Tribhuvan University

Institute of science and technology

Final Year Project on

PHARMATRACK: PHARMACY POINT OF SALE (POS) SYSTEM

(Subject Code: BIT404)

In partial fulfilment of the requirement for Bachelor's Degree in Information Technology (BIT)

Under the supervision of:

Mr. Bhim Bahadur Rawat

Department of Information Technology

Padmakanya Multiple Campus

Submitted by:

Sabu Bhuju (TU Reg: 5-2-38-339-2021)

Smriti Basu (TU Reg: 5-2-38-329-2021)

Kripa Banskota (TU Reg: 5-2-38-310-2021)

Submitted to

Department of Information Technology

Padmakanya Multiple Campus

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project report entitled "**PharmaTrack; Pharmacy Point of Sale(POS) System**" prepared under my supervision by **Ms. Sabu Bhuju**, **Ms. Smriti Basu** and **Ms. Kripa Banskota** in partial fulfillment of the requirements for the degree of Bachelor in Information Technology of Tribhuvan University be processed for review.

Bhim Bahadur Rawat
Faculty
Padmakanya Multiple Campus
(Project Supervisor)
November, 2025

LETTER OF APPROVAL

This is to certify that this project report prepared by **Ms. Sabu Bhuju**, **Ms. Smriti Basu** and **Ms. Kripa Banskota** entitled "**PharmaTrack; Pharmacy Point of Sale(POS) System**" in partial fulfillment of the requirements for the degree of Bachelor in Information Technology has been well studied. In our opinion, it is satisfactory in scope and quality as a project for the required degree.

Evaluation Committee

---	---
Bhim Bahadur Rawat (Project Supervisor)	Sunita Shrestha (Program Coordinator)
---	---
External Examiner	Internal Examiner

Date: **November, 2025**

DECLARATION

We hereby declare that this report entitled "**PharmaTrack; Pharmacy Point of Sale (POS) System**" submitted to the Department of Information Technology, Padmakanya Multiple Campus is my original work. We are the only author of this work and no sources other than listed here have been used in this work. This work has been carried out in this form in partial fulfillment of the requirements of the Bachelor's Degree in Information Technology (BIT) under the guidance of **Mr. Bhim Bahadur Rawat.**

_____

Authors:

Sabu Bhuju (5-2-38-339-2021)

Smriti Basu (5-2-38-329-2021)

Kripa Banskota (5-2-38-310-2021)

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all the individuals and organizations who supported and contributed to the successful completion of our project titled "**PharmaTrack: Pharmacy Point of Sale (POS) System**".

We are deeply indebted to our supervisor, **Mr. Bhim Bahadur Rawat**, for his invaluable guidance, continuous support, and insightful suggestions throughout the course of this project. His expertise and encouragement were instrumental in helping us navigate challenges and refine our work.

Our heartfelt thanks also go to **Mrs. Sunita Shrestha**, Head of the BIT Department, for her constant inspiration and for fostering an environment of learning and innovation that greatly influenced the outcome of our project.

Finally, we would like to extend our appreciation to our friends and peers for their constructive feedback, encouragement, and unwavering support, which kept us motivated and helped us bring this project to fruition.

Sabu Bhuju (5-2-38-339-2021)

Smriti Basu (5-2-38-329-2021)

Kripa Banskota (5-2-38-310-2021)

ABSTRACT

This project presents **PharmaTrack: Pharmacy Point of Sale (POS) System**, a user-friendly web-based solution designed to streamline pharmacy operations and enhance overall efficiency. PharmaTrack facilitates accurate and real-time management of inventory, sales, and customer transactions, ensuring both convenience and reliability for pharmacists. The system integrates advanced features such as automated stock alerts, expiry tracking, and sales analytics to help pharmacies maintain optimal inventory levels and minimize waste.

The system incorporates data-driven features to optimize pharmacy workflows. A greedy algorithm-based stock alert mechanism prioritizes products that are low in stock or nearing expiry, ensuring timely restocking and minimizing financial loss. Additionally, PharmaTrack employs a fuzzy search approach for product lookup, allowing pharmacists to quickly find items even with partial or approximate information, thereby reducing search time and improving customer service.

By combining inventory control, sales automation, and data-driven insights, PharmaTrack aims to optimize pharmacy workflows and decision-making. It serves as a reliable, efficient, and innovative tool for modern pharmacy management, enhancing operational efficiency while promoting better customer service.

Keywords: *Pharmacy POS system, Stock alerts, Greedy algorithm approach, Fuzzy Search approach, Expiry tracking, Data-driven insights*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

POS: Point of Sale

PHP: Hypertext Preprocessor

CSS: Cascading Style Sheets

SQL: Structured Query Language

CRUD: Create, Read, Update, Delete

JSON: JavaScript Object Notation

API: Application Programming Interface

ER: Entity-Relationship

DFD: Data Flow Diagram

UI: User Interface

RBAC: Role-Based Access Control

CHAPTER 1: INTRODUCTION

1.1 Introduction

In the sensitive sector like pharmacy, where transactions of medicines take place regularly, even a single mistake can cause severe problems. It is therefore essential that all records are managed carefully with extreme precautions. In Nepal, most pharmacies still maintain data manually, which creates a high risk of human error whenever new entries are made. Retrieving data is time-consuming, and generating meaningful information from the entered records requires additional manual effort.

With advancements in information technology and the availability of computer systems, it has become necessary for pharmacies to adopt digital solutions to replace traditional practices. While many other sectors in Nepal are embracing modern technology, most pharmacies continue to rely on older methods. A **Pharmacy Point of Sale (POS) System** comes in handy to automate record-keeping, data retrieval, and the generation of meaningful insights from stored data.

PharmaTrack is a web-based pharmacy POS system designed to keep pharmacy records securely while providing easy access whenever needed. The system automates the daily tasks of a pharmacy shop through a proper application interface, database, and computer system. It integrates greedy algorithm-based stock alerts to prioritize products that are low in stock or nearing expiry and fuzzy search functionality to allow pharmacists to quickly find products even with partial or approximate information.

PharmaTrack also maintains a complete transaction history for every sale, enabling pharmacies to track purchases accurately and generate detailed reports. Additionally, the system can generate digital receipts for each transaction, ensuring proper documentation and smooth communication with customers.

By replacing the old manual record-keeping system with PharmaTrack, the efficiency of pharmacy operations is significantly increased. Data is stored securely in the database, ensuring safety and accuracy, while the intuitive and user-friendly interface enables any

type of user to operate the system without special technical knowledge. This makes PharmaTrack an easily adaptable and effective solution for pharmacies in Nepal

1.2 Problem Statement

The pharmacy sector, which plays a critical role in healthcare delivery, is currently hampered by manual and fragmented processes for managing inventory, sales, and customer transactions. Although pharmacies handle medicines and health products on a daily basis, the reliance on manual record-keeping introduces significant risks of human error, leading to potential financial loss, expired stock, and compromised customer service.

Pharmacists struggle to track stock levels efficiently because existing manual systems are time-consuming and lack automation, making it difficult to identify products that are low in stock or nearing expiry. Product searches are often slow and imprecise, especially when names are entered incorrectly or partially, which reduces workflow efficiency and frustrates both staff and customers. Additionally, maintaining accurate transaction histories and generating receipts manually is cumbersome, limiting the ability to produce timely reports for decision-making and compliance.

There is therefore a critical need for an integrated, secure, and intelligent digital system to address these issues. Such a system should combine inventory management, easy product lookup, automated transaction history, and digital receipt generation into a single, user-friendly platform**.** By doing so, pharmacies can minimize errors, optimize workflow, improve customer satisfaction, and ensure reliable and efficient record-keeping.

1.3 Objectives

The primary objective of this project was to design and develop a fully functional prototype of the **PharmaTrack,** Pharmacy POS system. The key specific objectives include:

1. To implement core functionalities such as stock tracking, expiry alerts, billing automation, and transaction logging to streamline medicine handling.
2. To ensure data accuracy and operational efficiency by minimizing manual errors and enabling real-time updates of inventory and sales records.

3. To create a user-friendly interface that allows system operators to manage stock, generate bills, and monitor transactions with minimal training and maximum clarity.
4. To build a scalable and modular system architecture that supports future enhancements such as report generation, barcode scanning, and multi-user access.

1.4 Scope and Limitation

A. Scope: Core Deliverables

The application covers the end-to-end development of the Pharmacy POS System, including:

- User-friendly web interface for pharmacists to manage sales and customer transactions.
- Product search functionality with support for partial or approximate matches.
- Recording of all transactions with a complete transaction history.
- Generation of digital receipts for every sale.
- Basic sales reporting and analytics for tracking revenue and sales trends.
- Secure access and authentication for authorized users only.
- A responsive web frontend and backend capable of handling small to medium-sized pharmacy operations.

B. Limitations

The project is bound by the following limitations, which define its current scope:

- **Payment Integration:** The system currently does not support online payment gateways. In a production-ready system, secure payment integration would be required for handling digital transactions.
- **Supplier Communication:** The prototype cannot directly contact suppliers or automate supplier orders. Integration with supplier management systems would be needed for full automation.

- **Advanced Analytics:** The current system provides basic sales reporting but does not include predictive analytics, stock forecasting, or AI-based recommendations.
- **Offline Functionality:** The system requires an active internet connection; offline operation is limited.
- **Scalability and Testing:** The application has been tested primarily for small to medium-sized pharmacies. Its performance and reliability at large scale or under high traffic have not been validated.

1.5 Development Methodology

The development of **PharmaTrack** followed the **Waterfall Methodology**, a linear and sequential approach suitable for projects with clearly defined requirements. This methodology ensures that each phase is completed before moving on to the next, providing a structured and organized development process. The project was executed in the following phases:

1. Requirement Analysis: Detailed requirement gathering was conducted through observation of pharmacy workflows and consultations with pharmacists. The main functionalities identified included sales management, product search, transaction history, digital receipts, and basic sales reporting.

2. System Design: The system was designed using a three-tier architecture consisting of the frontend, backend, and database**.** DFD (Data Flow Diagrams) were created to understand how data moves through the system, while UML diagrams such as Use Case were used to model user interactions and system workflows.

3. Implementation: The frontend was developed using React.js for building interactive and component-based user interfaces, with Tailwind CSS providing responsive and clean styling. The backend was implemented using PHP **and** MySQL, handling business logic, transaction processing, and secure data storage. Each module, including product search, transaction recording, and receipt generation, was implemented sequentially according to the Waterfall approach.

4. Testing: After implementation, the system underwent unit testing for individual modules and integration testing for the complete system. Testing focused on the accuracy

of transaction history, correctness of digital receipts, and the functionality of product search.

5. Deployment and Maintenance: The system is ready for deployment on a web server for practical use in pharmacies. Maintenance procedures have been defined to address future updates or bug fixes once the system is deployed.

The Waterfall methodology provided a systematic and disciplined approach, resulting in a fully functional, reliable, and user-friendly Pharmacy POS system suitable for small to medium-sized pharmacies.

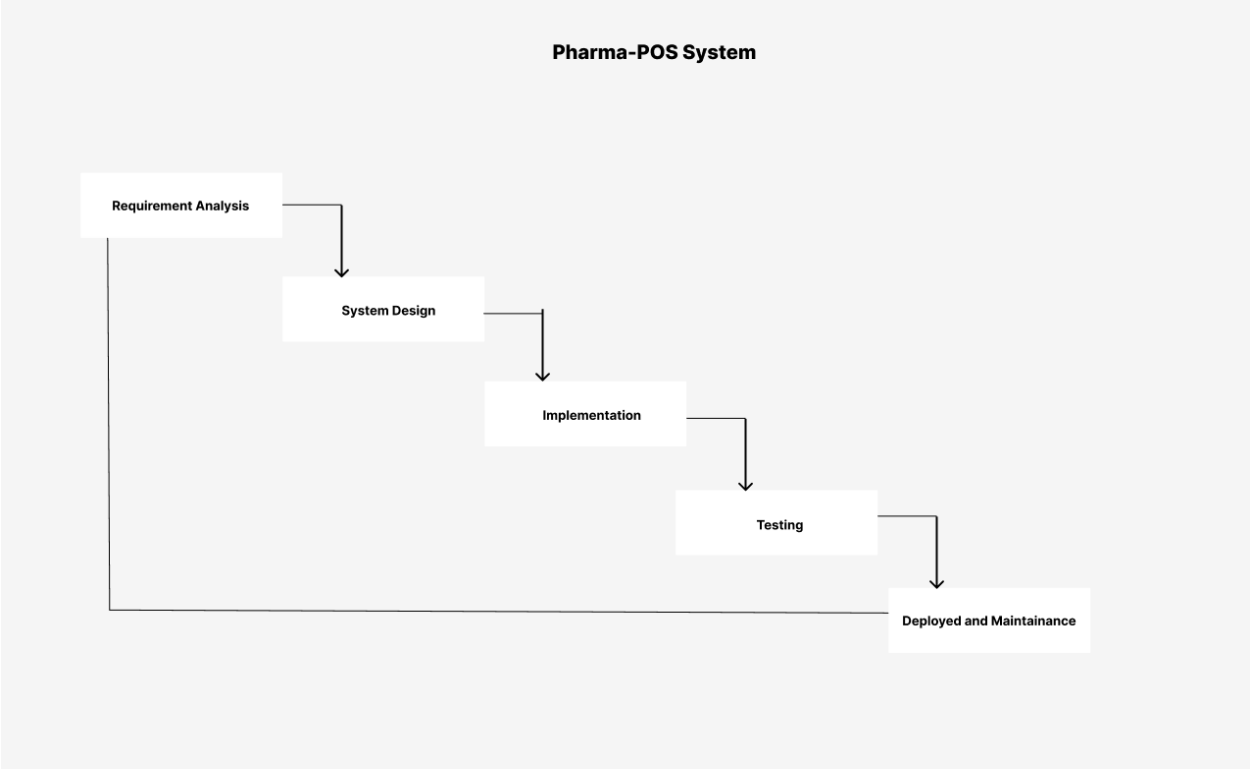


Figure 1: Development Methodology of PharmaTrack.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

The digital transformation in healthcare and the increasing need for accurate and efficient pharmacy operations have fundamentally changed the way pharmacies manage their stock, sales, and transactions. Traditional manual systems are prone to errors, inefficient record keeping, and delayed reporting, which can impact patient safety and operational efficiency. Digital pharmacy management systems address these challenges by providing structured, reliable, and automated solutions that streamline workflows and enhance service quality.

2.1.1 Pharmacy Management and Digital Automation

Pharmacy management involves tracking medicine stock, monitoring expiry dates, recording sales transactions, and maintaining customer records. Automated systems contribute to efficiency by:

- Reducing manual errors and minimizing risks of dispensing expired or incorrect medicines.
- Generating detailed transaction histories and digital receipts for transparency and accountability.
- Enabling timely reporting and decision-making based on collected operational data.

Despite these advantages, traditional pharmacy systems often face challenges such as incomplete stock data, lack of integrated alerts for expiry or low stock, and difficulties in quickly retrieving sales information. Modern systems aim to resolve these issues through centralized digital management platforms.

2.1.2 Key Concepts and System Fundamentals

The success of a digital pharmacy management system relies on several foundational components:

- **Secure User Authentication:** Ensures that only authorized staff can access sensitive pharmacy data.

- **Stock Monitoring and Alerts:** Tracks medicine quantities and expiry dates, generating notifications for critical items.
- **Transaction Recording:** Maintains detailed sales records to allow auditing and financial reporting.
- **Customer Management:** Stores relevant customer information to support loyalty programs and repeat sales.
- **Data Reporting:** Provides insights into sales trends, stock consumption, and operational efficiency.

These core principles form the basis for reliable, safe, and efficient pharmacy operations.

2.1.3 Data-Driven Decision Making in Pharmacies

Digital pharmacy systems rely on the use of stored data to optimize workflows and improve efficiency:

- **Inventory Analytics:** Helps in predicting demand and preventing stockouts or overstocking.
- **Transaction Analysis:** Tracks sales patterns to identify frequently sold medicines and peak periods.
- **Alert Prioritization:** Ensures that critical items requiring attention, such as near-expiry medicines, are addressed promptly.

By integrating these data-driven approaches, pharmacies can operate more safely, reduce waste, and maintain high service standards.

2.2 Literature Review

2.2.1 Studies on Pharmacy POS Systems

Research on pharmacy POS systems shows that these platforms have evolved from basic transaction tools to integrated solutions combining inventory control, electronic prescription handling, and medication management [1]. Modern POS systems enhance operational efficiency by supporting automated billing, real-time stock updates, and customer management [2]. The transition to cloud-based systems has allowed scalability, remote access, and synchronization across multiple branches, which are critical for

telepharmacy and chain pharmacies. Incorporating AI and predictive analytics enables accurate forecasting of stock requirements, reduces wastage, and optimizes inventory. Case studies demonstrate benefits such as up to 25% reduction in inventory discrepancies, faster prescription processing, improved workflow across multiple locations, and enhanced regulatory compliance through automated reporting of controlled substance [3].

2.2.2 Operational and Implementation Studies

Comparative studies highlight that successful pharmacy management system implementation relies on automation, multi-platform accessibility, and real-time monitoring of sales and inventory [4]. Active involvement of management, proper staff training, and technical expertise are crucial for smooth adoption. Research shows that digital systems positively impact workflow, minimize human errors in transactions, and improve the accuracy of stock and financial records.

2.2.3 Effectiveness of Automated Billing & POS in Retail Pharmacies

Studies comparing manual billing with POS-based billing systems show that digital billing significantly reduces transaction times, resulting in shorter waiting periods and improved customer satisfaction. POS systems eliminate pricing errors, support barcode scanning, and automatically update stock levels with each sale, reducing mismatches between physical and recorded inventory. Pharmacies using POS technology also experience improved accuracy in daily sales reporting and financial auditing [1].

2.2.4 Importance of Real-Time Stock Monitoring

Real-time stock visibility plays a critical role in preventing stock outs and ensuring timely medicine availability. Studies show that pharmacies maintaining real-time inventory records are better positioned to reorder medicines before shortages occur. When pharmacists can view livestock updates, they avoid over-ordering and can plan purchases efficiently, reducing emergency shortages and improving patient trust. Real-time monitoring also reduces the need for frequent manual stock audits, saving time and effort [5].

2.2.5 Customer Record Management in Pharmacies

Maintaining customer records allows pharmacies to track purchase history, preferred medicines, and potential allergies. Research suggests that pharmacies that maintain digital customer profiles provide better personalized care, such as refill reminders, alerts about medicine changes, and identification of potential drug interactions [6]. Customer databases also support quick retrieval of past invoices and improve service quality during follow-up visits.

2.2.6 Importance of Structured Workflow in Pharmacies

Operational research indicates that pharmacies with structured workflows such as the sequence of billing, stock deduction, expiry review, and reorder analysis perform better than those using informal or inconsistent processes. Structured workflows reduce confusion and ensure that every step is carried out in a timely manner. By reducing human dependency in critical tasks, pharmacies can maintain consistent quality, reduce stock discrepancies, and enhance overall service efficiency [7].

2.2.7 FEFO and Expiry Management

FEFO (First-Expire-First-Out) is essential in pharmacies because medicines have different shelf lives, and using the items closest to expiry first helps reduce wastage and ensures patient safety. Studies show that pharmacies relying only on FIFO often experience higher rates of expired stock [8]. Expired medicines lead to avoidable financial losses, especially when inventory is tracked manually and expiry dates are overlooked. Systems that highlight near-expiry batches and send expiry alerts help pharmacies act early by prioritizing sales or returning stock significantly reducing both wastage and financial loss [9].

CHAPTER 3: SYSTEM ANALYSIS

3.1 System Analysis

System analysis involves understanding the functional behavior, performance expectations, feasibility, and structural modelling of the system. This chapter presents a detailed examination of the requirements, feasibility dimensions, and analytical modelling applied during the development of **PharmaTrack: Pharmacy Point of Sale (POS) System.**

3.1.1 Functional Requirements

Functional requirements describe the expected behavior and core functionalities of the **PharmaTrack** system. These define how the software should respond to user inputs and support daily pharmaceutical operations.

User Authentication

- The system provides secure login and registration features.
- New users must create an account to access the system.
- Upon successful authentication, users gain access to all operational modules.

Stock Management

- Users can add new medicines with details such as name, quantity, expiry date, and price.
- The system updates stock levels in real time and displays current inventory status.

Billing and Sales Processing

- Users can generate bills by selecting medicines from the inventory.
- The system calculates total cost, applies discounts if applicable, and generates itemized receipts.
- Each transaction is logged for future reference.

Expiry Alert System

- The system monitors expiry dates and alerts users when medicines are nearing expiration.

- Expired items are flagged and excluded from billing operations.

Transaction History and Reporting

- Users can view past transactions, including bill details, dates, and items sold.
- The system supports basic reporting features such as daily sales summaries and stock movement logs.

Dashboard and Visual Insights

- The system provides a dashboard with visual charts showing stock levels, sales trends, and expiry statistics.
- Users can view summaries of top-selling items and low-stock alerts.

Logout

- The system includes a logout feature to securely end the user session.

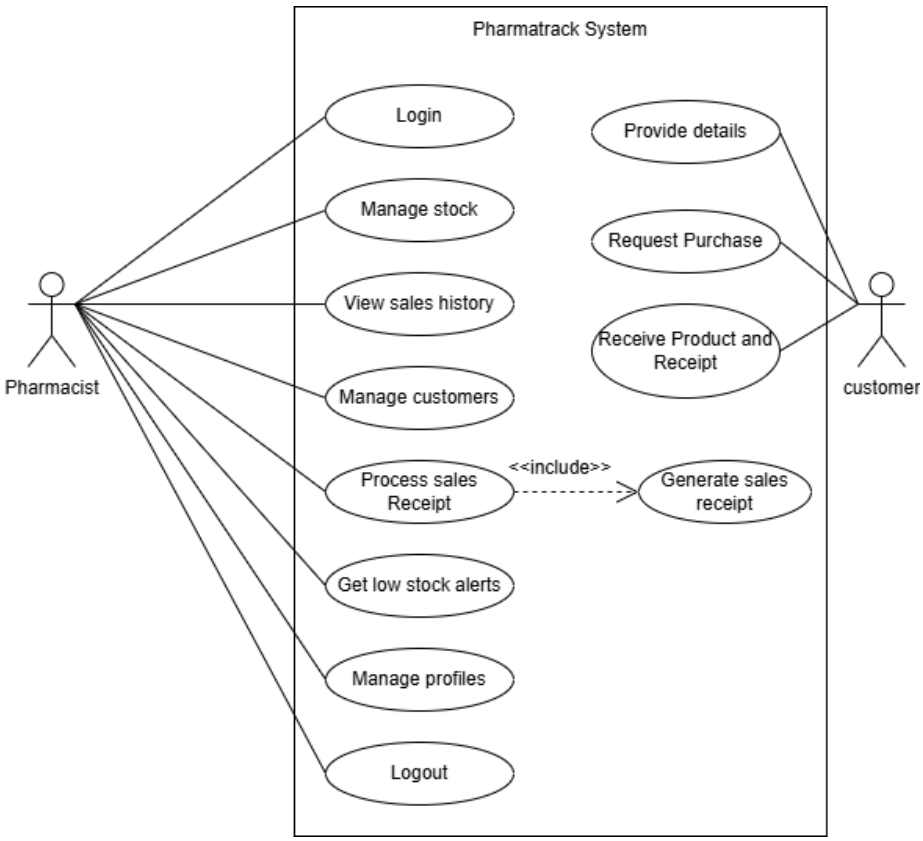


Figure 2: Use Case Diagram of PharmaTrack

3.1.2 Non-Functional Requirements

Non-functional requirements define the overall quality attributes of **PharmaTrack**, ensuring the system is reliable, efficient, and user-friendly in standalone pharmaceutical environments.

(a) Scalability

- The system must support increasing volumes of medicines, transactions, and users without performance degradation.
- It should allow future expansion, such as multi-user access, barcode integration, and report generation.

(b) Usability

- The user interface must be intuitive and easy to navigate, enabling users with minimal technical background to operate the system efficiently.
- Clear labels, responsive forms, and guided workflows should enhance user experience.

(c) Performance

- The system should respond to user actions (e.g., billing, inventory updates) within 2 seconds under normal load.
- It must handle concurrent operations smoothly and maintain high availability during peak usage.

3.1.3 Feasibility Study

The development and deployment of the **PharmaTrack: Pharmacy POS System** requires a comprehensive feasibility analysis to ensure its practicality, sustainability, and effectiveness in standalone pharmaceutical environments. The following dimensions were evaluated:

i. Technical Feasibility

The system's architecture is stable and manageable since it is made with well-known technologies:

- Backend: PHP is a popular choice for web application development because of its ease of use, adaptability, and MySQL compatibility.
- Frontend: React.js and Tailwind CSS work together to provide dynamic content updates and responsive design.
- Database: XAMPP-hosted MySQL provides a dependable and safe database management environment.

ii. Operational Feasibility

PharmaTrack is designed to be operationally feasible for users managing pharmaceutical inventory independently. Its intuitive interface allows users with minimal technical expertise to perform tasks such as medicine entry, billing, and expiry tracking efficiently. The system enhances accuracy, reduces manual workload, and improves overall workflow in self-contained setups.

iii. Economic Feasibility

The project is economically viable, as it leverages open-source tools and frameworks for development. No proprietary software licenses are required, and the system can be deployed on existing hardware infrastructure. This minimizes implementation costs and makes PharmaTrack accessible for academic, experimental, or small-scale retail use.

iv. Schedule Feasibility

Schedule feasibility guarantees the project tasks to be completed within the allotted time frames. The project timetable based on iteration wise is shown in the following table:

Task	Start Week	Duration (Weeks)
Requirement Analysis	1	1
System Design	2	1
Implementation	3	4
Testing	7	1
Deployment and maintainance	8	1
Documentation	1	9

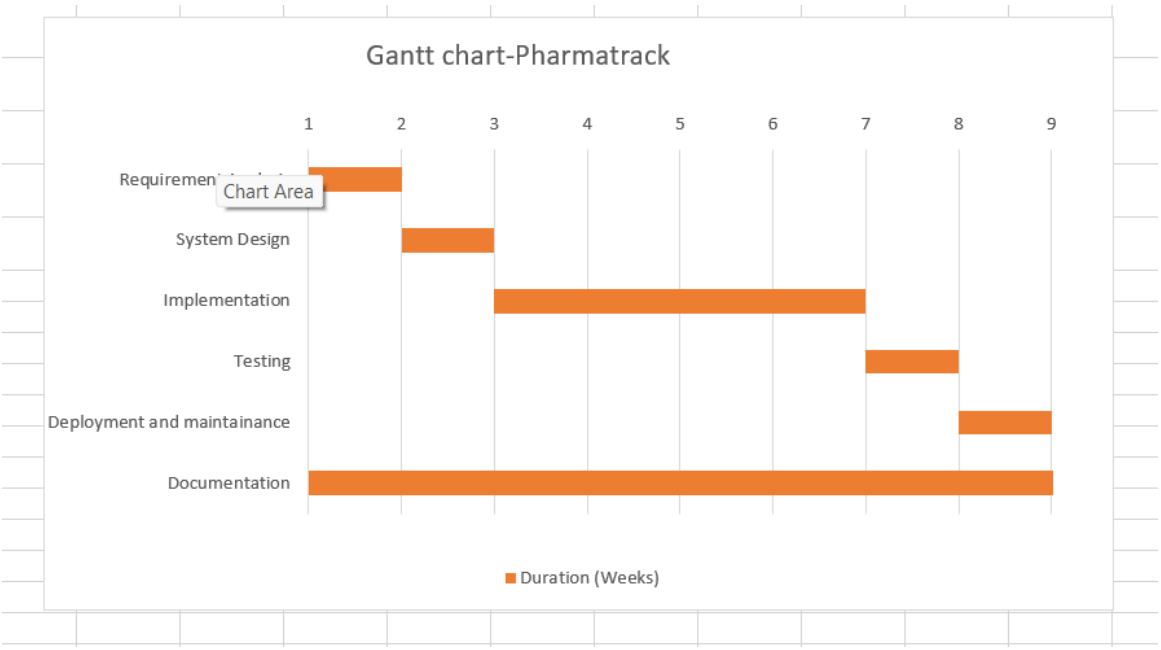


Figure 3: Gantt chart

The PharmaTrack project follows a structured development timeline starting with **Requirement Analysis** in Week 1, lasting 1 week. At the same time, **Documentation** begins in Week 1 and continues throughout the project for 9 weeks to maintain updated records.

In Week 2, the **System Design** phase is carried out for 1 week, focusing on defining the system architecture and workflow. This is followed by the **Implementation** phase from Week 3 to Week 6, lasting 4 weeks, where all system modules are developed and integrated.

Once development is complete, **Testing** is conducted in Week 7 for 1 week to ensure functionality, reliability, and system stability. Finally, in Week 8, the **Deployment and Maintenance** phase takes place for 1 week, where the system is deployed and initial support activities are performed.

This timeline provides a clear sequence of tasks, ensuring smooth progress from planning to final deployment.

3.1.4 Analysis

In this project, a structured method is used for both analysis and design phases. The approach focuses on organizing and documenting the system in a systematic which provides clarity in data and process flow.

In the analysis phase, the structured approach helps in thoroughly understanding and representing the system's requirements. In this method ER diagrams and Data Flow Diagrams (DFD) are used to model data and process of the systems. These tools allow for a clear representation of the data relationships and the flow of information between various system components. This phase provides a solid foundation for designing the database and system architecture.

In the design phase, the structured method converts the analysis into a detailed system blueprint. This includes designing the database schema, flow charts, creating forms and reports, and defining user interfaces and dialogue for smooth iteration to the system. The objective is to improve the system's architecture and make sure every part follows with the original requirements of the system.

3.1.3.1 ER Diagram

The Entity Relationship (ER) diagram depicts the entities, their attributes, and the relationships that connect them. It emphasizes the one-to-many interactions between different entities like users, customers, sales, products etc.

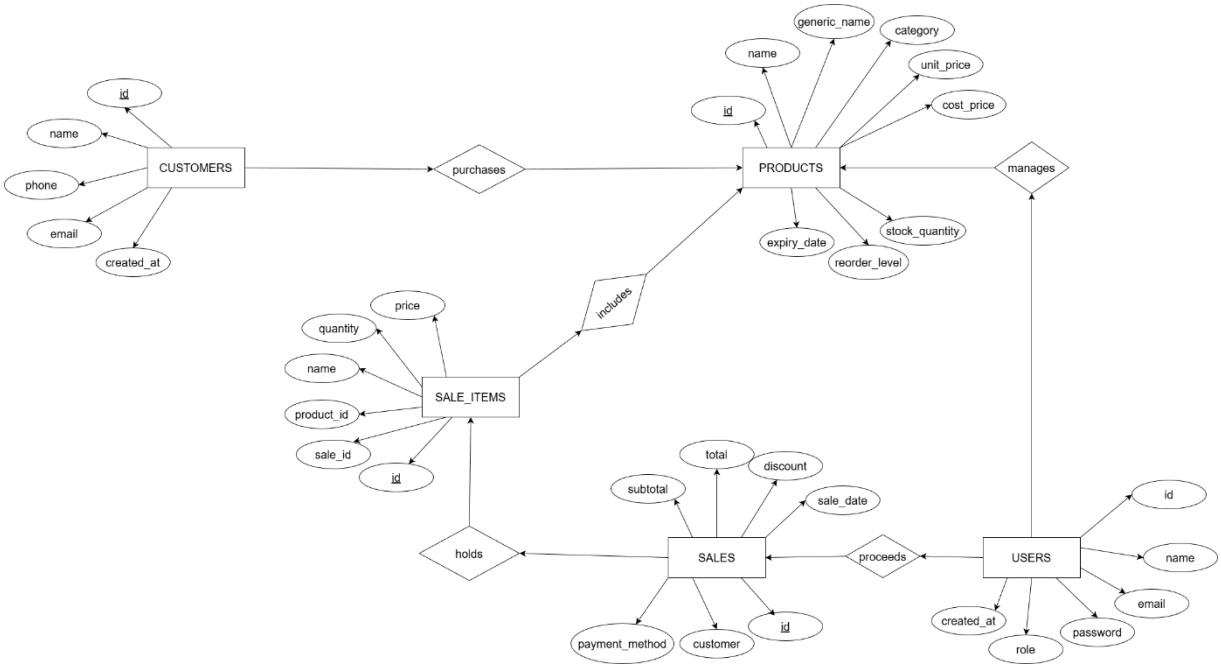


Figure 4: ER Diagram of PharmaTrack

3.1.3.2 DFD Diagram

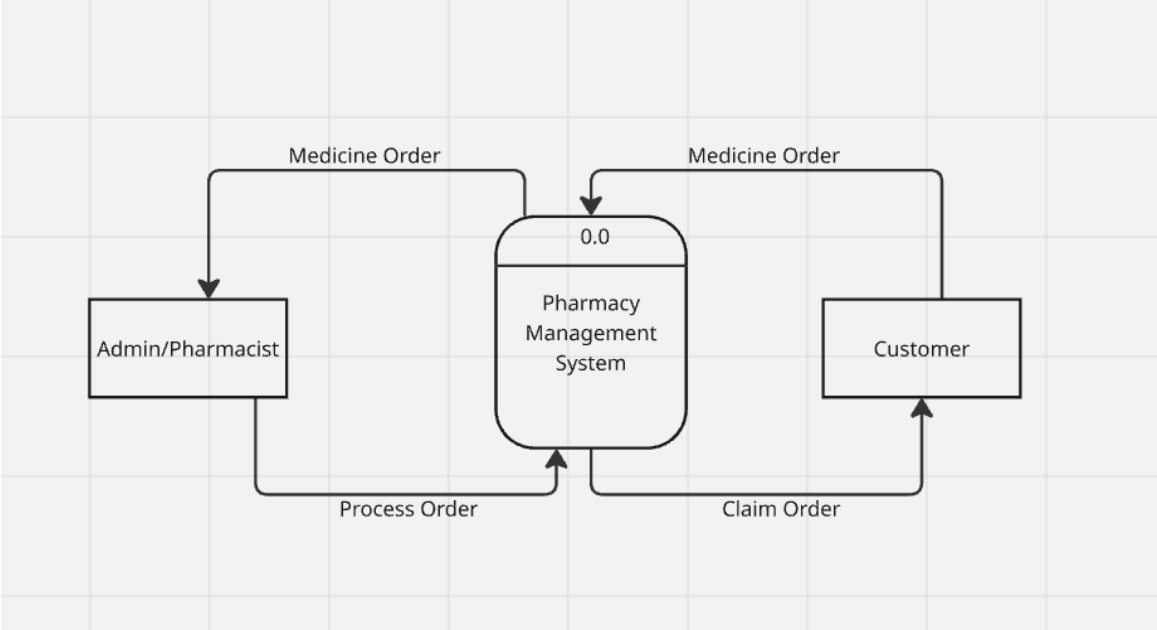


Figure 5: DFD Level 0 of PharmaTrack

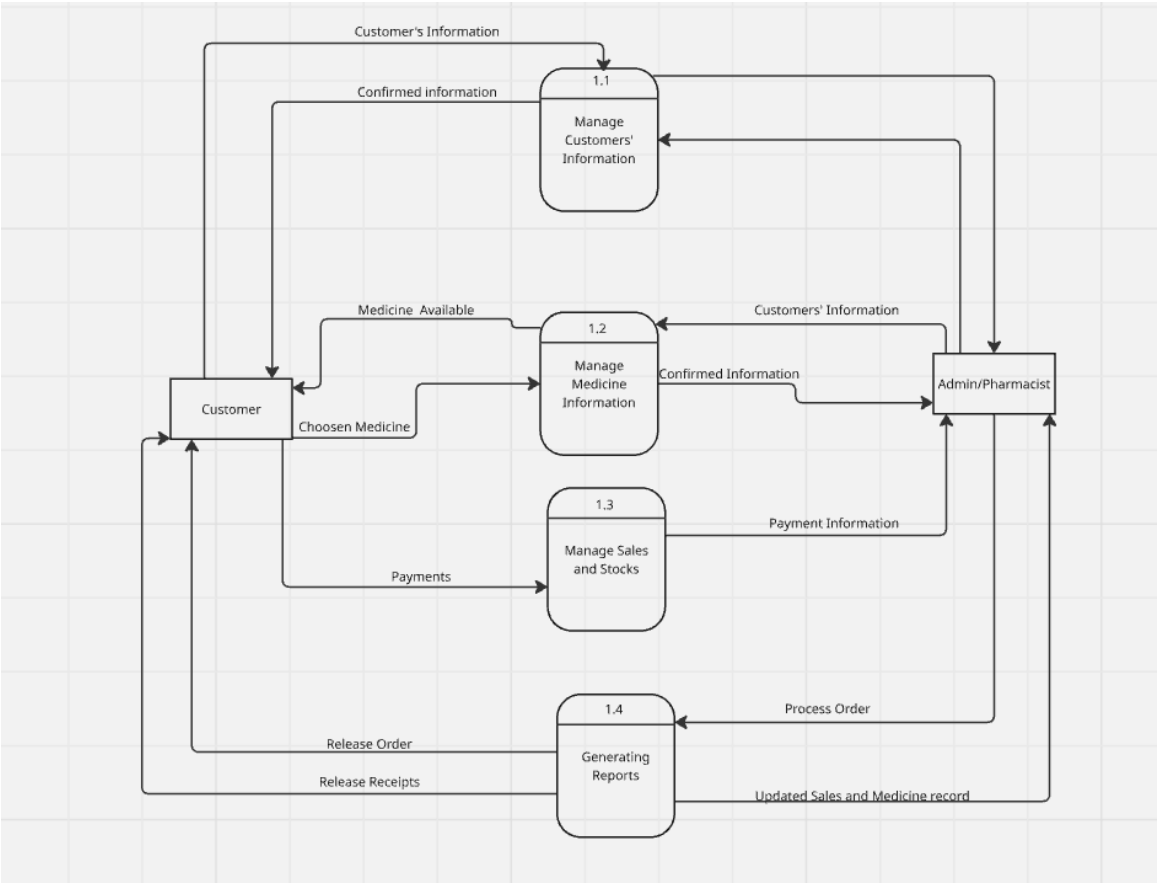


Figure 6: DFD Level 1 of PharmaTrack

CHAPTER 4: SYSTEM DESIGN

The system design outlines the components, architecture, and methods needed to construct the PharmaTrack system. This project uses structured method in both design and analysis where system is designed and analyzed in a systematic manner. This chapter examines the database structures, user interface designs, and architectural framework to give a comprehensive overview of the system's operation and structure.

4.1 Architectural Design

The system architecture design for PharmaTrack follows a client-server architecture**.** PharmaTrack's modular client-server structure enhances scalability and maintainability**,** ensuring that different components—such as user management, transactions, medicine records, and stock alerts—interact smoothly. This design supports efficient data processing, real-time access to critical information, and a responsive user experience for pharmacy staff, enabling secure and reliable management of pharmacy operations.

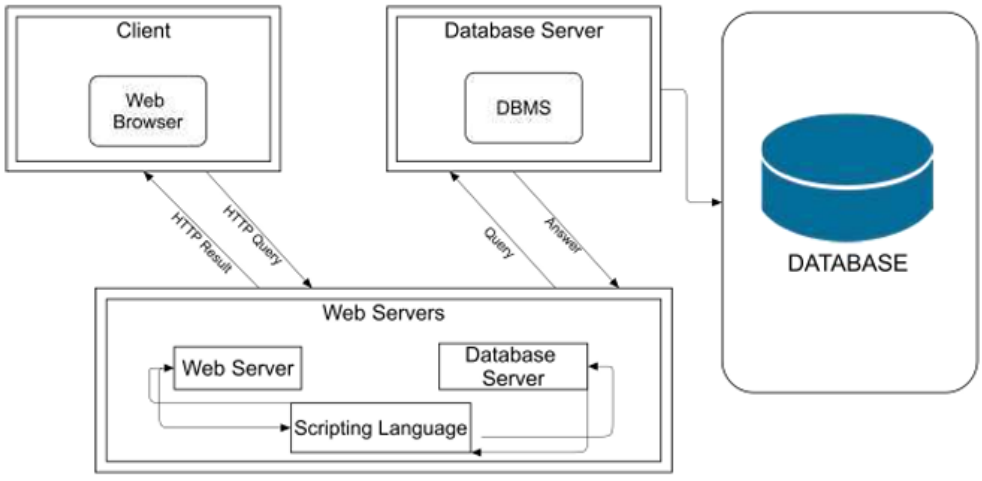


Figure 6: 3 tier client- server Architecture

In the above figure, it can be observed that the client and database server layers do not communicate directly; instead, a middleware layer facilitates this interaction. The main purpose of this layer is to allow the client to send requests, which are processed by the middleware and routed to the database server. The server's response then passes back through the middleware, which formats the data before sending it to the client.

In the context of PharmaTrack, the client corresponds to a web browser used by pharmacy staff, while the middleware layer is the web server handling business logic and data processing. This model is particularly suitable for dynamic web applications like a pharmacy management system, where transactions, stock updates, and alerts need real-time processing.

For data storage, MySQL is used as the database management system to ensure reliable access to user data, customer records, product information, sales transactions, and stock alerts. This architecture enables secure, efficient, and responsive operations, supporting the core functionality of PharmaTrack.

4.1.1 Database Design

The main goal of database design is to provide a schema that guarantees effective data management and organization. The schema supports the features of the system and is optimized for performance and scalability

Database Tables

- **Users Table:** Stores user credentials and basic profile information, such as name, email, password, and role (e.g., pharmacist, admin).
- **Customers Table:** Records customer details, including name, contact information, and purchase history.
- **Products Table:** Contains details of medicines or products available in the pharmacy, including name, category, price, quantity, and expiry date.
- **Sales Table:** Tracks each sale transaction, including the transaction ID, date, customer, and total amount.
- **Sale_Items Table:** Records the individual products included in each sale, linking products to specific sales with quantity and price information.

This table structure supports the core functionality of PharmaTrack, enabling **s**ecure user management, transaction tracking, and stock monitoring**.**

4.1.1.1. Schema Diagram

The schema diagram shows the links between database tables, highlighting foreign key constraints and data integrity.

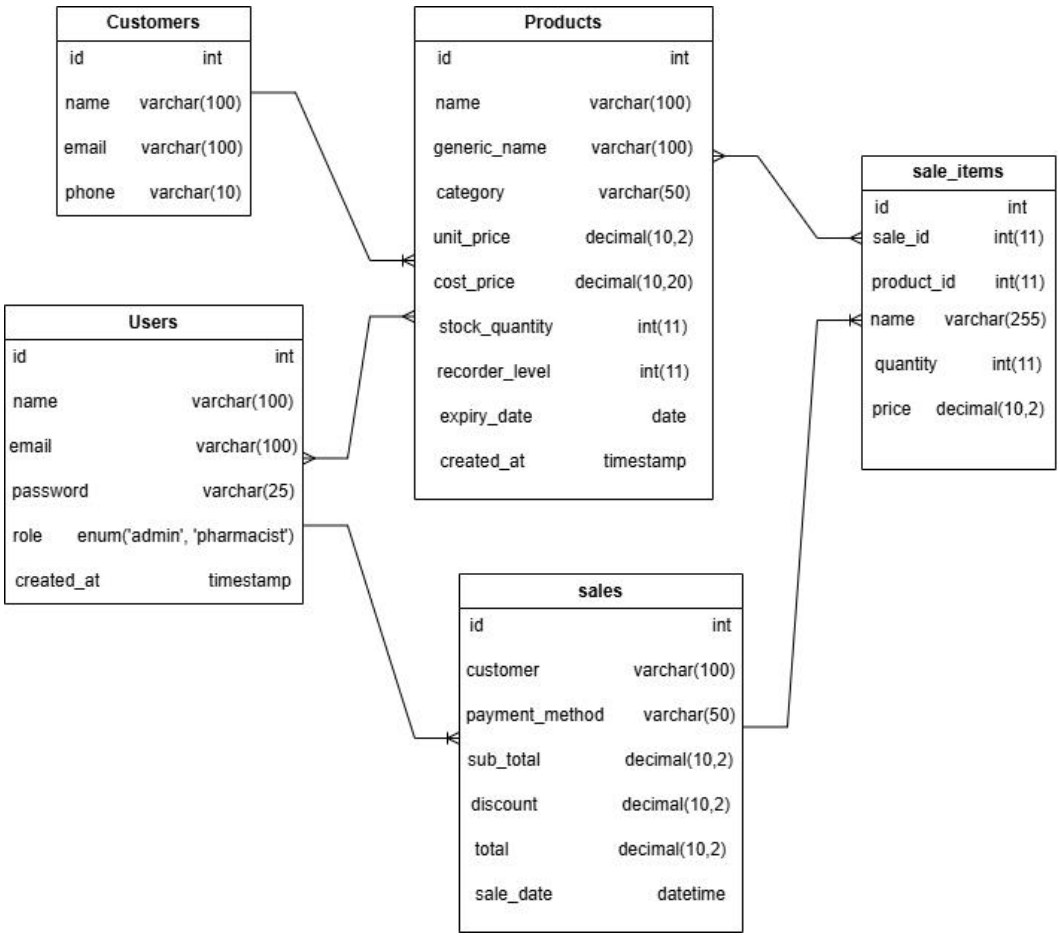


Figure 7: Schema Diagram of PharmaTrack POS System

4.1.2 Forms and Report Design

Registration Form:

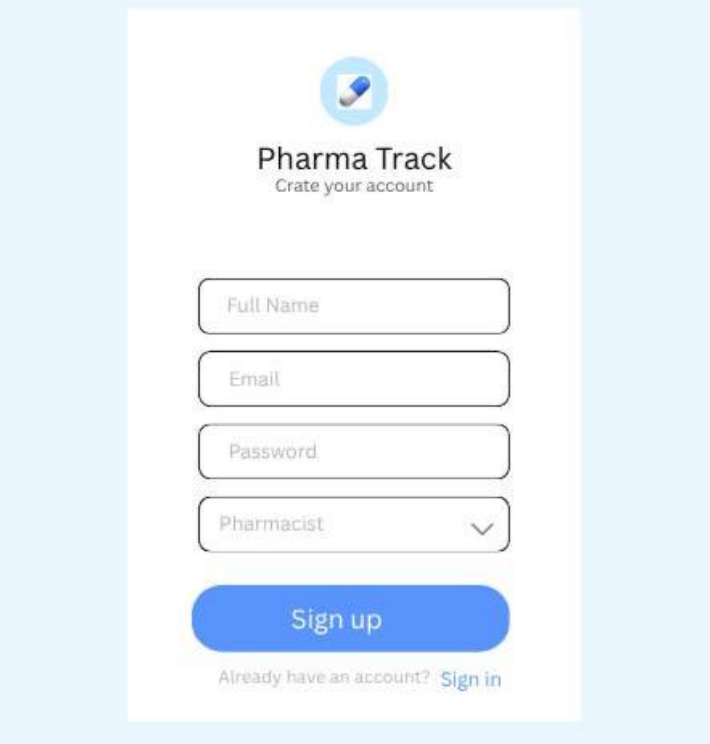


Figure 8: Registration Form Design

Login Form:

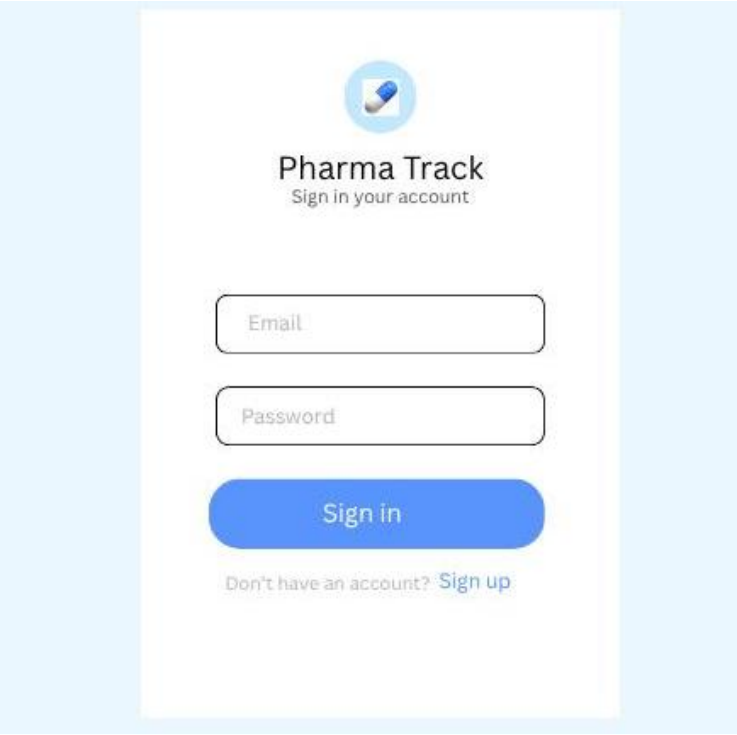


Figure 9: Login Form Design

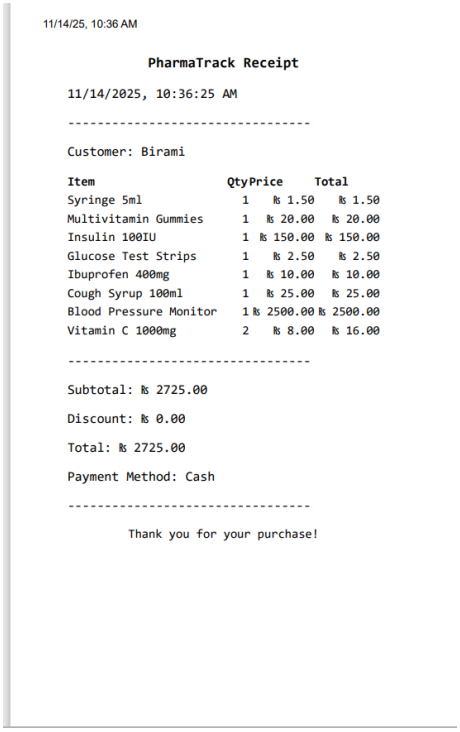
This form allows existing users to access the system using their registered credentials (e.g., email and password).It provides authentication to ensure that only authorized users can log in.If login credentials are invalid, an error message is displayed, prompting the user to retry.

Reports:

Analytics:

Figure 10: Analytics and Reports

Receipts:



11/14/25, 10:36 AM

PharmaTrack Receipt

11/14/2025, 10:36:25 AM

Customer: Birami

Item	Qty	Price	Total
Syringe 5ml	1	₨ 1.50	₨ 1.50
Multivitamin Gummies	1	₨ 20.00	₨ 20.00
Insulin 100IU	1	₨ 150.00	₨ 150.00
Glucose Test Strips	1	₨ 2.50	₨ 2.50
Ibuprofen 400mg	1	₨ 10.00	₨ 10.00
Cough Syrup 100ml	1	₨ 25.00	₨ 25.00
Blood Pressure Monitor	1	₨ 2500.00	₨ 2500.00
Vitamin C 1000mg	2	₨ 8.00	₨ 16.00

Subtotal: ₨ 2725.00

Discount: ₨ 0.00

Total: ₨ 2725.00

Payment Method: Cash

Thank you for your purchase!

Figure 11: Receipt Report

4.1.3. Interface and Dialogue Design

Dashboard UI:

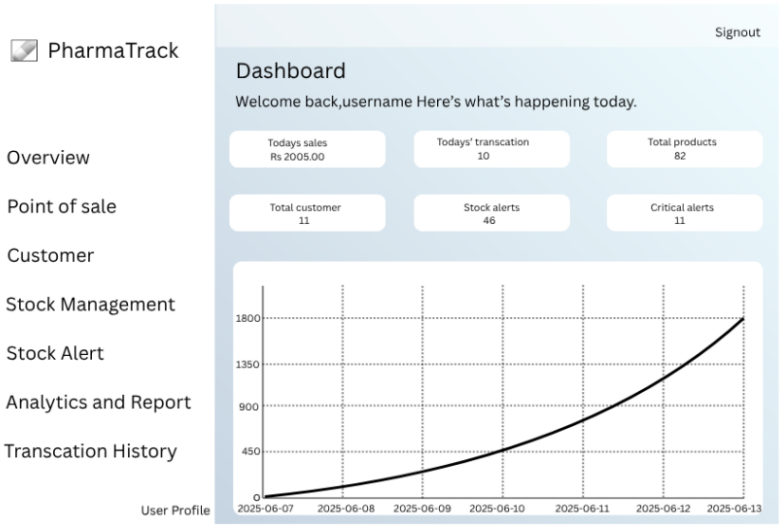


Figure 12: Dashboard UI

POS:

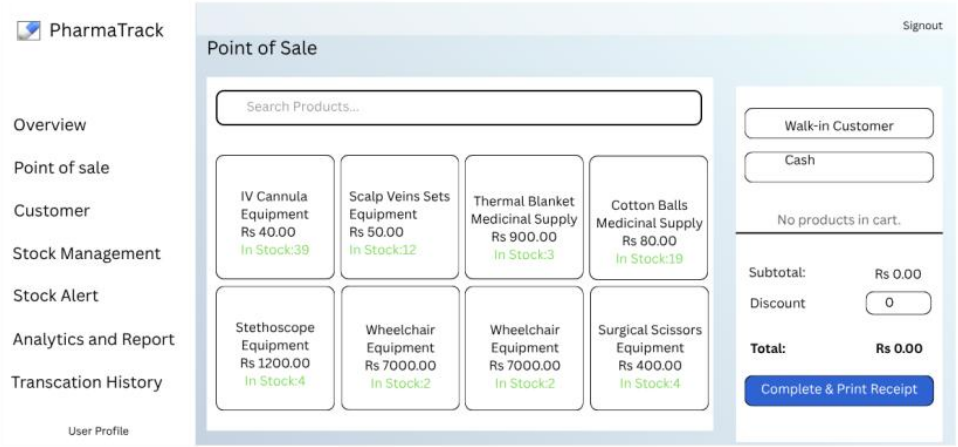


Figure 13: Point of Sale UI

Dialogue Design

Registration Form Dialogue Design:

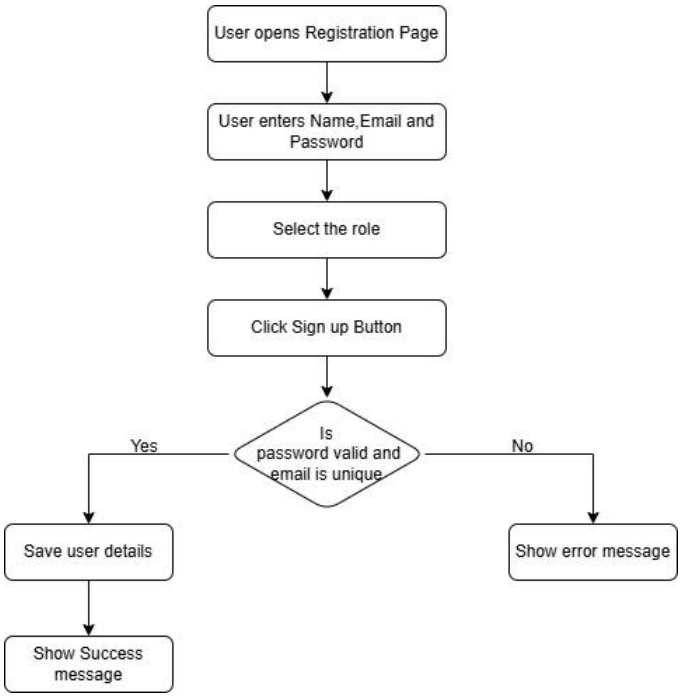


Figure 14: Registration Form Dialogue Design

Login Form Dialogue Design

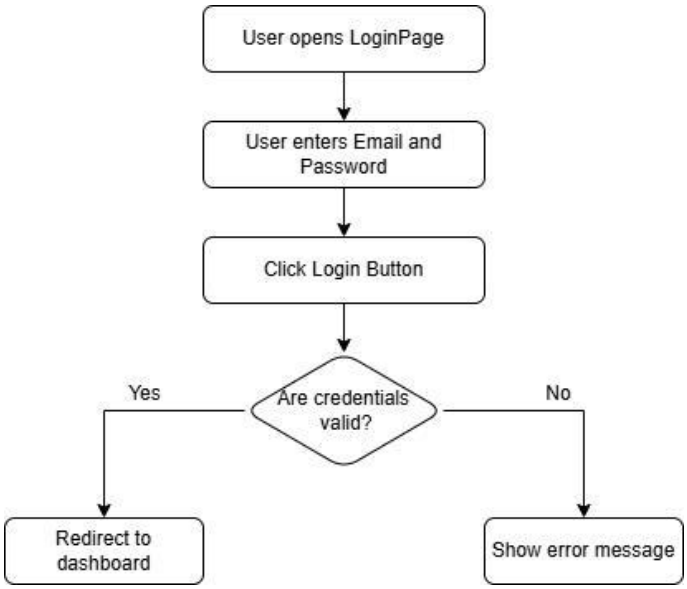


Figure 15: Login Form Dialogue Design

4.2 Algorithm Details

In this project we have implemented several algorithm that has significance impact in our project. These algorithm may be simpler but provides great value to the project. Here are the algorithms implemented in this project:

1. Fuzzy Search Implementation

To enhance the search functionality in PharmaTrack, a **fuzzy search approach** is implemented. This allows users to find products even if the search term partially matches the product name, category, or generic name. The system calculates a similarity score between the search term and each product field using the **Levenshtein Distance** algorithm.

Approach:

- The system considers multiple product fields—**name, category, and generic name**—for matching the search term.
- Initially, only fields whose first three letters match the search input are considered, reducing unnecessary comparisons.
- A **similarity score** is then calculated using a **Levenshtein Distance algorithm**, which measures the number of modifications required to convert the search term to the product field.
- The similarity score is normalized to a value between 0 and 1. For each product, the highest similarity score among its fields is taken.
- Products with positive similarity scores are retained and sorted in descending order, ensuring that the most relevant items appear at the top of the results.

Pseudocode

```
const similarity = (s1, s2) => {

s1 = s1.toLowerCase();

s2 = s2.toLowerCase();

let longer = s1.length > s2.length ? s1 : s2;

let shorter = s1.length > s2.length ? s2 : s1;

if (longer.length === 0) return 1.0;
```

```
const longerLength = longer.length;
const editDistance = (a, b) => {
const dp = Array(a.length + 1)
.fill(null)
.map(() => Array(b.length + 1).fill(0));
for (let i = 0; i <= a.length; i++) dp[i][0] = i;
for (let j = 0; j <= b.length; j++) dp[0][j] = j;
for (let i = 1; i <= a.length; i++) {
for (let j = 1; j <= b.length; j++) {
dp[i][j] =
a[i - 1] === b[j - 1]
? dp[i - 1][j - 1]
: Math.min(dp[i - 1][j - 1], dp[i][j - 1], dp[i - 1][j]) + 1;
}
}
return dp[a.length][b.length];
};
return (longerLength - editDistance(longer, shorter)) / longerLength;
};
```

2. Heuristic-Based Alert Priority Implementation

The Stock Alerts module in the system is designed to monitor and prioritize products based on urgency, using a **heuristic search approach**. Each product is assigned a **priority score** calculated from its stock quantity and expiry date. Low stock contributes to the score, while products that are expired or nearing expiration are assigned higher points. Products are then sorted by priority, ensuring that the most urgent items are highlighted first.

Approach:

- The system evaluates each product based on two main factors: **stock quantity** and **number of days remaining before expiry**.
- A **heuristic priority score** is calculated using predefined rules:

    i. If the stock quantity is **20 units or less**, the product is considered low-stock and receives a base score of **+50**.
    ii. Expired products get the highest score of **+100**, while products expiring within 14 days receive moderate priority with **+80** points.

- Additional heuristic "cost-like" adjustments are applied:

    i. Extra points are added as stock decreases further (lower stock → higher urgency).
    ii. Extra points are added as the expiry date gets closer (fewer days → higher urgency).

- Using this heuristic score, the system estimates which products need immediate attention without evaluating every possible combination or scenario.
- After assigning scores, a **greedy selection** is applied: Products are sorted in descending order of their heuristic priority score. The highest-priority alerts expired, expiring soon, or very low stock—are displayed at the top.
- Only products with a positive priority score are included in the final alert list, ensuring that the interface shows meaningful alerts.

Pseudocode

```
// Heuristic function to calculate priority
const heuristicPriority = (product) => {
const today = new Date();
const expiryDate = new Date(product.expiry_date);
const daysToExpiry = Math.ceil((expiryDate - today) / (1000 * 60 * 60 * 24));
let score = 0;
if (product.stock_quantity <= 20) score += 50;
if (daysToExpiry <= 0) score += 100; // expired
else if (daysToExpiry <= 14) score += 80; // expiring within 2 weeks
score += Math.max(0, 20 - product.stock_quantity); // extra for very low stock
score += Math.max(0, 14 - daysToExpiry); // extra for soon-to-expire
return score;
};
```

Chapter 5: Implementation and Testing

5.1 Implementation

The PharmaTrack system was implemented as a web-based pharmacy management system using React JS for the frontend, PHP for backend logic, and MySQL as the database platform. Development was carried out in a modular manner so that each feature such as product management, stock management, customer handling, sales recording, and analytics could be implemented, tested, and refined individually before full system integration.

5.1.1 Tools used

To develop PharmaTrack following tools, technologies, and libraries were used:

CASE Tools

- **Draw.io** - Used extensively for designing UML diagrams such as Use Case Diagrams, DFDs and ER diagrams. Draw.io provided a simple and efficient interface to create clean, professional system design visuals.

- **Canva** - Utilized for designing graphical components such as UI mockups, presentation visuals, report figures, and image assets required in the documentation. Canva helped maintain consistency and aesthetic quality in non-technical illustrations.

- **MS Word 2021** - Used for preparing the complete project report, formatting chapters, inserting diagrams, organizing tables, and maintaining final documentation as per academic guidelines.

Front End:

- **React.js:** Used for building the user interface and client-side functionalities. It allows component-based development for better modularity and performance.
- **JSON** – Used for API communication between client and server.

Libraries:

- **Tailwind CSS:** Used for styling

Back End:

- **PHP:** PHP is an open source server side scripting language used for creating the server side i.e. backend of web applications. In this project PHP is used for handling all server-side logic and for dynamic implementation of all functionalities such as user authentication and database interactions and all.
- **Apache server:** Apache Web Server is open source Web server creation, deployment and management software designed to create Web servers that have the ability to host one or more http based websites.

Database:

- **MySQL:** MySQL is widely used relational database management system which is used for storing user data. All the database connections and queries is managed via PHP.

Database Name: pharmatrack

Tables:

- **users:** Stores user credentials and basic profile information, such as name, email, password, and role (e.g., pharmacist, admin).
- **customers:** Records customer details, including name, contact information, and purchase history.
- **products:** Contains details of medicines or products available in the pharmacy, including name, category, price, quantity, and expiry date.
- **sales:** Tracks each sale transaction, including the transaction ID, date, customer, and total amount.
- **sale_items:** Records the individual products included in each sale, linking products to specific sales with quantity and price information.

Code Editor:

- **VS Code:** VS Code is a code editor for building and debugging modern web application and cloud application .Used VS Code for editing the project's frontend and backend code.

5.1.2. Implementation Details of Modules

Frontend Module:

The frontend of the PharmaTrack system was developed using **ReactJS**, allowing a fast, component-based, and scalable interface. **Tailwind CSS** was used to design a clean and responsive layout with minimal custom CSS. For icons, **Lucide React** was integrated to make navigation and actions visually clear and user-friendly. Data visualizations such as sales and stock analytics were implemented using **React chart libraries**, providing interactive and real-time insights. Additional tools such as **Axios** for API calls, **React Router** for page navigation, and React hooks for state management helped ensure a smooth and efficient user experience. Overall, the frontend delivers a modern, responsive, and intuitive interface tailored for effective pharmacy management.

B**ackend Module**

1. **User Registration/Login:**

   Used PHP scripts to implement user registration and login with JSON-based input from the React frontend. Input data is sanitized, passwords are hashed using password_hash() before storing in the MySQL `users` table. Login verifies credentials using password_verify() and returns user details in JSON format. CORS headers and OPTIONS request handling are included to enable secure communication between frontend and backend.

2. **CRUD operations on Stock Management:**

   In this module, users can create, read, update and delete the product details through PHP forms. Custom PHP logic handles all those CRUD operations and logic along with interaction with database to perform the operations. We have database name pharmatrack having users, products, customers, sales and sale_items table which stores relevant data.

3. **Transaction Management and Receipt Generation:**

Implemented PHP logic to handle sales transactions, generate receipts, and store all transaction details in the MySQL database for future reference. Each sale captures product details, quantity, total price, date, and customer information. Receipts are generated dynamically in JSON format for frontend display and printing, while transaction history is maintained in a dedicated sales table to support analytics, reporting, and audit purposes.

4. **Analytics & Reports Generation:**

This module provides real-time insights into pharmacy operations, including total sales, revenue, average order value, and inventory summary. React JS fetches JSON data from PHP scripts querying the MySQL database. Key analytics, such as top-selling products and inventory statistics, are computed using SQL queries (SUM, COUNT, AVG, JOIN) and displayed with charts and responsive cards. The dashboard updates automatically as new transactions occur, allowing efficient monitoring of sales trends, inventory levels, and product performance.

Database Module:

Implemented MySQL database to store pharmacy data named pharmatrack, including users, products, sales, sales_items, and customers. PHP scripts handle all CRUD operations, analytics queries, and report generation. A db.php class manages database connections using mysqli, ensuring secure and efficient interaction between the frontend and backend.

5.2 Testing

Testing was performed at multiple levels to ensure that the system behaves as expected across all workflows.

5.2.1. Test Cases for Unit Testing

The developers carry out unit testing on each unit within the designated source code sections. Each unit was fed different test data to check for flaws, which were then debugged.

Test cases for Login/Registration

Test Case ID	Test Case Description	Expected Result	Actual Result	Result	Test Data
001	Register new users with valid credentials.	User should be able to register and get successful message.	User registered successfully.	Pass	Email: sabbubhuju@mail.com Password:#sabu@@1997
002	Register new user with valid email but weak password.	The system must show the weak password message.	Weak password message is shown	Pass	Email: sabbubhuju@mail.com Password:sab
003	Register a new user with already registered email.	Email already used message must be shown.	Already used email message popped up.	Pass	Email: sabbubhuju@mail.com Password: hello@@
004	Verify registration with empty fields.	Display an error message showing the user to fill required fields.	Please fill out all the missing field error.	Pass	Email: Password:
005	Verify that the system displays an error for incorrect email.	Invalid Credentials. Try Again.	Invalid Credentials. Try Again.	Pass	Email: sabbubhuju11@mail.com Password:#sabu@@1997
006	Verify that the system displays an error for incorrect password.	Invalid Credentials. Try Again.	Invalid Credentials. Try Again.	Pass	Email: sabbubhuju11@mail.com Password:#sabu@
007	Verify that the system allows login with correct credentials.	Successful login and directs users to dashboard.	Successful login and directs users to dashboard.	Pass	Email: sabbubhuju@mail.com Password:#sabu@@1997

Table 1: Test cases for Login/Registration

Test case for POS

Test Case ID	Test Case Description	Expected Result	Actual Result	Result
001	Verify selecting the products and adding an them to cart	Selected product should appear in cart with correct price and quantity.	Products are added to cart.	Pass
002	Verify the system prevents adding out-of-stock products to the cart	System should display "Out of Stock" and prevent adding to cart	System displayed warning and did not add product	Pass
003	Verify the system prevents adding expired products to the cart	System should display "Expired Product" and block adding to cart	Products are added to cart.	Fail
004	Verify the system highlights near-expiry products	System should tag the item as "Expiring Soon"	System displayed near-expiry alert	Pass
005	Verify that the user can remove a product from the cart	Item should be removed and total recalculated	Item removed successfully	Pass
006	Verify that the cart total (subtotal, tax, discount) is calculated correctly	System should compute final bill as per calculation rules	Total calculated accurately	Pass
007	Verify that clicking **"Complete Sale"** processes the transaction & generates a printable receipt	Sale completed and receipt must be generated	Sale completed and receipt generated	Pass

Table 2: Test case for POS

These unit test cases confirm the correctness of individual system components and ensure stable module-level functionality.

5.2.2 Test Cases for System Testing

System-Level End-to-End Test Cases

Test Case ID	Test Case Description	Expected Result	Actual Result	Result	Test Data
001	Registration Page	Should show Registration successful! You can now log in	Registration successful! You can now log in	Pass	Email: kripabanskota@gmail.com Password: @kripa!@ba123#
Password:	Verify Login	Should successfully log in and redirects to dashboard	Successfully logs in and redirects to dashboard	Pass	Email: kripabanskota@gmail.com Password: @kripa!@ba123#
003	Verify stock quantity decreases after sale completion	Stock deducted immediately based on quantity sold	Stock reduced correctly	Pass	Paracetamol 500mg (Qty: 2)
004	Verify system updates product status to "Out of Stock" when quantity reaches zero	Status changes to "Out of Stock" and item becomes unavailable.	Status updated successfully	Pass	VitaminC (Stock = 1 → Sale = 1)
005	Verify dashboard updates inventory count in real time after sale	Dashboard must update the inventory details correctly showing today's changes.	Dashboard updated but shows all time changes.	Fail	
006	Verify system generates expired product alert	Expired product should appear in "Expired Items" section	Alert displayed	Pass	Expired Amoxicillin
007	Verify rec-eipt prints accurate item list and totals after sale	Receipt shows correct item names, quantities, totals, and sale ID	Receipt printed accurately	Pass	

008	Verify stock remains unchanged if sale is cancelled	No stock deduction occurs if sale doesn't complete	Stock unchanged	Pass	Syringe removed from cart.
009	Verify that authorized users can perform CRUD operations (Create, Read, Update, Delete) in the Stock Management module	System should allow users to add new stock, view stock details, update stock quantity/price, and delete stock records with proper validation and confirmation	CRUD operations performed successfully	Pass	Add/Update/Delete Paracetamol 500mg
010	Verify that when a receipt is generated after completing a sale, the transaction is saved in the Transaction History module	Receipt details (items, qty, total, date/time, transaction ID) should be saved accurately in the Transaction History section	Transaction recorded successfully in history	Pass	Sale of Paracetamol 500mg (Qty 2), Calcium 500mg (Qty 1)

Table 3: Test Cases for System Testing

These system tests validate the end-to-end behavior of the complete platform.

5.3 Result Analysis

The implemented **PharmaTrack** system performed efficiently and met all core objectives of a modern pharmacy management solution.

The following observations highlight the results:

- The stock alert mechanism accurately detected low-stock, expiring, and expired medicines using a greedy priority calculation, ensuring timely pharmacy decisions.

- The POS billing module generated receipts smoothly and updated sales, inventory, and transaction history in real time.
- The dashboard and analytics section displayed reliable insights such as total sales, revenue, top-selling products, and inventory value using SQL-driven calculations.
- User authentication worked securely with hashed passwords and validated backend processing.
- React-based UI remained responsive and user-friendly, ensuring smooth navigation across all modules.
- Integrated components such as stock management, customer records, sales history, and analytics worked cohesively without errors.

Overall, the system successfully demonstrated a stable, scalable, and intelligent pharmacy management platform capable of supporting secure operations, efficient inventory tracking, and data-driven decision-making.

CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATIONS

6.1 Conclusion

The project titled **PharmaTrack** was developed with the main goal of improving the way pharmacy operations are managed. The system helps to handle important daily activities such as adding and managing products, recording sales, tracking customers, and generating analytical reports. It reduces the amount of manual work that usually happens in a pharmacy and makes the whole process faster and more accurate.

The backend of the system was built using PHP and MySQL, which made it easy to connect and store data securely. The frontend, developed with React, provides a simple and responsive interface for users. With these combined technologies, the system performs smoothly and gives real-time results for stock levels, product expiry, and overall sales. Overall, PharmaTrack has achieved its objective of creating an efficient and user-friendly solution that supports pharmacies in managing inventory and sales more effectively.

6.2 Future Recommendations

Although the system fulfills its intended goals, several enhancements can further improve functionality, accuracy, and user experience. The following recommendations outline possible directions for future development:

1. **Cloud-Based Storage:** Enable cloud integration to support real-time data sharing and centralized management across multiple pharmacy branches.
2. **Barcode/QR Code Scanning:** Add scanning features for faster billing, quick product entry, and accurate stock updates.
3. **AI-Driven Forecasting:** Use machine learning to predict medicine demand, identify expiry risks, and automate restocking decisions.
4. **Advanced Security:** Implement RBAC, multi-factor authentication, and detailed activity logs to improve system security and prevent unauthorized access.
5. **Automated Backup System:** Introduce scheduled local or cloud backups to protect data and ensure business continuity.
6. **Improved Reporting:** Add customizable charts, downloadable reports, and trend analysis for deeper business insights.

References

[1] Shbaily, E. M. I. M. D. Alotaib and G. S. , "Effectiveness of Pharmacy Automation Systems Versus Traditional Systems in Hospital Settings: A Systematic Review," *A Systematic Review*, no. 1, 2025.

[2] A. K. Ali, "Inventory Management in Pharmacy Practice: A Review of Literature," *Archives of pharmacy practice*, vol. 2, no. 2, pp. 151-156, 2011.

[3] A. Almeman, "The digital transformation in pharmacy: embracing online platforms and the cosmeceutical paradigm shift," *Journal of Health, Population and Nutrition*, no. 3, p. 60, 2024.

[4] M. A. Yusiana, "ANALYSIS OF LOGISTICS INVENTORY MANAGEMENT IN PHARMACY SERVICES," *Pedagogic Research-Applied Literacy Journal*, vol. 1, no. 4, pp. 8-21, 2024.

[5] A. Chandel, "Expiry and Stock Alerts: Safeguarding Patients and Cutting Waste in Clinics," no. 8, 2025.

[6] Nasri Nasri, Sentosa and R. Arvin I. Miracelova, "Application of pharmacy management systems and digital marketing: Impact on highly efficiency and increased turnover," *World Journal of Advanced Research and Reviews*, vol. 24, pp. 1961-19691, 2024.

[7] M. Sallam, "Enhancing Hospital Pharmacy Operations Through Lean and Six Sigma Strategies: A Systematic Review," Cureus, 2024.

[8] Nyagah, Aldress Njagi and Milcah Nyaga, "Expired drugs: statistics, factors and mitigation strategies of med-gate pharmaceuticals," *Nyagah, David, Aldress Njagi, and Milcah Nyaga. "Expired drugs: statistics, factors and mitigation strategies of med-gate pharmaceuticals and tan pharmacy up to 31/07/2021." Journal of Advanced Education and Sciences*, vol. 2, no. 8, pp. 9-11, 2022.

[9] Hyder, M. Rashed and Tareq Khan, "Automatic expiry date notification system," *Int. J. Eng. Sci. Invent*, vol. 9, pp. 14-20, 2020.

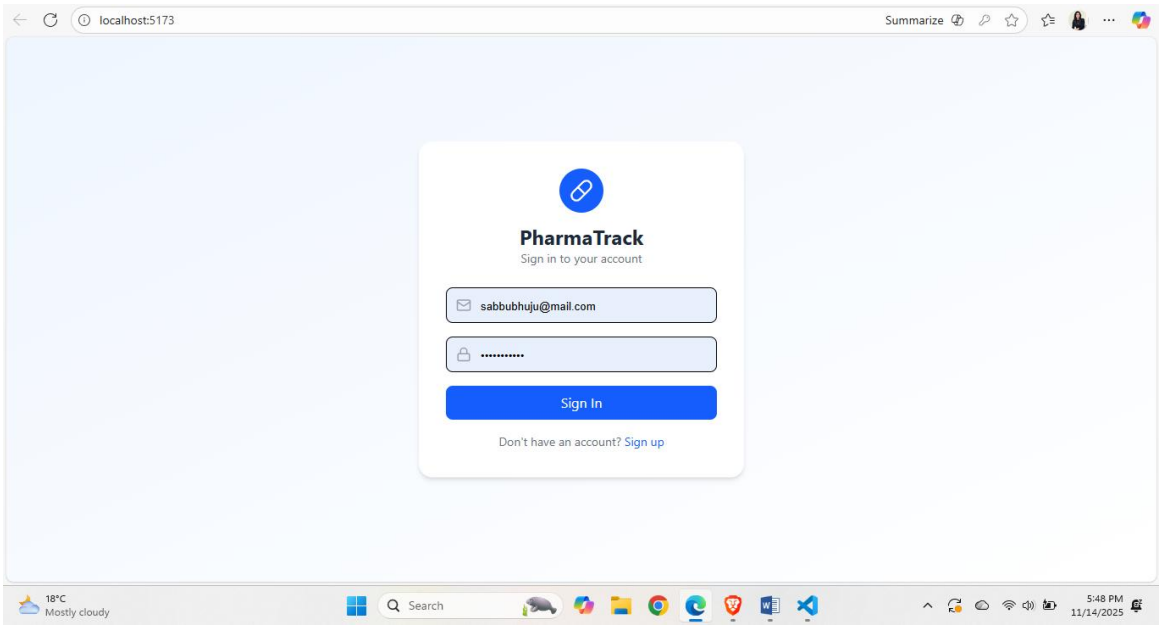
[10] Beck and . J. C. , "Heuristics for scheduling with inventory," *Journal of Scheduling*, vol. 5, no. 6, pp. 43-69, 2002.

[11] Nasution, Mahyuddin KM, Shahrul Azman Mohd Noah and Urip Harahap, "Syst. Rev. Pharm," *Overview of the pharmacy management system in a hospital.*, vol. 11, no. 5, pp. 650-655, 2020.

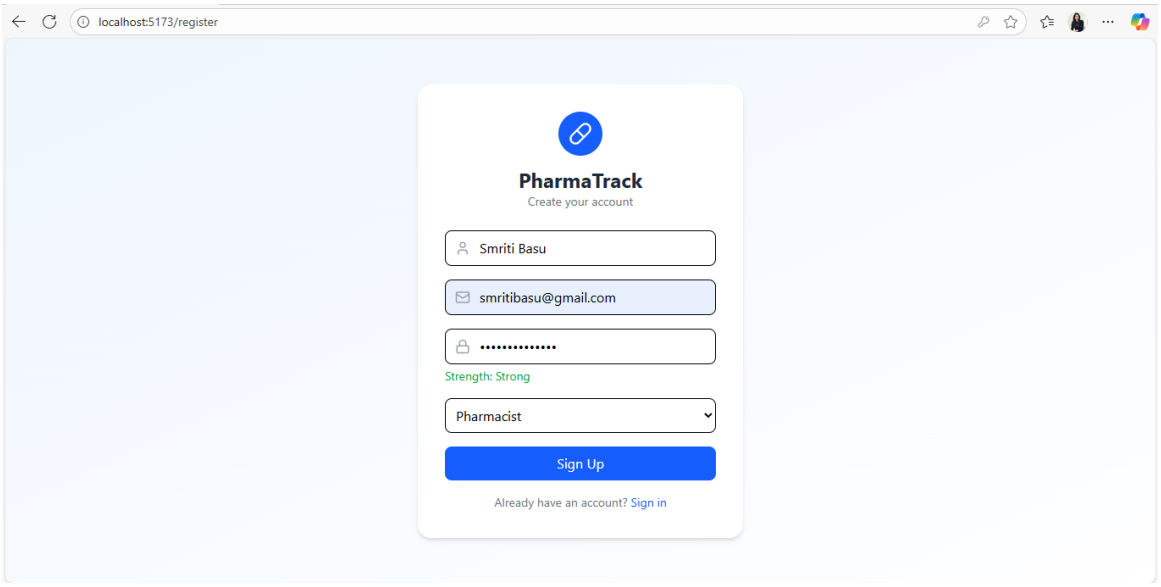
[12] Rustamovna and Abdurakhmanova Umida, "Understanding the levenshtein distance equation," *The American Journal of Engineering and Technology*, vol. 3, no. 7, pp. 134-139, 2021.

APPENDICES

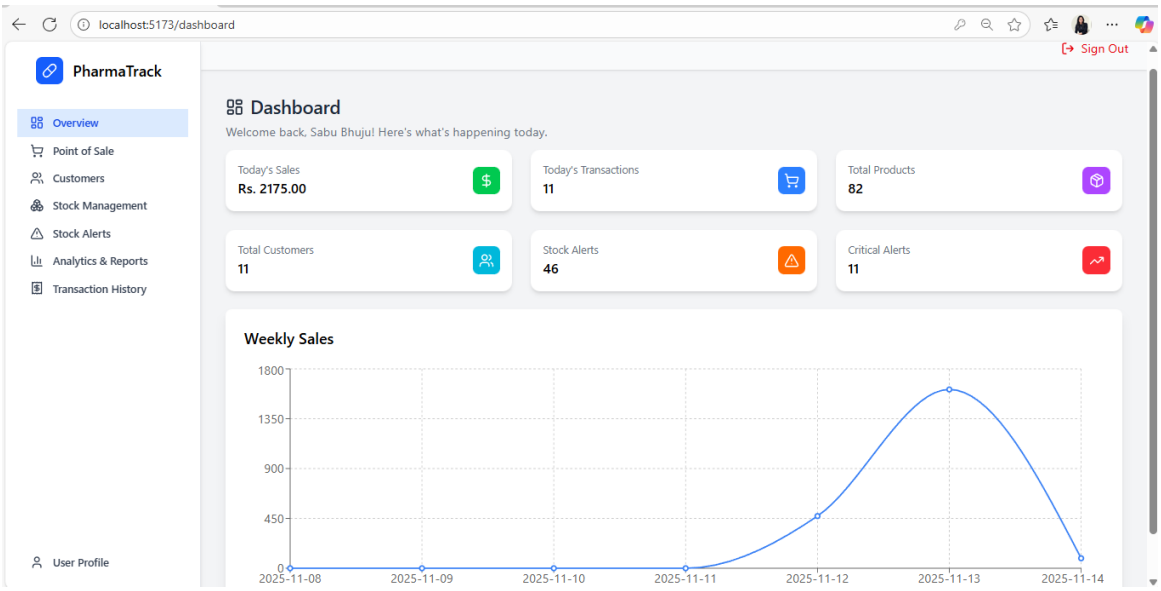
Login Page:



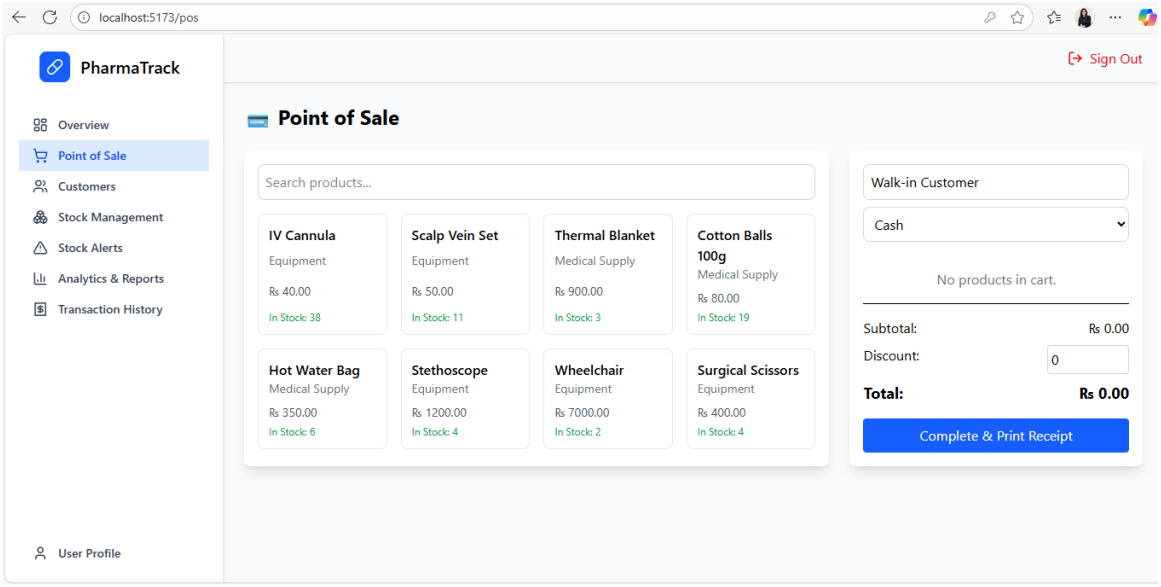
Registration Page:



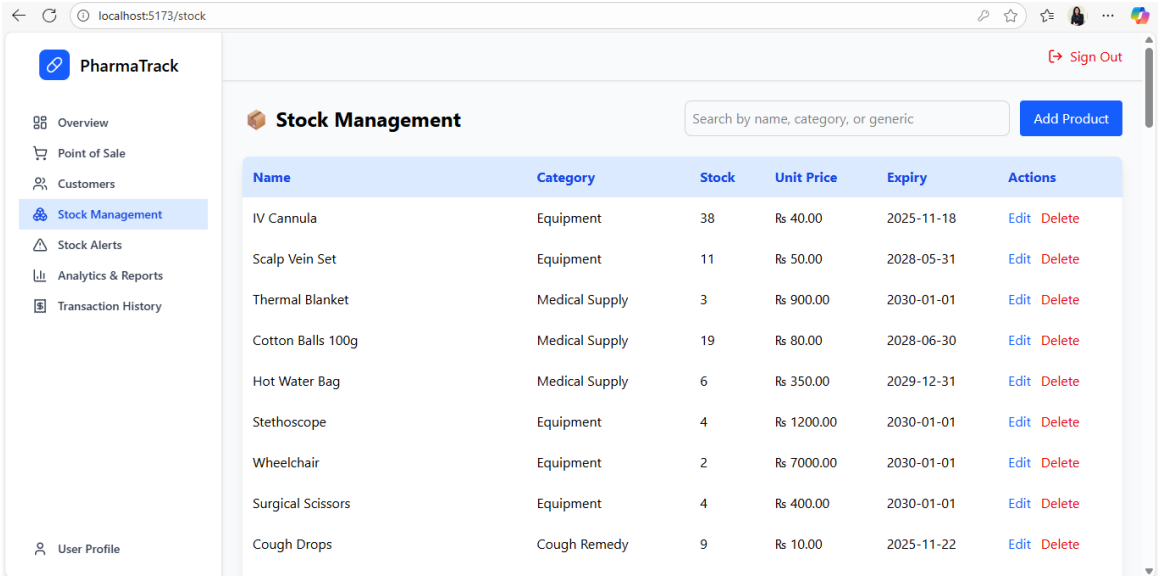
Dashboard:



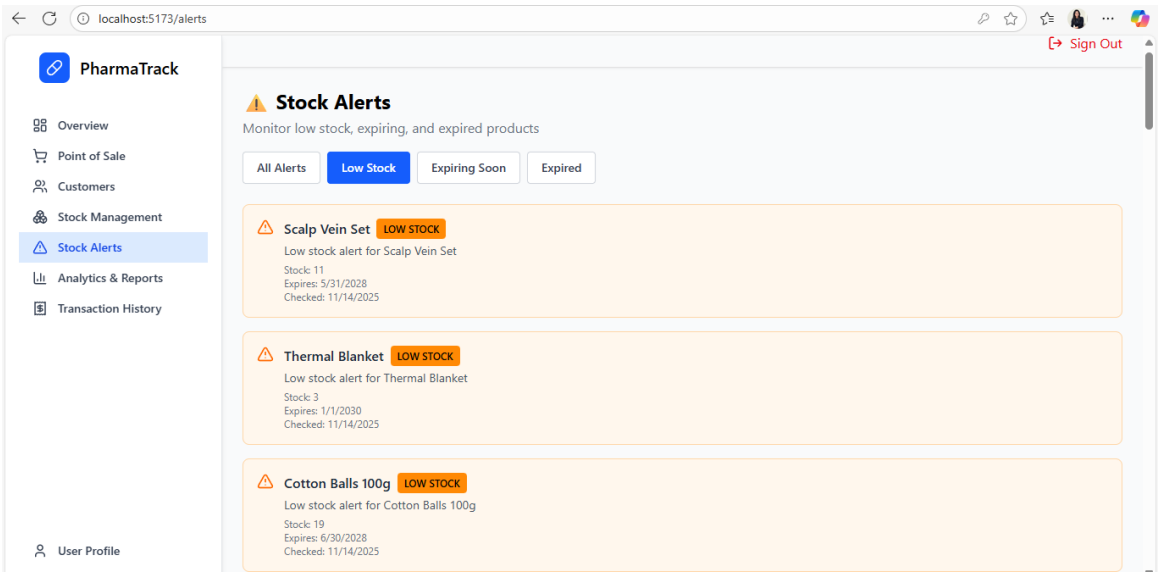
Point of Sale:



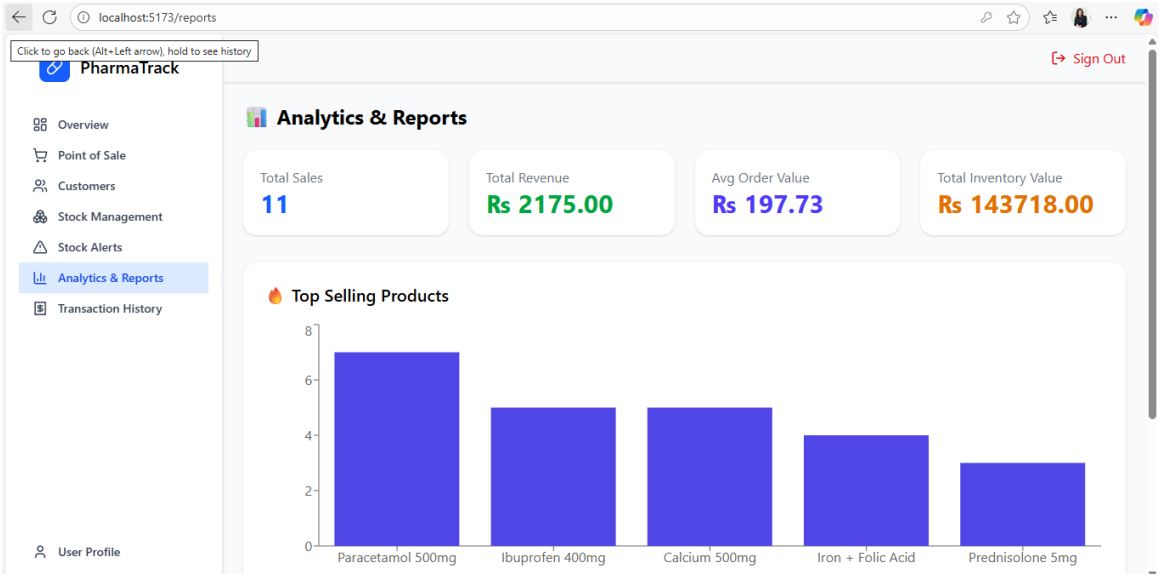
Stock Management:



Stock Alerts:



Analytics & Reports:



Transaction History: