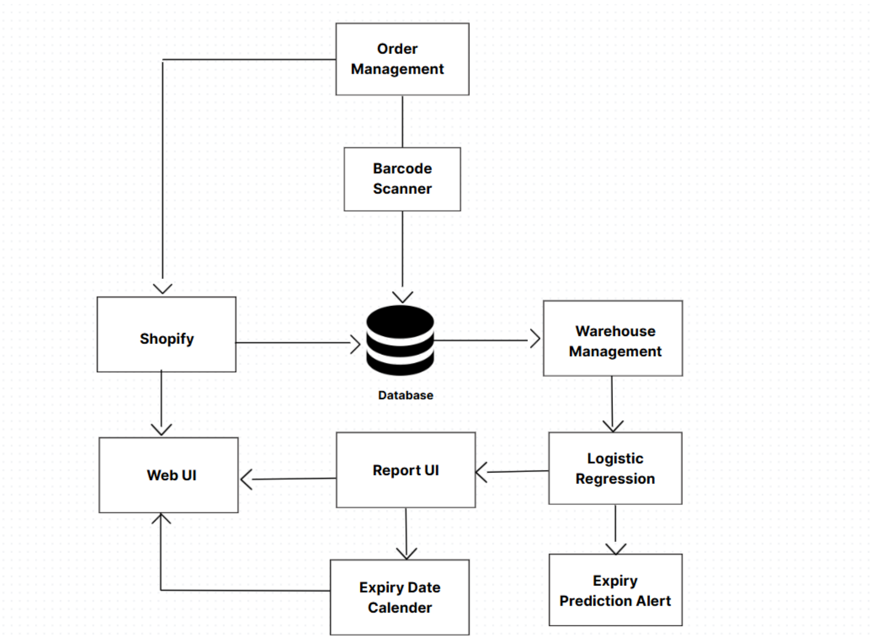



FIG 5.1 SYSTEM ARCHITECTURE

## 5.2 MODULE DESCRIPTION

### 5.2.1 ORDER MANAGEMENT MODULE

The Order Management module is a critical component of the proposed Dead Stock Management System, designed to systematically manage pharmaceutical orders, ensuring operational efficiency and real-time tracking. In the pharmaceutical industry, orders play a vital role in maintaining adequate inventory levels, fulfilling customer demands, and ensuring timely movement of stock. However, traditional manual order handling often leads to errors, delays, and the accumulation of dead stock. Therefore, an automated, intelligent Order Management system is essential to address these challenges effectively.

This module facilitates the creation, tracking, updating, and management of pharmaceutical orders through an intuitive web-based interface. Users, such as warehouse managers or pharmacists, can create new orders by inputting essential details such as order ID, customer information, medicine name, batch number, quantity, order date, delivery status, and total amount. Each order entry is stored systematically into the central database, ensuring that all transactions are recorded in real-time and accessible for further processing.

A key feature of the Order Management module is its integration with the barcode scanning functionality. Once a medicine or batch is scanned using the Barcode Scanner module, the order information is linked with the respective batch, ensuring authenticity and accuracy of the stock being ordered or dispatched. This real-time validation minimizes manual entry errors and speeds up the order fulfillment process.

The system also supports **editing and updating** existing orders. Users can modify critical information such as order quantity, status (e.g., Delivered, Pending, Cancelled), and update delivery details as needed. This flexibility allows the system to reflect real-world scenarios such as partial deliveries, back-orders, or cancellations due to stock unavailability. By offering edit and delete functionalities directly from the order listing page, the system ensures dynamic and up-to-date inventory flow management.

Additionally, the Order Management module provides **search and filter capabilities**. Users can easily search for specific orders by customer name, order ID, or batch number. Filters based on order status enable warehouse managers to quickly view pending, delivered, or cancelled orders, helping them prioritize tasks and ensure operational smoothness. These features significantly enhance usability, allowing large datasets to be navigated efficiently.

Another important feature is **status monitoring**. The status of each order is continuously tracked and updated to reflect its real-world lifecycle—from creation, packaging, dispatch, to delivery. Status tracking improves communication among different stakeholders and ensures that critical deadlines are not missed, especially when dealing with short-shelf-life medicines.

In summary, the Order Management module ensures that the pharmaceutical supply chain remains efficient, error-free, and responsive to market needs. By automating order creation, tracking, editing, and integration with barcode scanning and, this module addresses key pain points in pharmaceutical logistics. Its real-time updates, data-driven insights, and intelligent status monitoring collectively contribute to a more agile, reliable, and waste-reducing pharmaceutical warehouse management system.

### 5.2.2 WAREHOUSE MODULE DESCRIPTION

The proposed Warehouse Management System (WMS) introduces intelligence and automation into pharmaceutical warehouse operations. By integrating AI-driven predictive analytics, logistic regression models, and proactive alert mechanisms, the system transforms how inventory, expiry risks, and stock movements are managed.

#### 1. Predictive Expiry Risk Management and Stock Movement Optimization

Traditional inventory systems focus only on stock counts and basic expiry tracking, often missing opportunities to prevent dead stock and financial losses. The new WMS uses a **logistic regression model** to predict expiry risks by analyzing factors like expiry dates, last issued dates, stock movement rates, and supplier restock frequency. Medicines are automatically classified into **high-risk**, **medium-risk**, and **low-risk** categories.

An **Expiry Calendar** with color-coded zones (Green: safe, Yellow: warning, Red: critical) provides visual tracking of stock expiry status. Automated alerts via SMS and email notify managers about medicines nearing expiry (within 30 days), high-risk batches, or slow-moving overstock. This enables proactive actions such as redistribution, discounting, or disposal to minimize wastage.

Simultaneously, the system classifies stock as **fast-moving** or **slow-moving** based on historical sales data. Fast-moving items are prioritized for restocking, while slow-moving medicines trigger suggestions for redistribution to higher-demand locations. Dynamic procurement adjustments based on real-time trends help prevent overstocking and dead stock formation.

**2. Key Benefits and Impact**

By intelligently predicting expiry risks and optimizing stock movement, the system delivers major advantages:

- **Reduced Dead Stock:** Early expiry predictions and stock redistribution strategies drastically cut wastage.
- **Higher Inventory Turnover:** Prioritizing fast-moving stock boosts cash flow and warehouse space efficiency.
- **Proactive Supply Chain Management:** Data-driven insights replace reactive decisions, leading to smoother operations.
- **Better Financial Performance:** Minimizing losses from expired stock directly improves profitability for pharmaceutical warehouses.

**Predicting Medicine Expiry Risk Using Logistic Regression**

The logistic regression function is given by:

$$\boldsymbol{h_{\theta}(X) = \frac{1}{1 + e^{-z}}}$$

where

$$\boldsymbol{z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5}$$

**Explanation:**

$\boldsymbol{h_\theta(X)}$ the probability that the medicine batch will expire.

$\boldsymbol{z}$ is a weighted sum of input features.

$\boldsymbol{\theta}$ represents the regression coefficients (weights) that are learned from the data**.**

$\boldsymbol{x_1, x_2 \ldots . x_n}$ are the independent variables (features) affecting expiry risk.

**Risk Classification Using Thresholding**

Once we obtain the probability $\boldsymbol{h_\theta(X)}$, we classify the expiry risk into Low, Medium, or High Risk using predefined thresholds.

Threshold-Based Classification:

[ Expiry Risk = ⟦"$\{High\ Risk\}$, & $h\_\{\theta\}(X)\ \geq 0.7$ "

"$\{Medium\ Risk\}$, & $0.4\ \leq h\_\{\theta\}(X)\ <\ 0.7$ "

"$\{Low\ Risk\}$, & $h\_\{\theta\}(X)\ <\ 0.4$ "⟧ ]

**Interpretation:**

- If the predicted probability is **above 70%**, the medicine batch is **high risk**.
- If the probability is **between 40% and 70%**, it is **medium risk**.
- If the probability is **below 40%**, it is **low risk**.

**Cost Function (Categorical Cross-Entropy Loss)**

To train the logistic regression model, we minimize the **Binary Cross-Entropy Loss.**

The function **penalizes incorrect predictions**, making the model adjust the parameters to improve accuracy.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log\left(h_\theta(x_i)\right) + (1 - y_i)\log\left(1 - h_\theta(x_i)\right)\right]$$

**Explanation:**

- $y_i$ is the actual label (1 if the batch expired, 0 if not).

- $h_\theta(x_i)$ is the predicted probability.
- The function **penalizes incorrect predictions**, making the model adjust the parameters to improve accuracy.

**Parameter Optimization Using Gradient Descent**

To optimize $\boldsymbol{\theta}$ (model parameters), we use **Gradient Descent**, an iterative method that updates parameters to minimize the loss function.

**Gradient Descent Update Rule:**

$$\boldsymbol{\theta_j \coloneqq \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}}$$

**Optimization Process:**

1. Compute the current loss J($\boldsymbol{\theta}$)
2. Calculate the gradient (rate of change of loss with respect to each $\boldsymbol{\theta}$).
3. Update $\boldsymbol{\theta}$ by moving in the direction that **reduces the loss**.
4. Repeat until convergence (loss stops decreasing significantly).

### 5.2.3 BARCODE SCANNING MODULE

The Barcode Scanning Module plays a crucial role in enhancing the accuracy, speed, and efficiency of inventory operations in the proposed Dead Stock Management System. In traditional pharmaceutical warehouses, manual entry of product information often results in errors, delays, and inconsistencies, leading to poor inventory control and significant wastage. To address these challenges, the Barcode Scanning Module is introduced as a technological innovation that ensures real-time, error-free capture of batch-level product data.

This module leverages a laptop's webcam integrated with a barcode scanning library such as react-qr-barcode-scanner to decode barcodes from pharmaceutical product labels. When a barcode is scanned, the system instantly reads the embedded batch number or unique identifier from the product. This batch information is then transmitted to the backend server, where it is either matched with existing entries in the database or a new product record is created. This seamless process ensures that inventory records are updated accurately without the need for manual input.

The Barcode Scanning Module is tightly coupled with the warehouse and order management systems. When a new shipment arrives or a product is dispatched, scanning the barcode automatically updates the stock quantity, expiry dates, and relevant metadata associated with the batch. This automation reduces human error, accelerates processing times, and provides real-time visibility into inventory movements.

Moreover, the barcode scanning functionality supports not only batch number retrieval but also product detail enrichment. Upon scanning, users can be prompted to enter additional information if required, such as the medicine name, supplier details, received date, and expiry date. This additional layer ensures completeness of records and prepares the system for advanced analytics such as expiry prediction and stock lifecycle monitoring.

The Barcode Scanning Module also plays a significant role in dead stock management. When nearing expiry, scanned products are flagged automatically and can be pushed into clearance sales platforms such as Shopify. This early detection mechanism is crucial in preventing the accumulation of unsellable expired goods and maximizing potential revenue recovery.

Furthermore, the system maintains a comprehensive scan history, including timestamps of when each barcode was scanned and updated. This historical data not only supports auditing and compliance but also enables warehouse managers to analyze stock inflow and outflow patterns, identify bottlenecks, and optimize warehouse operations.

From a technical perspective, the Barcode Scanning Module is designed to be lightweight and highly responsive. It operates in real time, offering immediate visual feedback to users upon successful or failed scans. The UI is kept intuitive and user-friendly to ensure that even non-technical warehouse staff can easily operate the system with minimal training.

### 5.2.4 SHOPIFY INTEGRATION MODULE

The Shopify Integration Module is a critical component of the proposed Dead Stock Management System, designed to maximize the utility of near-expiry and dead stock by facilitating their quick clearance through an online sales platform. Pharmaceutical businesses often face challenges with surplus inventory that is approaching its expiration date. Traditionally, such inventory either results in financial loss or disposal costs. By integrating with Shopify, a leading e-commerce platform, the system offers an intelligent and automated pathway to market and sell these products before they lose commercial value.

The primary function of the Shopify Module is to synchronize the inventory database with an online storefront in real-time. When products in the warehouse are flagged as “near-expiry” by the expiry prediction engine or manually by warehouse managers, they are automatically pushed to the Shopify store. This allows these products to be listed with appropriate discounts and promotions aimed at accelerating sales while maintaining compliance with pharmaceutical sales regulations.

The Shopify Integration Module not only enables inventory clearance but also enhances the visibility of dead stock to a broader customer base, including smaller clinics, pharmacies, and healthcare centers that can utilize discounted medicines promptly. By leveraging Shopify's robust payment, order management, and customer communication features, pharmaceutical companies can efficiently manage transactions, issue invoices, and maintain a digital record of stock movements, all within a secure and scalable infrastructure.

Security and compliance are prioritized in this module. All product listings undergo validation to ensure that only non-expired, legally sellable products are listed. Additionally, once a product passes its expiry date, it is automatically delisted from the Shopify store, ensuring that expired medicines are not accidentally sold. This intelligent control maintains ethical standards, protects patient safety, and adheres to regulatory guidelines.

From a technical architecture standpoint, the Shopify Module communicates with the Shopify API via secure RESTful calls. It utilizes authentication tokens to safeguard data transmission and ensure authorized access. Inventory data, sales records, and product metadata are synchronized using webhooks and background tasks to maintain consistency between the warehouse database and the online storefront. This bi-directional synchronization ensures that all updates—whether initiated from the warehouse side or from online sales—are reflected system-wide without manual intervention.

. Warehouse managers can assess which categories of medicines are moving faster, identify buyer patterns, and adjust procurement and stocking strategies accordingly. This data-driven insight aids in further minimizing future dead stock accumulation.

Moreover, the Shopify Module offers customizable promotional strategies such as bundle discounts, flash sales, and loyalty rewards for purchasing near-expiry items. These tactics not only expedite inventory clearance but also strengthen brand reputation by promoting responsible stock management practices.

In terms of user experience, warehouse staff and administrators have access to a dedicated dashboard within the system where they can monitor Shopify listings, manage orders, track revenue from clearance sales, and generate compliance reports. This seamless interaction between internal inventory systems and external customer interfaces ensures operational transparency and efficiency.

### 5.2.5 REPORTING SYSTEM

The **Reporting System** integrated with the Web UI is designed to offer comprehensive analytics and real-time insights, empowering pharmaceutical warehouse managers, pharmacists, and business administrators to make data-driven decisions. The system serves as the backbone for tracking inventory performance, financial impacts, and optimizing stock management processes.

**1. Detailed Analytics and Performance Tracking**

The Report UI provides a user-friendly interface to view **stock movement**, **annual profitability**, and **medicine wastage**. It allows businesses to track and evaluate the overall performance of their inventory over time. Users can monitor:

- **Stock Movement Trends:** Identify fast-moving and slow-moving medicines.
- **Profitability Analysis:** Assess the financial impact of expired stock and determine which medicines contribute the most to profitability.
- **Medicine Wastage:** Track the wastage rates of expired medicines, enabling businesses to minimize losses.

**2. Real-Time Reporting on Key Metrics**

The reporting system enables businesses to generate **real-time reports** based on key metrics such as:

- **High-Risk Medicines:** Monitor medicines approaching expiration or identified as high-risk due to slow movement.
- **Stock Levels:** Real-time data on current inventory levels to avoid stockouts or overstocking.
- **Order Histories:** Track order patterns and identify trends for better future purchasing decisions.

This allows businesses to identify patterns, make accurate forecasts, and stay ahead in managing stock efficiently.

**3. Data-Driven Decision Making**

The system's core purpose is to empower stakeholders with **data-driven insights** that help improve decision-making processes. By analyzing historical data, warehouse managers can optimize:

- **Purchasing Decisions:** Adjust procurement strategies to prevent overstocking or understocking.
- **Redistribution Strategies:** Redirect stock from slow-moving to high-demand locations.

# CHAPTER VI

# RESULT AND DISCUSSION

## 6.1 RESULTS

The analysis underscores the persistent challenge organizations face in managing non-moving or obsolete inventory, which results in resource wastage, inefficiencies, and increased administrative overhead. From the literature review and real-world system evaluations, several key insights were derived:

### 1. Critical Need for Centralized Inventory Control

Many organizations struggle with disparate systems and inconsistent inventory tracking, which hinder transparency. A centralized inventory database is vital to provide a unified view of stock, allowing for improved traceability and management of dead stock across departments and locations.

### 2. Importance of Item Classification and Historical Tracking

Effective inventory control requires the ability to classify items based on usage frequency, expiry status, and movement history. Historical tracking helps in identifying patterns of non-moving items and supports proactive decision-making regarding stock clearance or redistribution.

### 3. Barcode Integration Enhances Accuracy and Speed

Technological tools like barcode scanning significantly improve data entry accuracy and speed up stock audits. Most modern systems are expected to support barcode-based input to streamline daily operations and reduce human error.

### 4. Digital Reporting Improves Procurement and Decision-Making

Access to detailed reports on stock movement, expiry trends, and inventory classification improves strategic planning. Systems equipped with such analytics allow stakeholders to make data-driven decisions that reduce waste and optimize resource use.

## 6.2 DISCUSSION

The findings from the Literature Critical Analysis (LCA) and system evaluation reveal that while technological advancements have transformed many aspects of inventory management, **dead stock remains an unresolved challenge** in several industries, particularly retail, manufacturing, and healthcare. This discussion reflects on the implications of these findings, the potential of AI integration, and the future direction of dead stock management systems.

### 1. Persistent Inefficiencies Despite Technological Adoption

Although inventory systems have adopted tools like RFID, barcoding, and basic analytics, they often fall short in **real-time decision-making** and **non-moving stock identification**. Many organizations use legacy systems or spreadsheets, which are reactive in nature and lack the intelligence to **predict, classify, or optimize** inventory usage effectively.

This highlights the need to move beyond digitization toward **intelligent automation**, where systems not only track stock but **make predictions and recommendations** regarding obsolete inventory.

### 2. Missed Opportunities with Predictive Analytics

The literature indicates that while predictive models (e.g., decision trees, logistic regression, neural networks) have been explored, **practical adoption is low**. This is partly due to poor data quality, lack of integration, and limited understanding of AI among end-users.

However, integrating AI in dead stock management has the potential to **forecast stock expiry**, **identify low-turnover items**, and even **suggest clearance strategies**, making inventory systems more proactive and intelligent.

# CHAPTER – VII

# CONCLUSION

## 7.1 CONCLUSION

The Dead Stock Management system for pharmacies successfully addresses the critical challenges of medicine wastage, inventory inefficiencies, and regulatory non-compliance. By integrating modules such as order management, warehouse tracking, barcode scanning, expiry prediction, and Shopify-based e-commerce, the system provides a comprehensive and automated solution for pharmaceutical inventory control. The use of machine learning for expiry prediction, along with real-time alerts and calendar views, empowers warehouse managers to make timely decisions that prevent loss and ensure medicine availability. Additionally, the ability to clear near-expiry stock through online sales not only reduces dead inventory but also boosts revenue recovery. The user-friendly interface, automated operations, and data-driven insights delivered by this system make it a robust and scalable solution for modern pharmaceutical supply chains. Overall, the project demonstrates how intelligent technology can revolutionize inventory management, reduce wastage, and ensure safer, more efficient distribution of medicines.

## SAMPLE CODE

### HomePage.jsx

```
import React from 'react';

import { Link } from 'react-router-dom';
import './HomePage.css';
import warehouse from "../assets/warehouse-inventory-check-concept-illustration_114360-30596.jpg";
const HomePage = () => {
 return (
  <div className="homepage-wrapper">
   {/* Navigation Bar */}
   <nav className="navbar">
    <div className="nav-logo">PharmaStock</div>
    <div className="nav-links">
     <Link to="/dashboard">Dashboard</Link>
     <Link to="/reports">Reports</Link>
     <Link to="/shop">Shop</Link>
     <Link to="/barcode">Scan</Link>
     <Link to="/ordermanagement">Orders</Link>
    </div>
   </nav>


   {/* Hero Section */}
   <header className="homepage-hero fade-in">
    <h1>Welcome to Pharma Stock Manager</h1>
    <p>Manage your stock, predict expiry risk, and streamline pharmacy operations.</p>
    <div className="hero-buttons">
     <Link to="/dashboard" className="hero-button">Explore Dashboard</Link>
```

```
      <Link to="/reports" className="hero-button">View Reports</Link>
    </div>
   </header>

   {/* About + Image Section */}
<section className="about-image-section fade-in">
  <div className="about-text">
   <h2>About Us</h2>
   <p>
    PharmaStock is dedicated to revolutionizing inventory management for the healthcare sector.
    We help pharmacies, warehouses, and distributors manage medicines, predict expiry risks using machine learning,
    and reduce dead stock. Our solution ensures timely distribution, zero wastage, and smarter stock control.
   </p>
  </div>
  <div className="about-image">
   <img src={warehouse} alt="Warehouse" className="warehouse-image" />
  </div>
</section>


   {/* Footer Section */}
   <footer className="homepage-footer fade-in delay2">
    <h3>Contact Us</h3>
    <p>Email: support@pharmastock.com</p>
    <p>Phone: +91 98765 43210</p>
    <p>Location: Chennai, India</p>
    <p>© 2025 PharmaStock Inc. All rights reserved.</p>
   </footer>
  </div>
```

```
  );
};

export default HomePage;
```

**WareHouse.jsx**

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';
import './WarehouseForm.css'; // Import the separate CSS file

const WarehouseForm = () => {
  const [formData, setFormData] = useState({
    warehouseId: '',
    location: '',
    medicineName: '',
    batchNo: '',
    stockQuantity: '',
    receivedDate: '',
    expiryDate: '',
    expiryRisk: 'Low',
    supplierName: '',
  });

  // Auto-generate warehouseId like WH001
  useEffect(() => {
    const generateId = async () => {
      try {
        const res = await axios.get('http://localhost:5000/api/warehouse');
        const count = res.data.length + 1;
        const newId = WH${String(count).padStart(3, '0')};
        setFormData((prev) => ({ ...prev, warehouseId: newId }));
```

```
    } catch (err) {
      console.error('Error generating warehouseId:', err);
    }
  };
  generateId();
}, []);

const handleChange = (e) => {
  const { name, value } = e.target;
  setFormData((prev) => ({
    ...prev,
    [name]: name === 'stockQuantity' ? Number(value) : value,
  }));
};

const handleSubmit = async (e) => {
  e.preventDefault();
  try {
    await axios.post('http://localhost:5000/api/warehouse', formData);
    alert('Warehouse data added successfully!');
    window.location.reload();
  } catch (err) {
    console.error(err);
    alert('Error adding warehouse data');
  }
};

return (
  <form onSubmit={handleSubmit} className="form-container">
    <h2 className="form-title">Add Warehouse Record</h2>
    <div className="form-grid">
```

```
<div className="form-group">
  <label>Warehouse ID</label>
  <input type="text" name="warehouseId" value={formData.warehouseId} disabled />
</div>
<div className="form-group">
  <label>Location</label>
  <input type="text" name="location" value={formData.location} onChange={handleChange} required />
</div>
<div className="form-group">
  <label>Medicine Name</label>
  <input type="text" name="medicineName" value={formData.medicineName} onChange={handleChange} required />
</div>
<div className="form-group">
  <label>Batch Number</label>
  <input type="text" name="batchNo" value={formData.batchNo} onChange={handleChange} required />
</div>
<div className="form-group">
  <label>Stock Quantity</label>
  <input type="number" name="stockQuantity" value={formData.stockQuantity} onChange={handleChange} required />
</div>
<div className="form-group">
  <label>Received Date</label>
  <input type="date" name="receivedDate" value={formData.receivedDate} onChange={handleChange} required />
</div>
<div className="form-group">
  <label>Expiry Date</label>
```

```
        <input type="date" name="expiryDate" value={formData.expiryDate} onChange={handleChange} required />
      </div>
      <div className="form-group">
        <label>Expiry Risk</label>
        <select name="expiryRisk" value={formData.expiryRisk} onChange={handleChange} required>
          <option value="High">High</option>
          <option value="Medium">Medium</option>
          <option value="Low">Low</option>
        </select>
      </div>
      <div className="form-group">
        <label>Supplier Name</label>
        <input type="text" name="supplierName" value={formData.supplierName} onChange={handleChange} required />
      </div>
     </div>
     <button type="submit" className="submit-btn">Add Record</button>
   </form>
  );
};

export default WarehouseForm;
```

**Barcode.jsx**

```
import React, { useRef, useState } from "react";
import { BrowserMultiFormatReader } from "@zxing/browser";
import "./BarcodeScanner.css";
import axios from 'axios';  //

const BarcodeScanner = () => {
  const fileInputRef = useRef(null);
```

```
const [result, setResult] = useState("");
const [preview, setPreview] = useState(null);
const [saveMessage, setSaveMessage] = useState("");

const handleImageUpload = async (event) => {
  const file = event.target.files[0];
  if (!file) return;

  const reader = new FileReader();
  reader.onload = async () => {
    const img = new Image();
    img.src = reader.result;
    setPreview(reader.result);

    img.onload = async () => {
      const codeReader = new BrowserMultiFormatReader();
      try {
        const barcode = await codeReader.decodeFromImageElement(img);
        setResult(barcode.text);

        // 📤 Send barcode to server
        await saveBarcode(barcode.text);

      } catch (error) {
        setResult("❌ Barcode not detected.");
      }
    };
  };
  reader.readAsDataURL(file);
};

const saveBarcode = async (code) => {
```

```
  try {
    const response = await axios.post('http://localhost:5000/api/barcodes', { code });
    setSaveMessage('☑ Barcode saved to database!');
  } catch (error) {
    setSaveMessage('❌ Failed to save barcode.');
    console.error(error);
  }
 };

 return (
  <div className="barcode-container">
   <h2>📷 Upload Barcode Image</h2>
   <input type="file" ref={fileInputRef} accept="image/*" onChange={handleImageUpload} />
   {preview && <img src={preview} alt="Preview" className="barcode-preview" />}
   <p className="barcode-result"><strong>Result:</strong> {result}</p>
   {saveMessage && <p className="barcode-save-message">{saveMessage}</p>}
  </div>
 );
};

export default BarcodeScanner;
```

**SERVER.js**

```
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');

const warehouseRoutes = require('./routes/warehouseRoutes');
const reportRoutes = require('./routes/reportRoutes');
const expiryRoutes = require('./routes/expiryRoutes');
const orderRoutes = require('./routes/orderRoutes');
```

```
const app = express();

// Middleware
app.use(cors());
app.use(express.json());

// MongoDB connection
mongoose.connect('mongodb://localhost:27017/Deadstock', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
})
.then(() => console.log('✅ Connected to MongoDB'))
.catch((err) => console.error('❌ MongoDB connection error:', err));

// Routes
app.use('/api/warehouse', warehouseRoutes);
app.use('/api/reports', reportRoutes);      // New: Reports API
app.use('/api/expiry', expiryRoutes);
 app.use('/api/orders', orderRoutes);
    // New: Expiry prediction API
const barcodeRoutes = require('./routes/barcodeRoutes');
app.use('/api/barcodes', barcodeRoutes);

// Root endpoint
app.get('/', (req, res) => {
  res.send('Welcome to Pharma Dead Stock Management API');
});

// Start the server
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log( Server running on port ${PORT}));
```

**OUTPUT:**

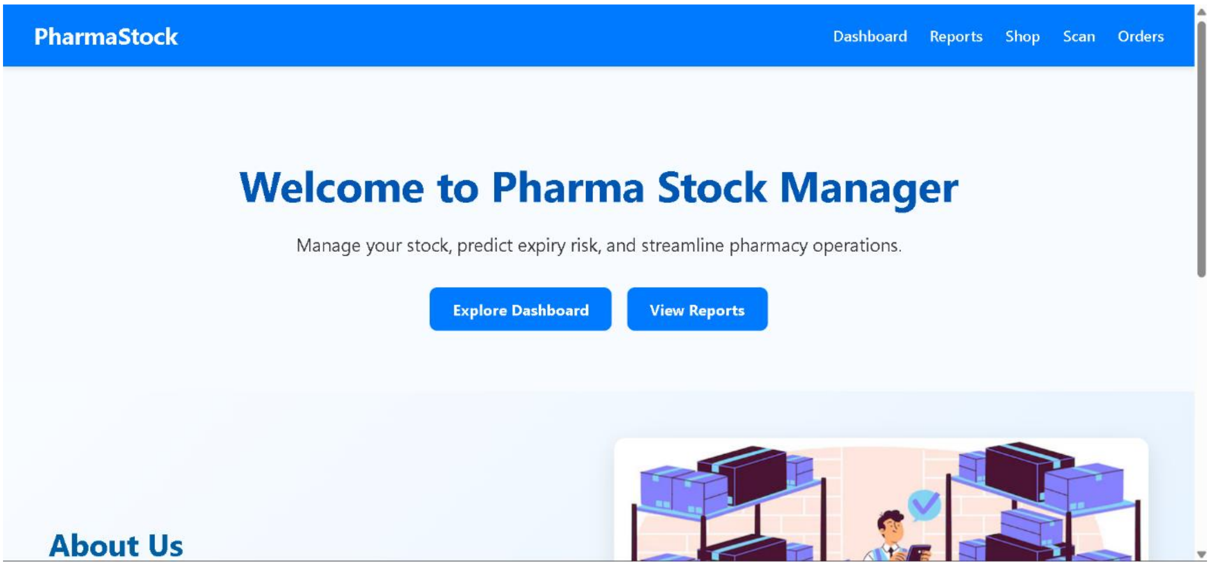



FIG 1.1 HOME PAGE

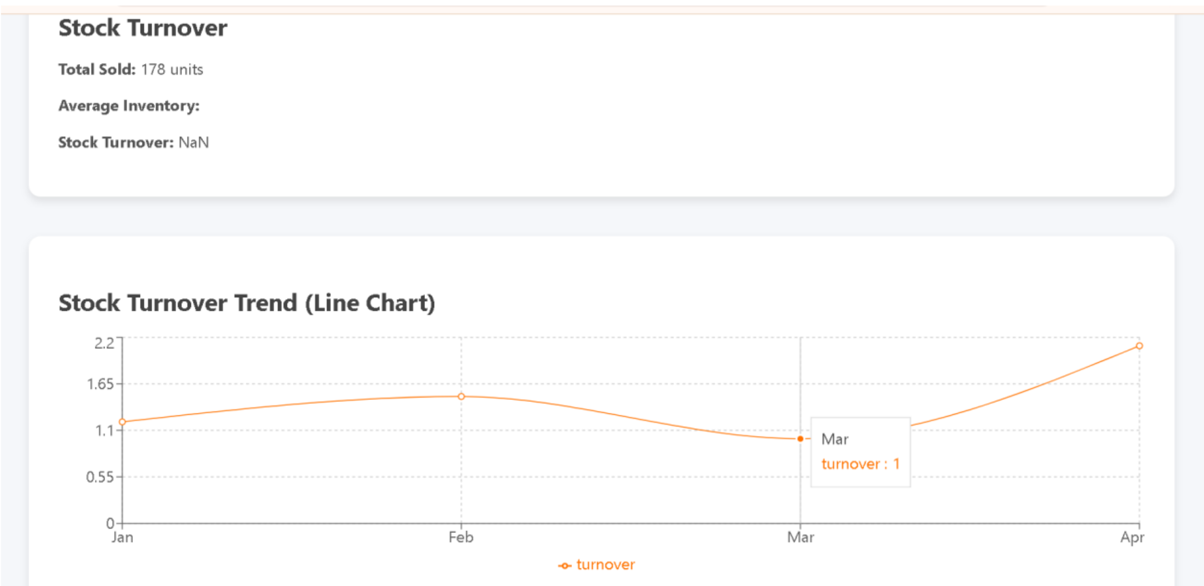
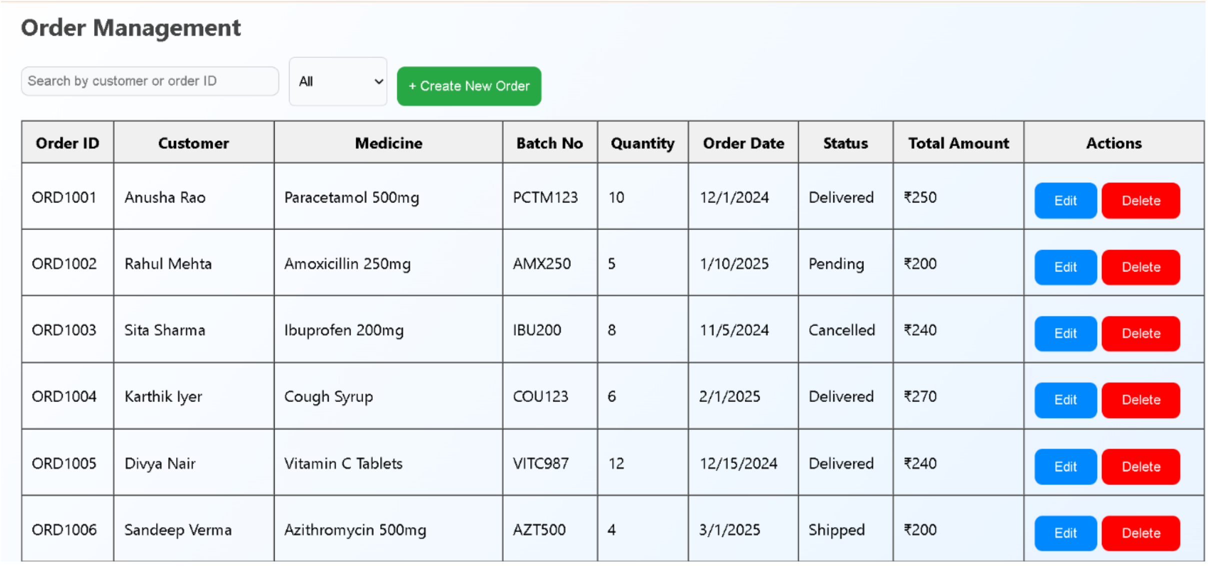



FIG 1.2 REPORT PAGE

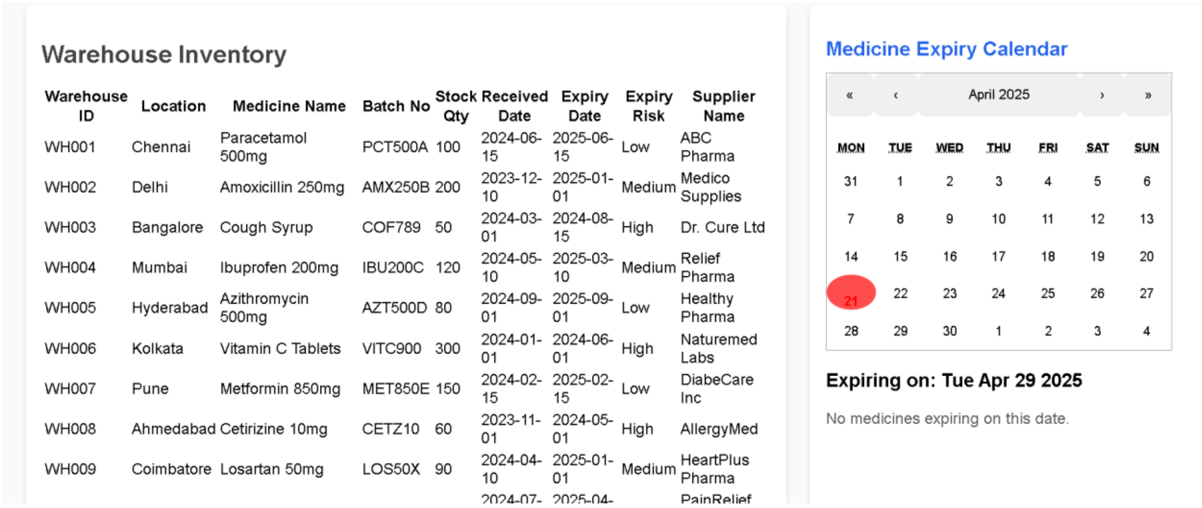


## Order Management

Search by customer or order ID | All | + Create New Order

| Order ID | Customer | Medicine | Batch No | Quantity | Order Date | Status | Total Amount | Actions |
|---|---|---|---|---|---|---|---|---|
| ORD1001 | Anusha Rao | Paracetamol 500mg | PCTM123 | 10 | 12/1/2024 | Delivered | ₹250 | Edit Delete |
| ORD1002 | Rahul Mehta | Amoxicillin 250mg | AMX250 | 5 | 1/10/2025 | Pending | ₹200 | Edit Delete |
| ORD1003 | Sita Sharma | Ibuprofen 200mg | IBU200 | 8 | 11/5/2024 | Cancelled | ₹240 | Edit Delete |
| ORD1004 | Karthik Iyer | Cough Syrup | COU123 | 6 | 2/1/2025 | Delivered | ₹270 | Edit Delete |
| ORD1005 | Divya Nair | Vitamin C Tablets | VITC987 | 12 | 12/15/2024 | Delivered | ₹240 | Edit Delete |
| ORD1006 | Sandeep Verma | Azithromycin 500mg | AZT500 | 4 | 3/1/2025 | Shipped | ₹200 | Edit Delete |

FIG 1.3 ORDER PAGE



## Warehouse Inventory

| Warehouse ID | Location | Medicine Name | Batch No | Stock Qty | Received Date | Expiry Date | Expiry Risk | Supplier Name |
|---|---|---|---|---|---|---|---|---|
| WH001 | Chennai | Paracetamol 500mg | PCT500A | 100 | 2024-06-15 | 2025-06-15 | Low | ABC Pharma |
| WH002 | Delhi | Amoxicillin 250mg | AMX250B | 200 | 2023-12-10 | 2025-01-01 | Medium | Medico Supplies |
| WH003 | Bangalore | Cough Syrup | COF789 | 50 | 2024-03-01 | 2024-08-15 | High | Dr. Cure Ltd |
| WH004 | Mumbai | Ibuprofen 200mg | IBU200C | 120 | 2024-05-10 | 2025-03-10 | Medium | Relief Pharma |
| WH005 | Hyderabad | Azithromycin 500mg | AZT500D | 80 | 2024-09-01 | 2025-09-01 | Low | Healthy Pharma |
| WH006 | Kolkata | Vitamin C Tablets | VITC900 | 300 | 2024-01-01 | 2024-06-01 | High | Naturemed Labs |
| WH007 | Pune | Metformin 850mg | MET850E | 150 | 2024-02-15 | 2025-02-15 | Low | DiabeCare Inc |
| WH008 | Ahmedabad | Cetirizine 10mg | CETZ10 | 60 | 2023-11-01 | 2024-05-01 | High | AllergyMed |
| WH009 | Coimbatore | Losartan 50mg | LOS50X | 90 | 2024-04-10 | 2025-01-01 | Medium | HeartPlus Pharma |
| | | | | | 2024-07- | 2025-04- | | PainRelief |



FIG 1.4 WAREHOUSE PAGE

## REFERENCES


[1] Sarda AN, Gharpure YJ. Trends in Inventory Management in health care sector in India Issues & Challenges, APOTHEOSIS: Tirpude's National Journal of Business Research. 2023; 4(2):59-67.

[2] Abbott RD, Friedman BA, Williams GW. Recycling older blood by integration into the inventory of a single large hospital blood bank: a computer simulation application. Transfusion. 2024; 18(6):709-15.

[3] Vila-Parrish AR, Ivy JS, King RE. A simulation-based approach for inventory modeling of perishable pharmaceuticals. In2018 winter simulation conference 2008 Dec 7 (pp. 1532-1538). IEEE.

[4] Vila-Parrish AR, Ivy JS, King RE, Abel SR. Patient-based pharmaceutical inventory management: a two-stage inventory and production model for perishable products with Markovian demand. Health Systems. 2012; 1(1):69-83.

[5] Atal A, Lee H, Ozer O. If the inventory manager knew: Value of RFID under imperfect inventory information. Working paper, Dept. of Management Science and Engineering, Stanford University, CA; 2005.

[6] Ballentine R, Ravin RL, Gilbert JR. ABC inventory analysis and economic order quantity concept in hospital pharmacy purchasing. American journal of hospital pharmacy. 1976; 33(6):552-5

[7] Civelek I, Karaesmen I, Scheller-Wolf A. Blood platelet inventory management with protection levels. European Journal of Operational Research. 2015; 243(3):826-38.

[8] Cakici OE, Groenevelt H, Seidmann A. Efficient inventory management by leveraging RFID in service organizations. In2010 43rd Hawaii International Conference on System Sciences 2010 Jan 5 (pp. 1-11). IEEE