INVENTORY MANAGEMENT SYSTEM FOR MEDICINES

Minor Project-II

(ENSI252)

Submitted in partial fulfilment of the requirement of the degree of

BACHELOR OF TECHNOLOGY

to

K.R Mangalam University

by

Arghydipta Goswami (2301010192)
Puneet Kumar Gupta (2301010171)
Priyanshu Rajput (2301010164)

Under the supervision of

Supervisor Name
Dr. Meenu
Mentor

Department of Computer Science and Engineering

School of Engineering and Technology

K.R Mangalam University, Gurugram- 122001, India

April 2025

CERTIFICATE

This is to certify that the Project Synopsis entitled, "**Inventory Management System for Medicines**" submitted by "**Arghydipta Goswami (2301010192), Puneet Kumar Gupta (2301010171), Priyanshu Rajput (2301010164)**" to K.R Mangalam University, Gurugram, India, is a record of bona fide project work carried out by them under my supervision and guidance and is worthy of consideration for the partial fulfilment of the degree of **Bachelor of Technology in Computer Science and Engineering** of the University.

Type of Project (Tick One Option)

Industry/Research/University Problem

Signature of Internal supervisor

Dr. Meenu

Signature of Project Coordinator

Dr. Vandna Batra

Date: 28th April 2025

INDEX

ABSTRACT

We are all aware of the importance of proper inventory management in pharmacies and medical stores. One of the most essential aspects of an inventory system is its ability to maintain accurate and timely stock information, which ensures medicine availability and improves overall workflow in healthcare operations. An effective system helps manage live and historical stock data that gives pharmacy operators a clear view of inventory status. With this information, staff can make better decisions, respond to shortages, and plan restocking efficiently. Stock-related issues often arise due to factors like manual errors, lack of real-time data, and outdated record-keeping methods. There has been a growing need for digital inventory solutions in small and medium medical businesses, especially in areas with limited resources. While spreadsheets and manual logs have been widely used, they often fall short in preventing overstocking, understocking, or expired medicine circulation. Hence, in this project, we developed a dynamic *Medicine Inventory Management System* using web technologies like HTML, CSS, JavaScript, PHP, and SQLite/Excel. The system allows users to add, view, update, and delete medicine records, track expiry dates, and get alerts for low stock in real time. It includes an intuitive interface and clean structure that supports efficient stock management and is easy to use. With automation and live tracking, pharmacy staff can operate more reliably and ensure better service delivery to their customers.

KEYWORDS: Inventory Management, Pharmacy Software, Medicine Tracking, Real-Time Alerts, PHP, SQLite, Stock Monitoring, Web Application

Chapter 1

Introduction

1. Background of the project

In today's fast-paced and growing urban environment, efficient management of resources has become crucial, especially in sectors related to public health and medicine. Pharmacies and small medical stores, which serve as the backbone of local healthcare access, are increasingly burdened by outdated processes and inefficient systems. Traditional inventory methods—such as manual record-keeping, spreadsheets, and paper logs—fail to support the evolving needs of these establishments. Factors like human error, time-consuming updates, and the inability to monitor real-time stock levels often result in overstocking, expired medicines, and critical shortages of essential drugs. These issues not only cause financial losses but can also negatively impact patient safety and healthcare outcomes.

Moreover, the COVID-19 pandemic further exposed the weaknesses in supply chain and inventory management systems. Sudden lockdowns and unpredictable demand placed immense pressure on pharmacies to ensure the availability of medicines, often with very little data to rely on. Pharmacies, especially smaller establishments without advanced tools, struggled to maintain inventory levels, resulting in missed sales, expired stock, and customer dissatisfaction. The healthcare sector, now more than ever, demands robust digital solutions to manage medical inventory efficiently and reduce dependency on manual methods.

An effective medicine inventory management system can address these challenges by offering streamlined processes, real-time stock monitoring, and automated alerts for low or expired stock. Just like surveillance cameras provide constant visibility and act as a preventive tool for security, a well-designed inventory system functions as a preventive mechanism against inventory errors and stock disruptions. With digital access and an easy-to-use interface, store owners and staff can take quick actions, manage medicine flow, and improve the overall reliability of their operations.

The rise in digitization and the growing affordability of web-based tools have made inventory automation more accessible than ever before. Pharmacies can now leverage technology to gain complete control over their medicine stock, ensuring that essential drugs are always available when needed. A smart inventory system not only simplifies operations but also supports compliance with medical regulations and enhances service quality. In this project, we aim to deliver a scalable and efficient inventory solution using web technologies like HTML, CSS, JavaScript, PHP, and SQLite/Excel. The system enables store owners to add, edit, delete, and monitor medicine records with accuracy and ease.

Ultimately, this project is driven by the need to provide an affordable and practical solution for small to mid-sized pharmacies that cannot invest in expensive enterprise-level software. It empowers them to optimize their resources, eliminate inefficiencies, and ensure that patients always receive the medications they require. Just as investing in CCTV has become essential for safety, investing in a smart inventory system is now a necessity for efficient and safe pharmacy operations.

Table 1. Existing systems

Factors	**Evaluation Criteria**	**System A**	**System B**	**System C**
Ease of Use	User-friendly interface	Very Intuitive	Moderate	Intuitive
Database Management	Types of Databases	SQLite	MYSQL	SQLite
Integration	Compatibility with existing systems	Seamless	Limited	Moderate
Scalability	Ability to scale	Highly Scalable	Limited	Scalable
Inventory Tracking	Tracking stock levels, expiry dates & batch numbers	Yes	Yes	Yes

Data Security	Features of data protection	Comprehensive	Limited	Comprehensive
Cost	Setup Cost	Moderate	High	Moderate
Reliability	System uptime & backup solutions	Very reliable	Reliable	Reliable
Customer Support	Availability of support	Excellent	Good	Moderate

2. MOTIVATION

In recent years, the healthcare sector—particularly pharmacies and small medical stores—has faced increasing challenges in managing medicine inventory efficiently. Factors such as rapid urbanization, population growth, frequent changes in demand, and limited access to digital solutions have made it difficult for small businesses to keep up. Additionally, the COVID-19 pandemic highlighted significant gaps in medicine availability and inventory tracking, resulting in stockouts, overstocking, and wastage of critical drugs. These issues not only lead to financial losses for businesses but also affect the timely delivery of healthcare services to those in need.

Despite the critical role pharmacies play in ensuring public health, many still rely on outdated manual systems for managing their stock. These methods are prone to errors, time-consuming, and lack the ability to provide real-time insights. The need for a reliable, affordable, and easy-to-use inventory management system has become more urgent than ever. By using a digital system that automates stock tracking, generates real-time alerts for low or expired medicines, and maintains accurate records, pharmacy operations can be greatly improved.

The increasing adoption of digital tools in small businesses around the world indicates a shift towards smarter inventory control. A user-friendly inventory system not only reduces the burden on pharmacy staff but also ensures that patients always have access to the medicines they need. With this motivation in mind, the project aims to develop a practical solution that simplifies medicine inventory management, reduces manual workload, and increases overall efficiency through features such as stock alerts, expiry tracking, and intuitive dashboards.

The goal is to empower pharmacy owners and staff with a tool that not only organizes their inventory but also enables them to make informed decisions based on real-time data. A system that is scalable, affordable, and designed specifically for small to mid-sized pharmacies can significantly improve service delivery and ensure that medicine shortages or wastage are minimized. This project is a step toward that goal.

Chapter 2

LITERATURE REVIEW

1. Review of existing literature

DIGITAL INVENTORY SYSTEMS IN PHARMACIES:
Various studies and implementations have explored the advantages of digitizing inventory systems in pharmacies and small-scale healthcare facilities. Traditional methods, such as manual record-keeping and spreadsheet tracking, have shown limitations in accuracy, efficiency, and scalability. Research indicates that digital systems significantly improve stock visibility, reduce wastage, and enhance operational flow. A common feature in most digital inventory systems is real-time stock tracking and alert generation, which helps in maintaining consistent medicine availability. These systems also simplify the process of monitoring expiry dates and reordering stock, reducing human errors in inventory control.

WEB-BASED INVENTORY MANAGEMENT:
Web-based platforms are gaining popularity for their accessibility, ease of updates, and centralized data storage. Unlike standalone desktop systems, web-based inventory systems allow multiple users to access real-time data across different locations. This feature is particularly useful for pharmacy chains or medical distributors. A study on the effectiveness of web-based inventory applications revealed that pharmacies could reduce operational time and restocking delays by over 30% compared to manual systems. These systems typically offer additional features like automated reports, exportable logs, and visual dashboards for better decision-making.

USE OF DATABASE INTEGRATION AND AUTOMATION:
Databases like SQLite and Excel have proven to be highly effective for small-to-medium-sized pharmacies due to their lightweight nature and ease of integration. When paired with scripting languages like PHP or Python, these databases allow dynamic record manipulation and automated stock checks. Systems that integrate database management with alert mechanisms for low or expired stock have shown marked improvements in inventory turnover rates and customer satisfaction. Several case studies highlight how such systems reduce overstocking and understocking issues by ensuring timely alerts and reminders for restocking.

USER EXPERIENCE AND INTERFACE DESIGN:
Research has also emphasized the importance of a clean and intuitive user interface in inventory software. Systems that are overly complex or not user-friendly tend to be underutilized, especially by pharmacy staff without technical backgrounds. Interfaces built using front-end technologies like HTML, CSS, and JavaScript, when paired with responsive frameworks like Bootstrap, have been found to significantly improve user adoption and efficiency. Systems with role-based access and visual

indicators (e.g., color-coded alerts for low stock or expiry) are particularly effective in enhancing the usability and speed of operations.

CLOUD VERSUS LOCAL STORAGE APPROACHES:
In terms of data storage, both cloud-based and on-premise systems have their pros and cons. Cloud-based solutions provide real-time synchronization and access from multiple devices but may raise concerns regarding data privacy and require a reliable internet connection. On the other hand, local storage systems offer more control and are suitable for pharmacies operating in low-connectivity areas. Recent developments in hybrid models allow syncing local databases with the cloud during specific intervals, offering the benefits of both methods.

This review of the existing systems and literature reflects the need for a cost-effective, user-friendly, and feature-rich medicine inventory management system tailored to small and medium pharmacies. The current project builds upon these findings to develop a reliable and efficient solution that supports real-time stock management, expiration alerts, and user-friendly controls.

Table 2. LITERATURE REVIEW/COMPARITIVE WORK

Project Title	Objectives	Technologies Used	Outcomes
Medicine Inventory Management System	Manage and track medicine stock efficiently	RFID technology, Database Management Systems (SQL), Cloud storage	Improved stock tracking, reduced stockouts, and waste
Stock Monitoring System in Pharmacy A	Ensure timely replenishment of medicines	Barcode scanning, IoT sensors, Cloud-based inventory tracking	Increased order accuracy, optimized restocking process
Automated Inventory System for Medicines	Automate stock count and expiry tracking	AI-powered stock prediction, Real-time tracking	Reduced human error, optimized shelf-life management

Medicine Supply Chain Management	Enhance supply chain visibility and streamline procurement	Blockchain technology, GPS tracking, Database management	Improved traceability, faster procurement processes
Pharmacy Management System	Track sales and manage customer orders	POS system, Data analytics, Cloud storage	Higher customer satisfaction, better inventory control
Medicine Expiry Monitoring System	Prevent distribution of expired medicines	RFID, Real-time alerts, Cloud storage	Reduced distribution of expired medicines, improved safety

2. GAP ANALYSIS

While several inventory management systems have been developed for large-scale hospitals and retail chains, there remains a significant gap in tools designed specifically for small and medium-sized pharmacies. Most existing solutions are either too complex, expensive, or feature-heavy for small stores that require only essential functionalities. These systems often overlook the practical needs of local pharmacies, such as real-time stock alerts, expiry tracking, and ease of use for staff with minimal technical expertise. Furthermore, many available tools lack scalability, offline support, or seamless integration with simple databases like Excel or SQLite, which are more feasible for smaller operations.

Our project aims to bridge this gap by offering a simplified yet powerful web-based system that caters specifically to the needs of local pharmacies. It incorporates all the core features—such as medicine tracking, expiry alerts, low-stock notifications, and automated record updates—into a clean and easy-to-use interface. Unlike more generalized or enterprise-level software, this system is designed to be lightweight, cost-effective, and scalable. By focusing on user experience, affordability, and practicality, our system brings all the necessary tools under one platform, making it highly accessible to pharmacies that want to modernize their operations without overcomplicating them.

3. PROBLEM STATEMENT

Many pharmacies and small medical stores continue to rely on outdated manual methods for tracking their medicine inventory. While these methods may have worked in the past, they now pose significant challenges in ensuring accurate stock levels, monitoring expiration dates, and managing reordering processes. Manual systems are prone to human error, time-consuming updates, and a lack of real-time data, which can lead to critical issues such as overstocking, expired medicines, or shortages of essential drugs.

Existing digital inventory solutions either fall short in addressing the specific needs of small pharmacies or are too complex and expensive to implement. As a result, store owners face difficulty maintaining operational efficiency and ensuring that patients have timely access to the medicines they need. This creates an urgent need for a smarter, user-friendly, and affordable inventory management system that automates essential tasks like stock updates, low-inventory alerts, and expiry tracking. A system with such capabilities would significantly improve inventory accuracy, reduce waste, save time, and enhance the overall service quality in the pharmaceutical sector.

4. OBJECTIVES

The main objective of this project is to develop a smart, user-friendly, and affordable Medicine Inventory Management System that helps pharmacies and medical stores manage stock efficiently, minimize errors, and ensure timely availability of essential medicines.

Key objectives include:

1. **Real-Time Stock Monitoring** – Keep track of available medicine stock with real-time updates to avoid overstocking or stockouts.
2. **Expiry Tracking** – Automatically detect and alert users when medicines are nearing or past their expiration dates.
3. **Low Stock Alerts** – Notify users when stock levels fall below a set threshold to ensure timely restocking.

4. **Batch Tracking and Management** – Maintain detailed records for each batch, including manufacturing and expiry dates, for better traceability.
5. **User-Friendly Interface** – Create a clean and intuitive web-based interface using technologies like HTML, CSS, JavaScript, PHP, and SQLite/Excel, so that even non-technical users can manage inventory easily.
6. **Supplier Integration** – Facilitate the tracking of supplier details for reordering and managing procurement history.
7. **Data Accuracy and Efficiency** – Minimize manual workload, reduce errors, and speed up the inventory process for better operational efficiency.

The system aims to bridge the gap between manual stock management and complex enterprise systems by offering a practical and scalable solution. It empowers pharmacy staff with reliable tools to maintain inventory accurately and improve healthcare service delivery.

CHAPTER 3

1. METHODOLOGY

The project is divided into various phases, from requirement analysis to deployment. Each phase has been carefully designed to ensure that the system is efficient, scalable, and easy to maintain.

Technologies and Tools Used:

- **HTML:** Used to create the basic structure and layout of the web application.
- **CSS:** Applied to style the user interface and enhance the visual presentation.
- **JavaScript:** Implemented to add interactivity, form validation, and dynamic updates on the client side.
- **PHP:** Used as the server-side scripting language to handle business logic, server communication, and interaction with the database.
- **SQL (PHPMyAdmin):** Used for creating and managing database tables and records to store medicine details securely.

Platforms Used:

- **XAMPP:**
  A free and open-source cross-platform web server solution package that includes Apache, MySQL, PHP, and Perl. XAMPP is used to host the web application locally and manage the backend server.
- **VS Code:**
  Visual Studio Code is used as the main Integrated Development Environment (IDE) for writing and managing code files for HTML, CSS, JavaScript, and PHP.
- **PHPMyAdmin:**
  A web-based database management tool used to create and manage the SQL database easily through a graphical interface. It is used to design and organize tables related to medicines, stock quantities, expiry dates, suppliers, etc.

Development Process:

1. **Requirement Analysis:**
   Understanding the essential functionalities like adding medicines, updating stock, monitoring expiry dates, and generating stock alerts.
2. **System Design:**
   Planning the database schema, web application structure, and user interface design using wireframes and entity-relationship diagrams.
3. **Implementation:**
   Developing frontend interfaces (HTML, CSS, JavaScript), backend functionality (PHP scripts), and the database (SQL via PHPMyAdmin).
4. **Integration:**
   Connecting the frontend forms and dashboards with backend PHP scripts and the database using server-side programming and AJAX where necessary.
5. **Testing:**
   Checking each module individually (unit testing) and then as a whole (integration testing) to ensure that all features work properly across different platforms (Windows, Linux).
6. **Deployment:**
   Hosting the complete project on a local server using XAMPP for testing. Preparing it for future deployment on a cloud server if required.
7. **Maintenance:**
   Ensuring regular updates, database backups, and bug fixes to maintain smooth operation.

2. Overall architecture /Flow chart:

User Interface (UI) Layer:

- **Components:** Dashboard, Inventory Overview, Medicine Search, Reports, Alerts.
- **Interaction:** This is the front-end interface used by hospital/pharmacy staff to interact with the system.
- **Tools:** HTML, CSS, JavaScript (Vue.js/React for dynamic features).

2. Application Layer (Backend):

- **Components:**
  - **API Server (PHP or Node.js):** Handles all the business logic and processing of requests from the UI layer.
  - **Inventory Management:** Responsible for tracking stock levels, entering new medicine details, managing expiry dates, and processing orders.
  - **Expiry Alerts & Notifications:** Module for checking expiry dates and sending alerts when stock is about to expire or when the stock is low.
  - **Reports Generation:** Generates and provides reports about inventory health, sales, and stock alerts.
- **Interaction:** The backend receives requests from the front-end, processes them, interacts with the database, and returns responses (e.g., updated inventory status, reports, etc.).

3. Data Layer:

- **Components:**
  - **Database (SQLite/Excel):** Stores all data related to medicines, stock levels, expiry dates, order history, and system logs.
- **Interaction:** The application layer interacts with the database to perform CRUD (Create, Read, Update, Delete) operations on data. The database sends queries to retrieve information like stock status, expiry dates, etc.

4. Security and User Authentication:

- **Components:** User Login, Role-based Access Control, Data Encryption.
- **Interaction:** Ensures that only authorized personnel have access to the system, based on their roles. Encryption ensures that sensitive data is secure.

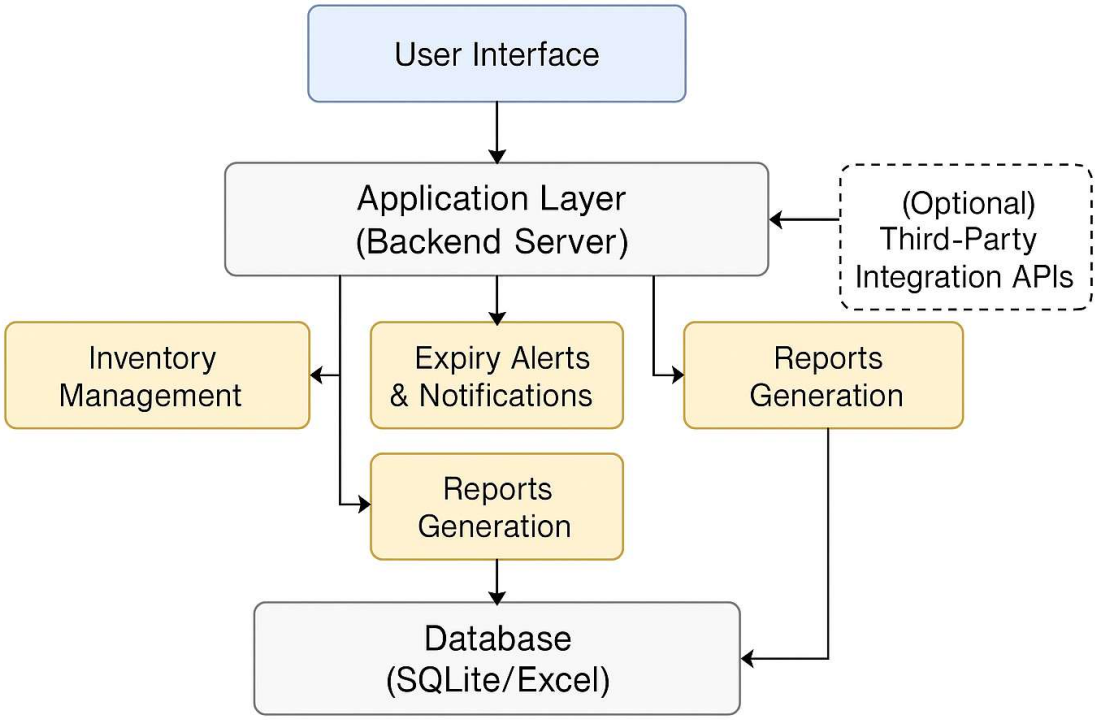


Figure 1. Architecture diagram

3. Data Description

- **Data Source:** The data used in this system is collected from local inventory records maintained by the pharmacy or medical suppliers. This data includes medicine details such as names, batch numbers, quantities, expiry dates, and supplier contact information. The source is primarily internal, with manual entries made by pharmacy staff or suppliers.
- **Data Collection Process:** Data is collected through a web-based interface where users manually input medicine details. The system allows users to add, update, and manage medicine records. PHP scripts process the data and store it in an SQLite database for easy retrieval and management. Data entry is done using forms, and validation checks are in place to ensure correctness.
- **Data Type:** The data in the system is a combination of textual (e.g., medicine names, supplier names) and numerical data (e.g., quantities, batch numbers). The expiry date

is a date-type field, ensuring proper management of time-sensitive data. This allows for detailed tracking of medicines and their respective statuses.

- **Data Size:** The dataset size depends on the number of medicine records entered into the system. It typically includes hundreds or thousands of entries, depending on the size of the inventory. The table structure has approximately 7 fields (variables), with each record representing one medicine entry.
- **Data Format:** The data is stored in an SQLite database in tabular format, structured into fields like medicine name, batch number, quantity, expiry date, and supplier details. Each entry is indexed by a unique ID for easy reference and search.
- **Data Preprocessing:** Data preprocessing includes basic validation to ensure that no fields are left empty, quantities are positive integers, and expiry dates are in proper date format. For consistency, all text fields (e.g., medicine names, supplier names) are checked for valid input formats (e.g., no special characters). Any incorrect or incomplete data is flagged for manual correction.
- **Data Sampling:** Since this is a full inventory system, data is continuously updated and not sampled. The system uses the entire dataset for tracking and reporting purposes. No specific sampling strategy is applied since data is always considered in its entirety.
- **Data Quality Assurance:** To ensure data integrity, the system includes validation checks on input fields, ensuring that no invalid data is entered. Users are prompted to correct any inconsistencies, and a backup mechanism is in place to recover data in case of system failures. Regular database maintenance is also performed to check for any errors in data.
- **Data Variables:**
  - id: Unique identifier for each medicine record (integer).
  - medicine_name: Name of the medicine (text).
  - batch_number: The batch number for tracking the specific batch of medicine (text).
  - quantity: Quantity available in stock (integer).
  - expiry_date: The expiry date of the medicine (date).
  - supplier_name: Name of the supplier providing the medicine (text).

- supplier_contact: Contact details for the supplier (text).

- **Data Distribution and Summary Statistics:** The distribution of medicine quantities typically shows a high number of medicines with low quantities, while fewer medicines have higher quantities. Expiry dates are spread out, with a concentration of medicines nearing expiration. Common summary statistics include:
  - **Mean quantity:** Average quantity across records.
  - **Median quantity:** Midpoint quantity for balancing skewed data.
  - **Standard deviation:** Variation in quantities across medicines.
  - **Expiry date:** Range from near-expiration to far-expiration medicines.

  Visualizations like histograms or pie charts of expiry dates and quantities could be used for more in-depth analysis.

4. Procedure /Development Life Cycle

1. Requirements Gathering:

- **Objective:** Understand the needs of the hospital/pharmacy regarding inventory management, including tracking stock levels, expiry dates, and generating reports.
- **Execution:**
  - Conducted interviews with hospital staff and pharmacists to define functional and non-functional requirements.
  - Analyzed existing manual processes and identified pain points such as overstocking, expired medicines, and misplaced entries.

2. System Design:

- Objective: Design the architecture and user interface of the system.
- Execution:
  - Created high-level system architecture, outlining interactions between the frontend, backend, database, and third-party integrations.
  - Developed wireframes for the user interface to ensure intuitive navigation and usability.
  - Designed database schema for storing medicine data, stock levels, and expiry dates.

3. Development Phase:

- Objective: Develop the system's core functionality and features.
- Execution:
  - Frontend Development: Built interactive dashboards, inventory management tools, and expiry alerts using HTML, CSS, and JavaScript.
  - Backend Development: Implemented server-side logic using PHP to handle medicine inventory, stock updates, expiry checks, and report generation.
  - Database Development: Created an SQLite database to store and manage inventory records, including medicine details, stock status, and expiration dates.

4. Testing:

- Objective: Ensure that the system works as expected and meets user requirements.
- Execution:
  - Unit Testing: Tested individual functions like stock updates, expiry checks, and alert notifications.
  - Integration Testing: Ensured smooth communication between the frontend, backend, and database, verifying data consistency.
  - System Testing: Conducted full system testing, simulating real-world scenarios such as stock depletion and expiration alerts.

5. Deployment:

- Objective: Deploy the system for live use in hospitals or pharmacies.
- Execution:
  - Deployed the system on a local server (XAMPP) for initial testing and debugging.
  - Moved the application to a production server for actual deployment.
  - Set up backup systems, and ensured that data security protocols (encryption, role-based access) were implemented.

6. Maintenance & Updates:

- Objective: Ensure ongoing system performance and address issues post-deployment.
- Execution:
  - Monitored system performance and user feedback to identify areas for improvement.
  - Regularly updated the system to fix bugs, implement new features (e.g., supplier integration), and improve functionality.

Details of tools, software, and equipment utilized.

PLATFORM USED

For this project, we have utilized various modern technologies that have been carefully selected based on the project's needs. This chapter explains the main technologies and platforms used, along with reasons for choosing them.

PROGRAMMING LANGUAGES AND TECHNOLOGIES: PHP, HTML, CSS, JavaScript

We have chosen PHP as the server-side scripting language for backend development because it is highly reliable, widely supported, and perfect for creating dynamic web applications. Combined with HTML, CSS, and JavaScript for the frontend, these technologies provide a seamless, responsive, and interactive user experience.

PHP is a popular open-source general-purpose scripting language that is especially suited for web development. It was created in 1994 by Rasmus Lerdorf and has evolved significantly over the years. Its easy integration with HTML and databases like SQLite and MySQL makes it ideal for building lightweight, efficient inventory management systems.

Reasons for Selecting These Technologies:

PHP:

- Open-source and free to use.
- Easy integration with database systems.
- Strong community support and frequent updates.
- Simple to learn and deploy on various servers (like XAMPP).

HTML/CSS:

- Essential for structuring and designing web pages.
- CSS ensures responsive design and improves the visual layout.
- JavaScript:
- Provides dynamic features on the client side.
- Enables real-time interactivity, like instant validation and dashboard updates.

SQLite/Excel:

- Lightweight and fast database engine (SQLite) ideal for small to medium-sized systems.
- Excel support for exporting and viewing reports offline.

Some Specific Features of PHP and Web Technologies Used:

- An interpreted language, executed at the server-side before rendering the final output to the client.
- Open-source, freely available for building commercial or private software.
- Multi-platform compatibility: runs on Windows, Linux, Mac OS.
- Easy integration with databases and front-end technologies.
- Excellent for session management, file handling, and security controls.
- Object-Oriented Programming support in PHP 7+ versions.

Main Features Implemented in the Project:

- Real-time Stock Monitoring – Keeps track of medicine availability and updates stock levels automatically.
- Expiry Date Tracking – Alerts pharmacy staff before medicines expire.
- Low Stock Detection – Detects and alerts when stock reaches a minimum threshold.
- User-Friendly Management – Simple and efficient interface to add, update, delete, or view medicine records.

Each feature is discussed in detail in the Methodology section, where the workflows, process models, and design strategies used for the project are explained.

ENVIRONMENTAL SETUP

SOFTWARE REQUIREMENTS

Below are the necessary software requirements to successfully install and run the *Medicine Inventory Management System*:

1. **Operating System:**
    - Windows / Linux / Mac OS (any modern version) — the system is cross-platform and can run on any of these operating systems.
2. **Local Server Environment:**
    - **XAMPP** (or any similar server stack that supports PHP and SQLite) — required to host the PHP files and manage database operations locally.
3. **Web Browser:**

- Any modern browser (such as Google Chrome, Mozilla Firefox, or Microsoft Edge) to access the web-based application interface.

4. **Programming Languages and Tools Installed:**

   - **PHP** (version 7.0 or above) — for server-side scripting.
   - **SQLite** — lightweight database for storing inventory records.
   - **HTML, CSS, JavaScript** — used for frontend development.

5. **Optional Tools:**

   - **Text Editor/IDE** — like VS Code, Sublime Text, or Atom for editing and maintaining project files.
   - **SQLite Browser** — for easier database management and visualization.

HARDWARE REQUIREMENTS

In terms of hardware requirements there is not much required at all but still below requirements are must:

1. Working PC or Laptop
2. Stable Internet Connection
3. External Storage/Backup Device

PLATFORMS ALREADY TESTED ON:

- It is tested on Linux Mint, Linux Ubuntu, Windows 7 and Windows 10.
- The system was also tested on various modern web browsers like Google Chrome, Mozilla Firefox, Microsoft Edge, Opera Browser

Chapter 4

IMPLIMENTATION

Detailed Explanation of How the Project Was Implemented:

The project started with selecting a real-world topic. A structured folder system was created (config, CSS, JS, PHP). HTML and CSS were used for the frontend design, JavaScript for validations and interactivity, and PHP for backend operations. MySQL/SQLite was used to build and connect the database. The project was tested on a local server, and necessary fixes were applied before final submission.

Description of Algorithms, Code Snippets, or Design Diagrams

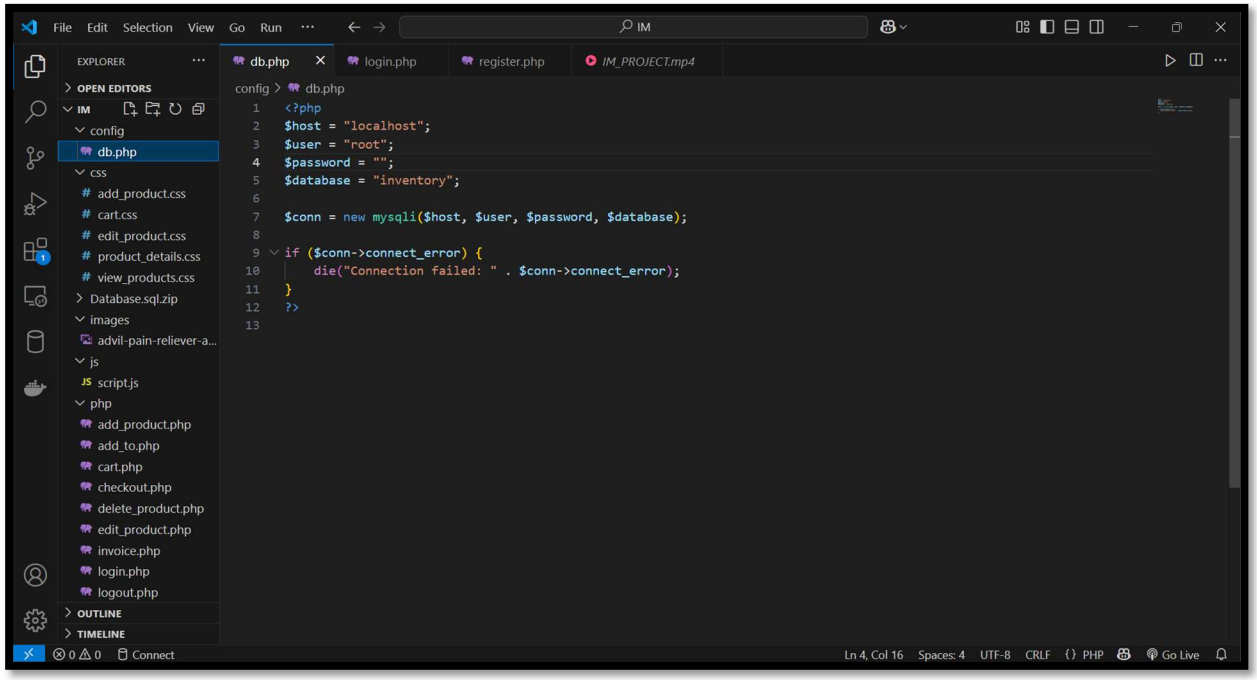
Simple algorithms were used, such as form validation, user authentication, and data management (add, update, delete). Code snippets included PHP database connections, JavaScript validations, and basic SQL queries. A simple system flow was followed: User → Frontend → Backend (PHP) → Database.

Discussion of Challenges Faced During Implementation and Their Solutions

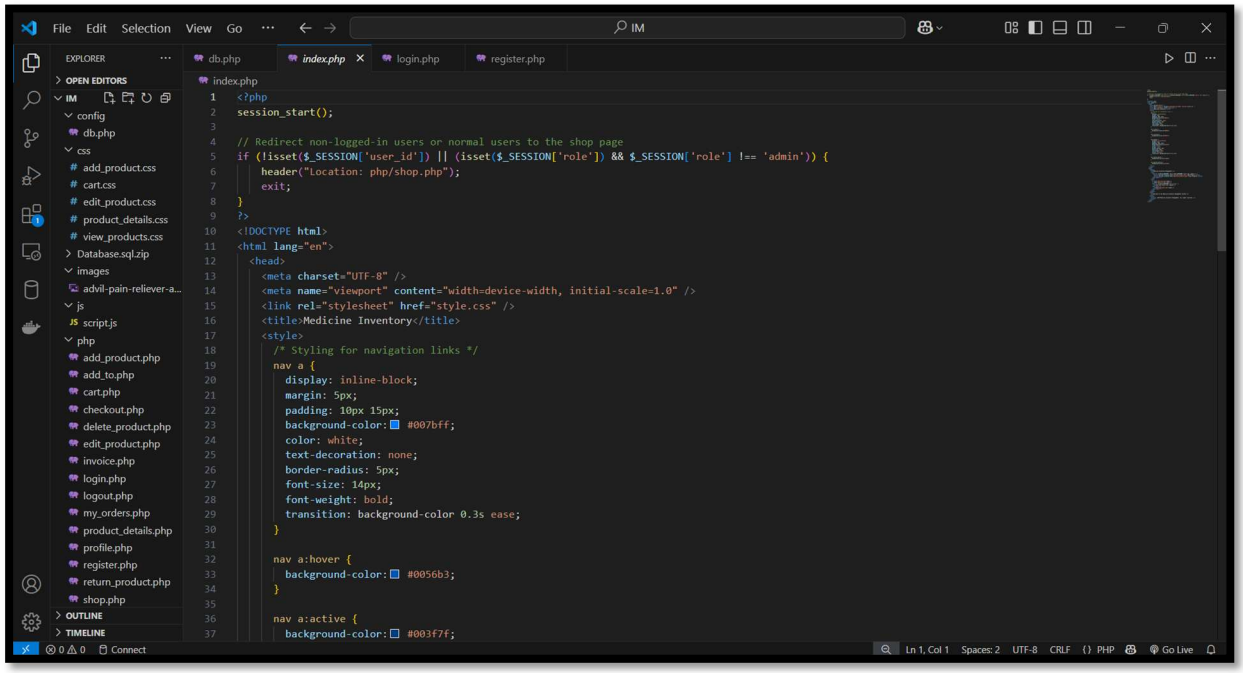
Common challenges included CSS/JS file linking issues (solved by correcting paths), database connection failures (fixed by verifying credentials), and page reload problems after form submission (solved using JavaScript or AJAX methods).

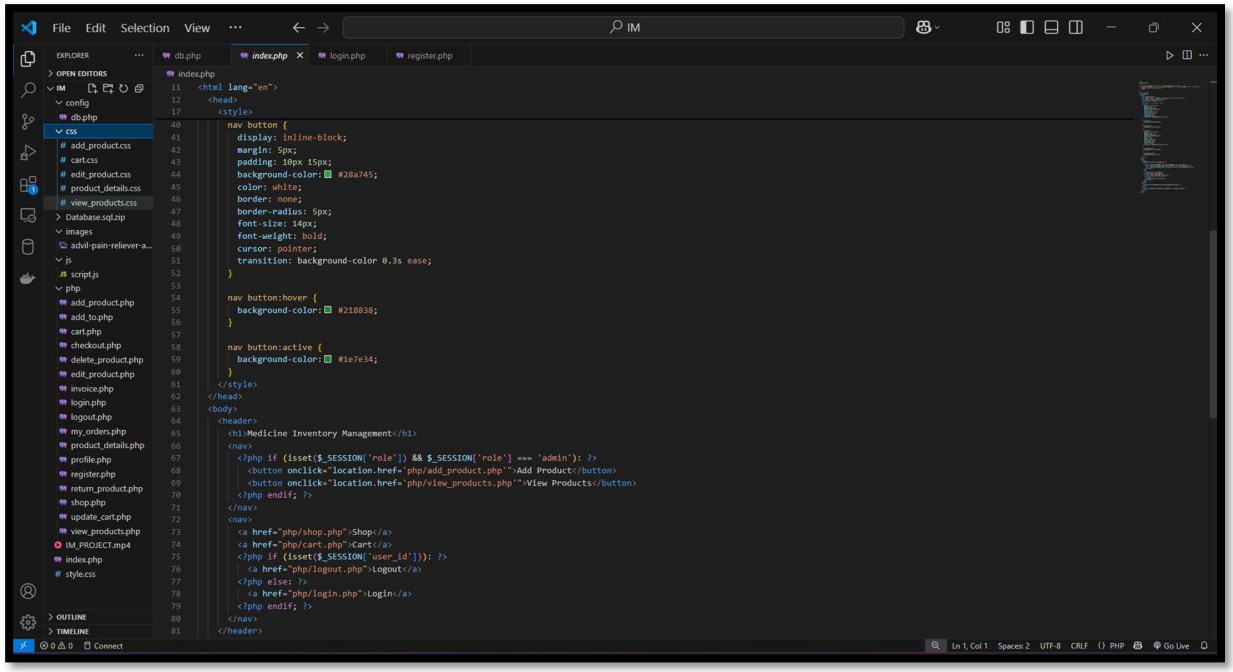
CODE SNIPPETS

DATABASE CONNECTION:

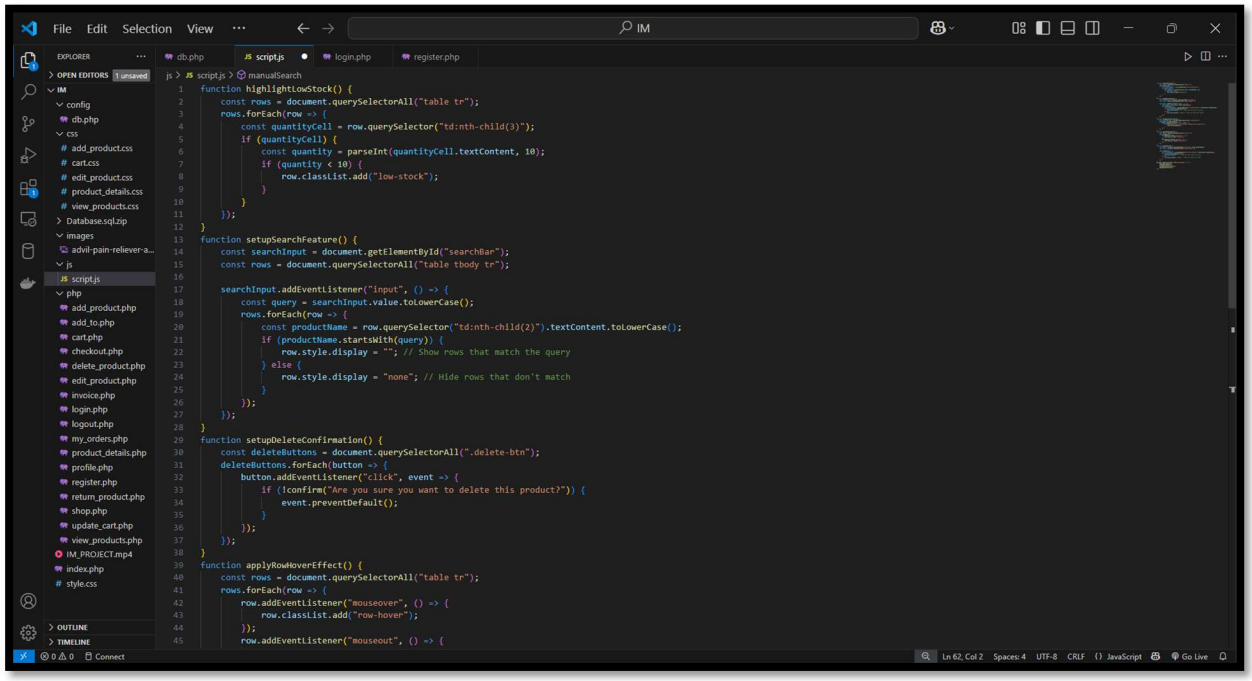


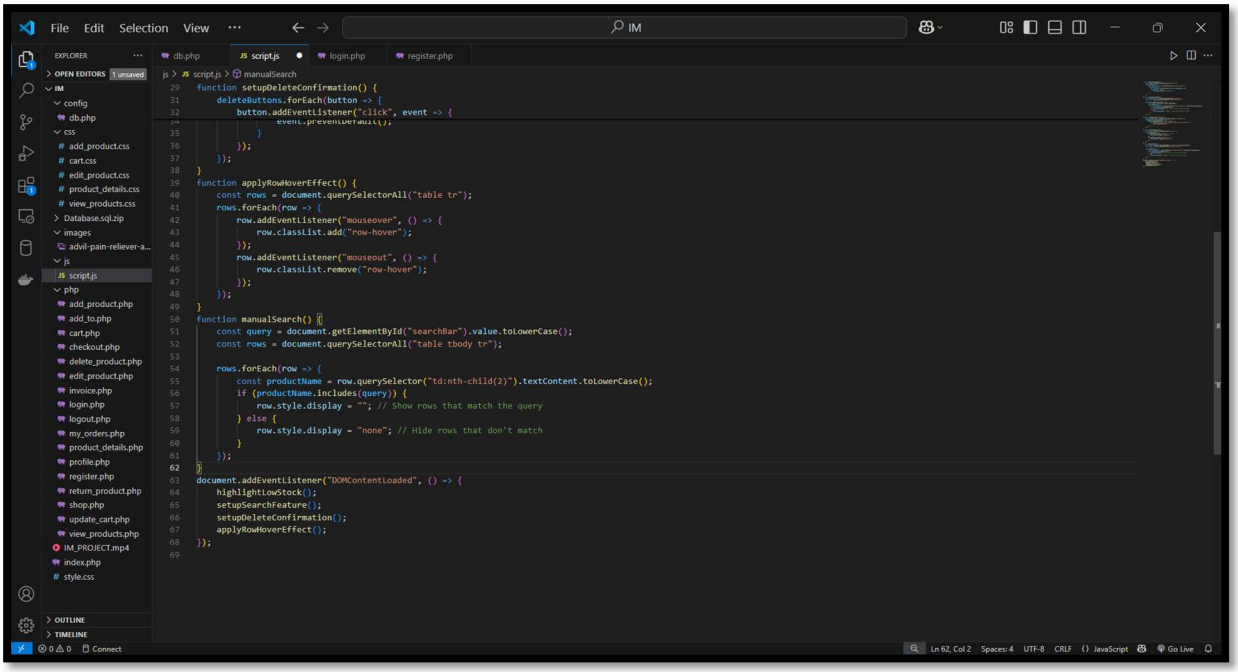
INDEX (MAIN PAGE):




File Edit Selection View
IM
db.php
index.php
login.php
register.php
<html lang="en">
<head>
<style>
nav button {
display: inline-block;
margin: 5px;
padding: 10px 15px;
background-color: #28a745;
color: white;
border: none;
border-radius: 5px;
font-size: 14px;
font-weight: bold;
cursor: pointer;
transition: background-color 0.3s ease;
}
nav button:hover {
background-color: #218838;
}
nav button:active {
background-color: #1e7e34;
}
</style>
</head>
<body>
<header>
<h1>Medicine Inventory Management</h1>
<nav>
<?php if (isset($_SESSION['role']) && $_SESSION['role'] === 'admin'): ?>
<button onclick="location.href='php/add_product.php'">Add Product</button>
<button onclick="location.href='php/view_products.php'">View Products</button>
<?php endif; ?>
</nav>
<nav>
Shop
Cart
<?php if (isset($_SESSION['user_id'])): ?>
Logout
<?php else: ?>
Login
<?php endif; ?>
</nav>
</header>

JAVA SCIPT:

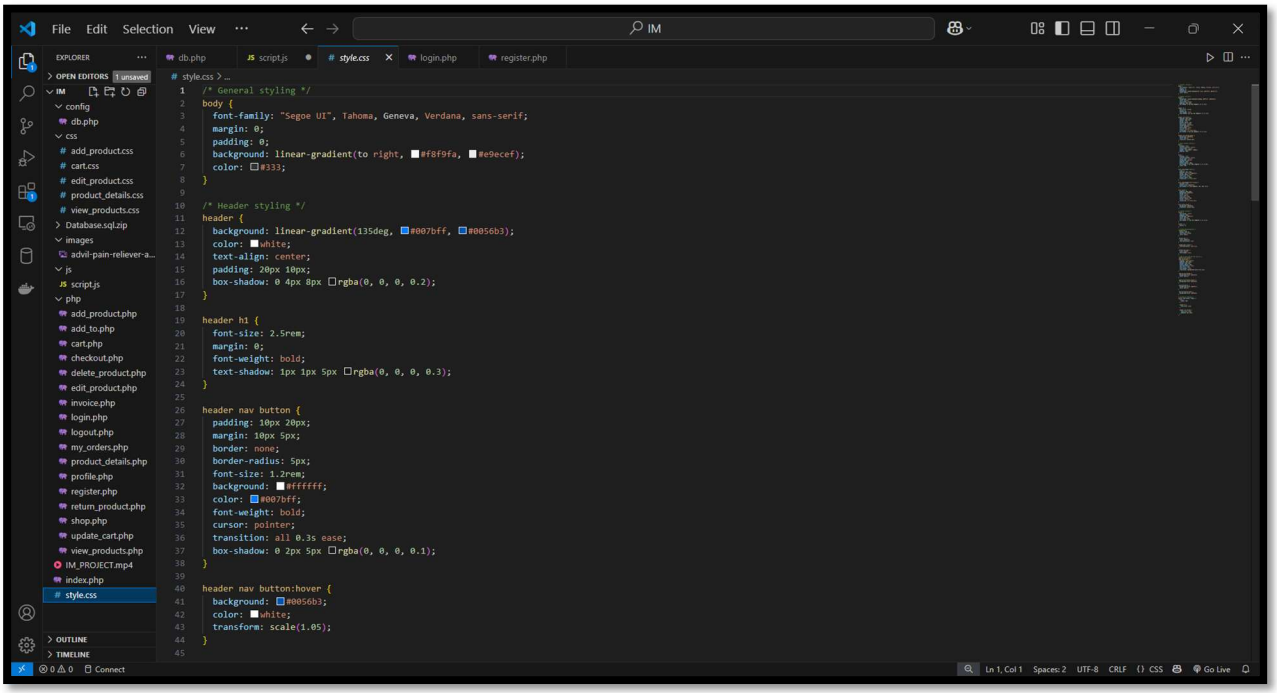

File Edit Selection View
IM
db.php
script.js
login.php
register.php
function highlightLowStock() {
const rows = document.querySelectorAll("table tr");
rows.forEach(row => {
const quantityCell = row.querySelector("td:nth-child(3)");
if (quantityCell) {
const quantity = parseInt(quantityCell.textContent, 10);
if (quantity < 10) {
row.classList.add("low-stock");
}
}
});
}
function setupSearchFeature() {
const searchInput = document.getElementById("searchBar");
const rows = document.querySelectorAll("table tbody tr");
searchInput.addEventListener("input", () => {
const query = searchInput.value.toLowerCase();
rows.forEach(row => {
const productName = row.querySelector("td:nth-child(2)").textContent.toLowerCase();
if (productName.startsWith(query)) {
row.style.display = ""; // Show rows that match the query
} else {
row.style.display = "none"; // Hide rows that don't match
}
});
});
}
function setupDeleteConfirmation() {
const deleteButtons = document.querySelectorAll(".delete-btn");
deleteButtons.forEach(button => {
button.addEventListener("click", event => {
if (!confirm("Are you sure you want to delete this product?")) {
event.preventDefault();
}
});
});
}
function applyRowHoverEffect() {
const rows = document.querySelectorAll("table tr");
rows.forEach(row => {
row.addEventListener("mouseover", () => {
row.classList.add("row-hover");
});
row.addEventListener("mouseout", () => {



```
function setupDeleteConfirmation() {
    deleteButtons.forEach(button => {
        button.addEventListener("click", event => {
            event.preventDefault();
        });
    });
}
function applyRowHoverEffect() {
    const rows = document.querySelectorAll("table tr");
    rows.forEach(row => {
        row.addEventListener("mouseover", () => {
            row.classList.add("row-hover");
        });
        row.addEventListener("mouseout", () => {
            row.classList.remove("row-hover");
        });
    });
}
function manualSearch() {
    const query = document.getElementById("searchBar").value.toLowerCase();
    const rows = document.querySelectorAll("table tbody tr");

    rows.forEach(row => {
        const productName = row.querySelector("td:nth-child(2)").textContent.toLowerCase();
        if (productName.includes(query)) {
            row.style.display = ""; // Show rows that match the query
        } else {
            row.style.display = "none"; // Hide rows that don't match
        }
    });
}
document.addEventListener("DOMContentLoaded", () => {
    highlightLowStock();
    setupSearchFeature();
    setupDeleteConfirmation();
    applyRowHoverEffect();
});
```

CSS CODE:



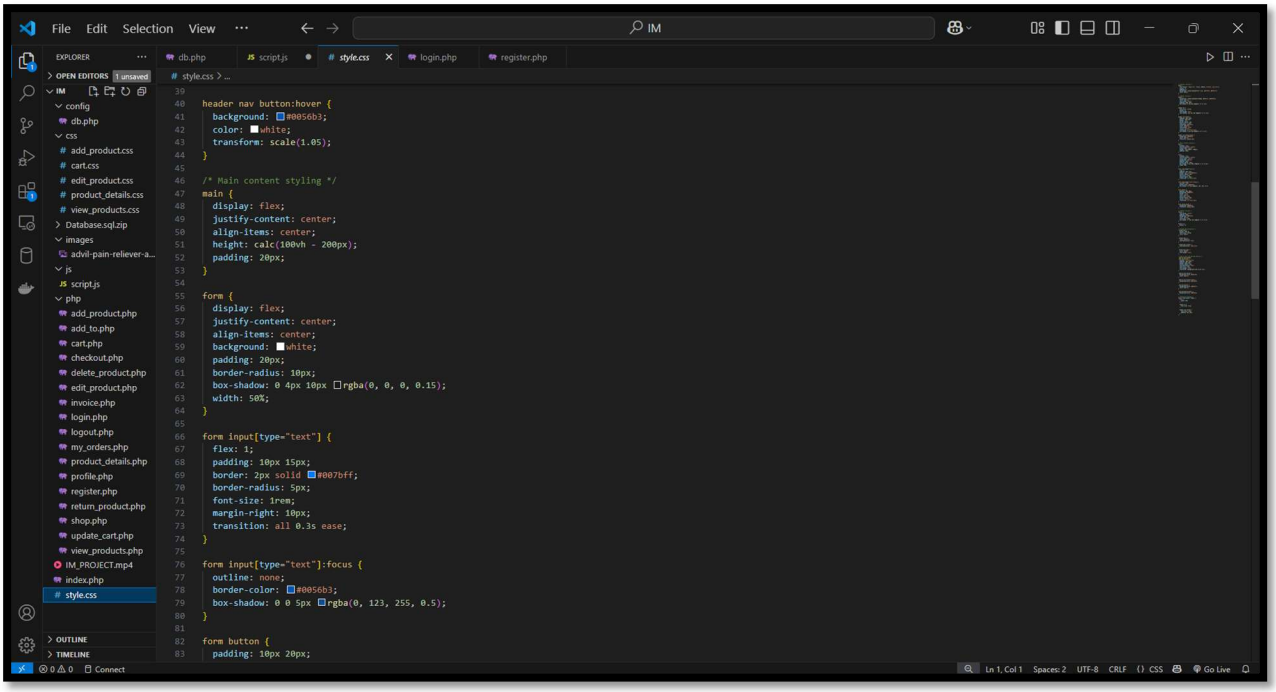
```
/* General styling */
body {
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
  margin: 0;
  padding: 0;
  background: linear-gradient(to right, #f8f9fa, #e9ecef);
  color: #333;
}

/* Header styling */
header {
  background: linear-gradient(135deg, #007bff, #0056b3);
  color: white;
  text-align: center;
  padding: 20px 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
}

header h1 {
  font-size: 2.5rem;
  margin: 0;
  font-weight: bold;
  text-shadow: 1px 1px 5px rgba(0, 0, 0, 0.3);
}

header nav button {
  padding: 10px 20px;
  margin: 10px 5px;
  border: none;
  border-radius: 5px;
  font-size: 1.2rem;
  background: #ffffff;
  color: #007bff;
  font-weight: bold;
  cursor: pointer;
  transition: all 0.3s ease;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

header nav button:hover {
  background: #0056b3;
  color: white;
  transform: scale(1.05);
}
```

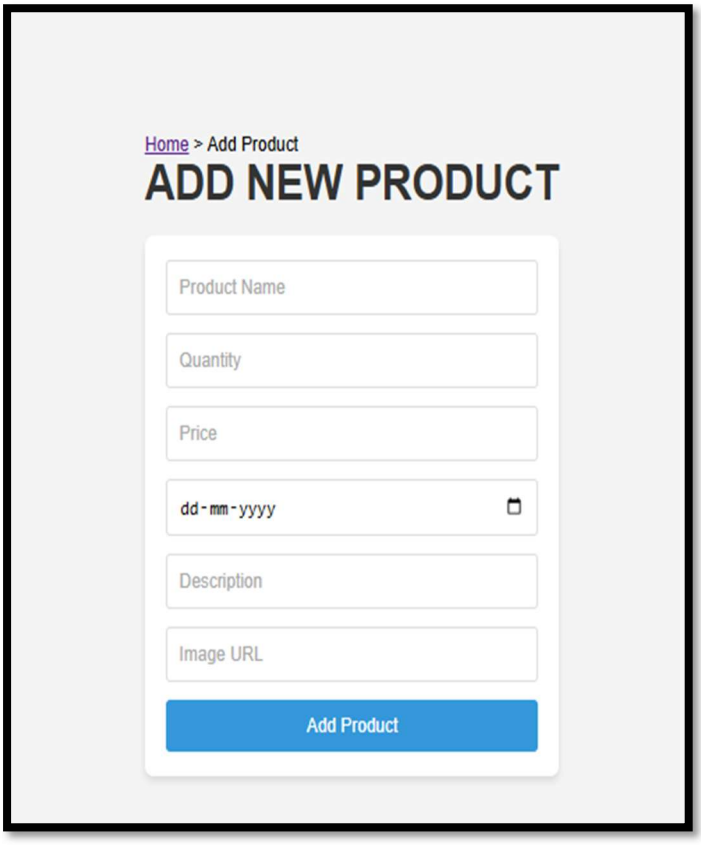

File Edit Selection View ···
IM
EXPLORER
OPEN EDITORS 1 unsaved
IM
config
db.php
css
add_product.css
cart.css
edit_product.css
product_details.css
view_products.css
Database.sql.zip
images
advil-pain-reliever-a...
js
script.js
php
add_product.php
add_to.php
cart.php
checkout.php
delete_product.php
edit_product.php
invoice.php
login.php
logout.php
my_orders.php
product_details.php
profile.php
register.php
return_product.php
shop.php
update_cart.php
view_products.php
IM_PROJECT.mp4
index.php
style.css
OUTLINE
TIMELINE
db.php
script.js
style.css
login.php
register.php
style.css > ...
header nav button:hover {
background: #0056b3;
color: white;
transform: scale(1.05);
}
/* Main content styling */
main {
display: flex;
justify-content: center;
align-items: center;
height: calc(100vh - 200px);
padding: 20px;
}
form {
display: flex;
justify-content: center;
align-items: center;
background: white;
padding: 20px;
border-radius: 10px;
box-shadow: 0 4px 10px rgba(0, 0, 0, 0.15);
width: 50%;
}
form input[type="text"] {
flex: 1;
padding: 10px 15px;
border: 2px solid #007bff;
border-radius: 5px;
font-size: 1rem;
margin-right: 10px;
transition: all 0.3s ease;
}
form input[type="text"]:focus {
outline: none;
border-color: #0056b3;
box-shadow: 0 0 5px rgba(0, 123, 255, 0.5);
}
form button {
padding: 10px 20px;
Connect
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS Go Live

Chapter 5

RESULTS AND DISCUSSIONS

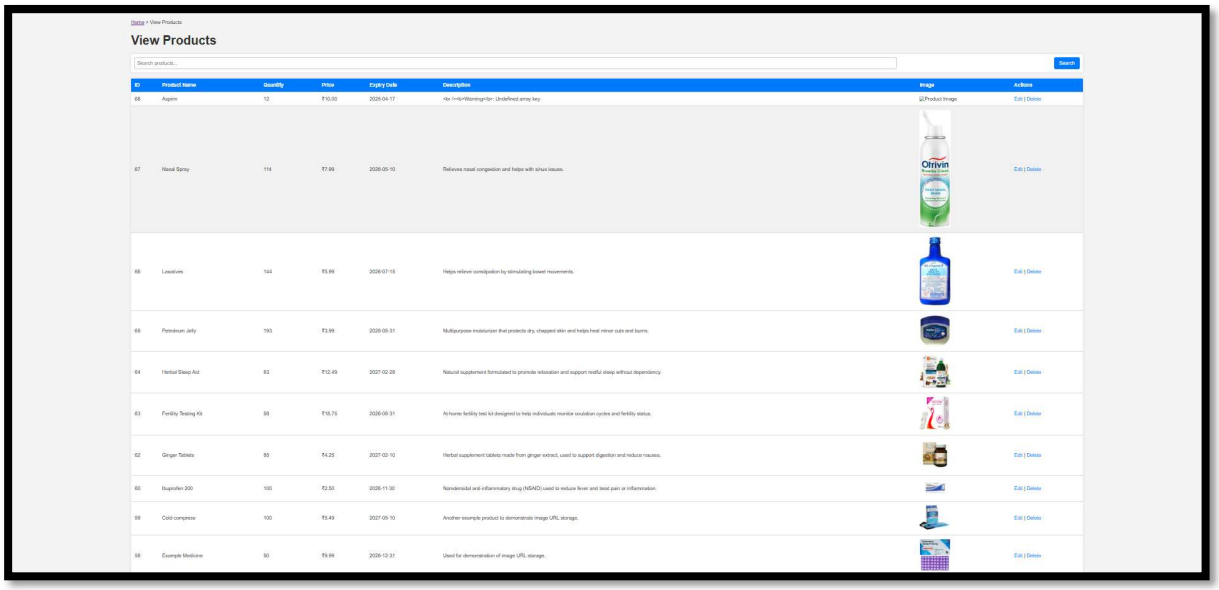
THE GUI:

ADD PRODUCT:



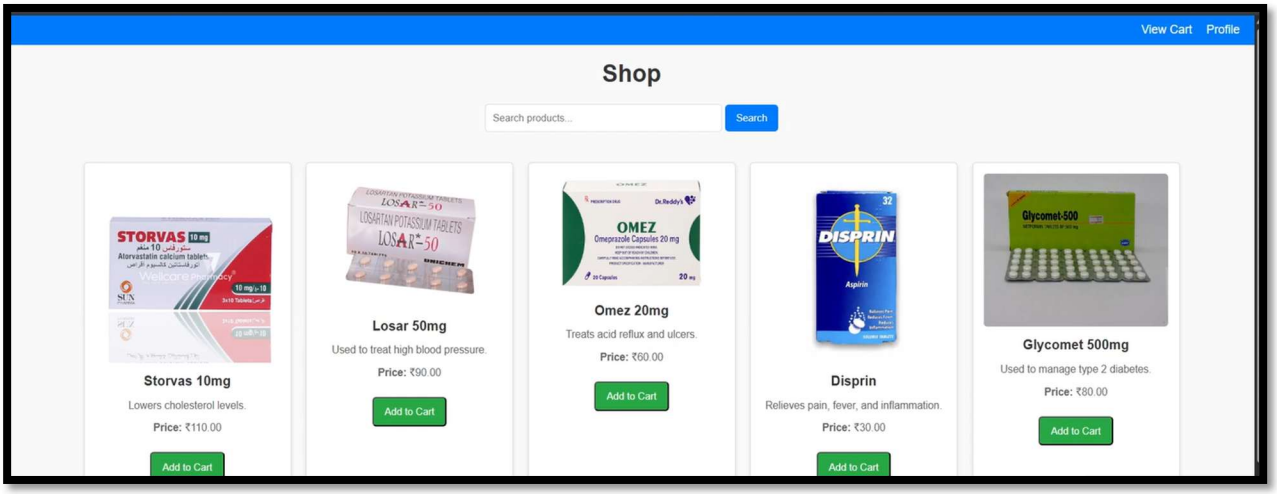
VIEW PRODUCT:

For Adjusting the Stock amount & deleting.



THE SHOP:

For The Users for Purchasing

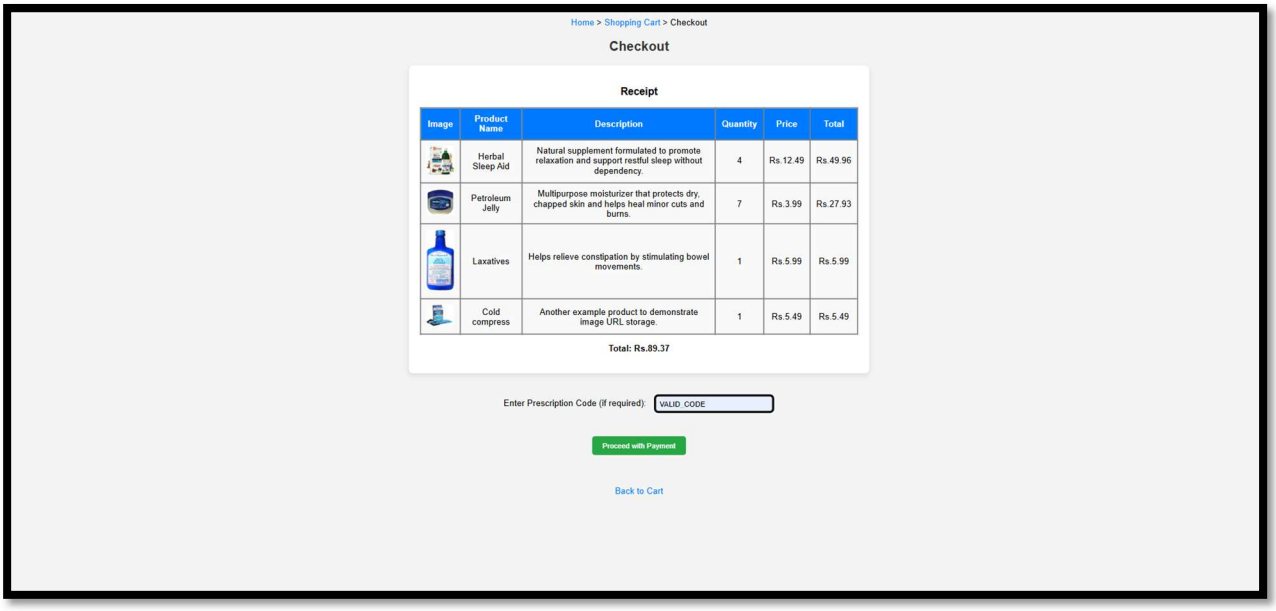


VIEW CART:

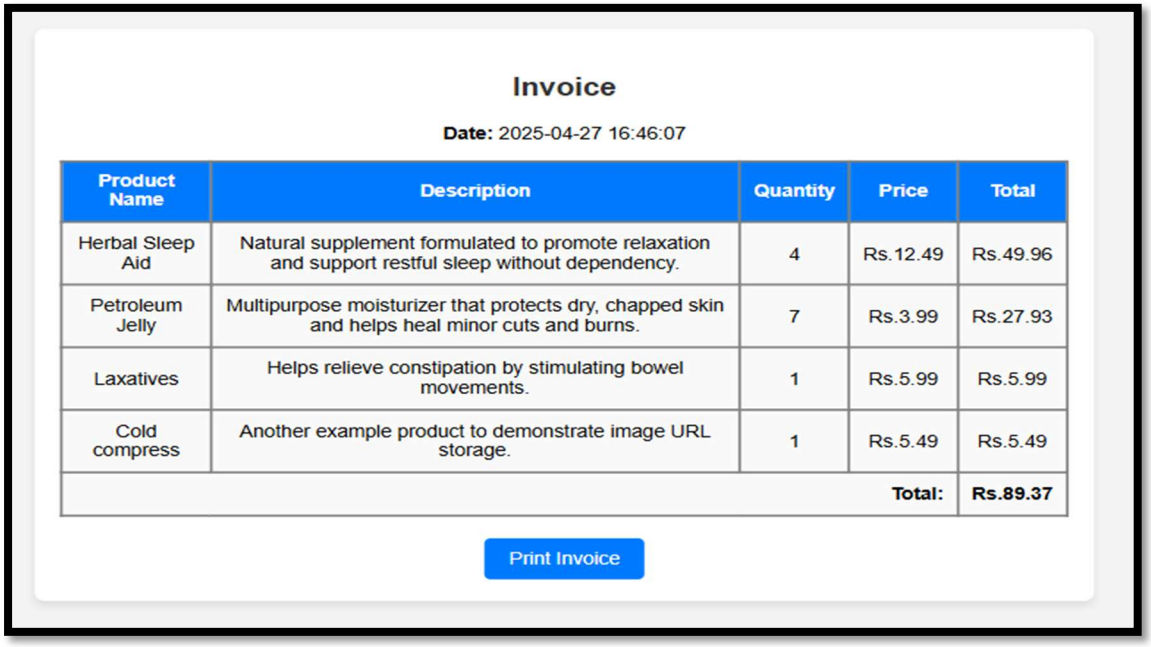
Cart Feature for Purchased Products.

CHECKOUT:

Checkout For the Medicine with Prescription Code as Proof.



INVOICE:

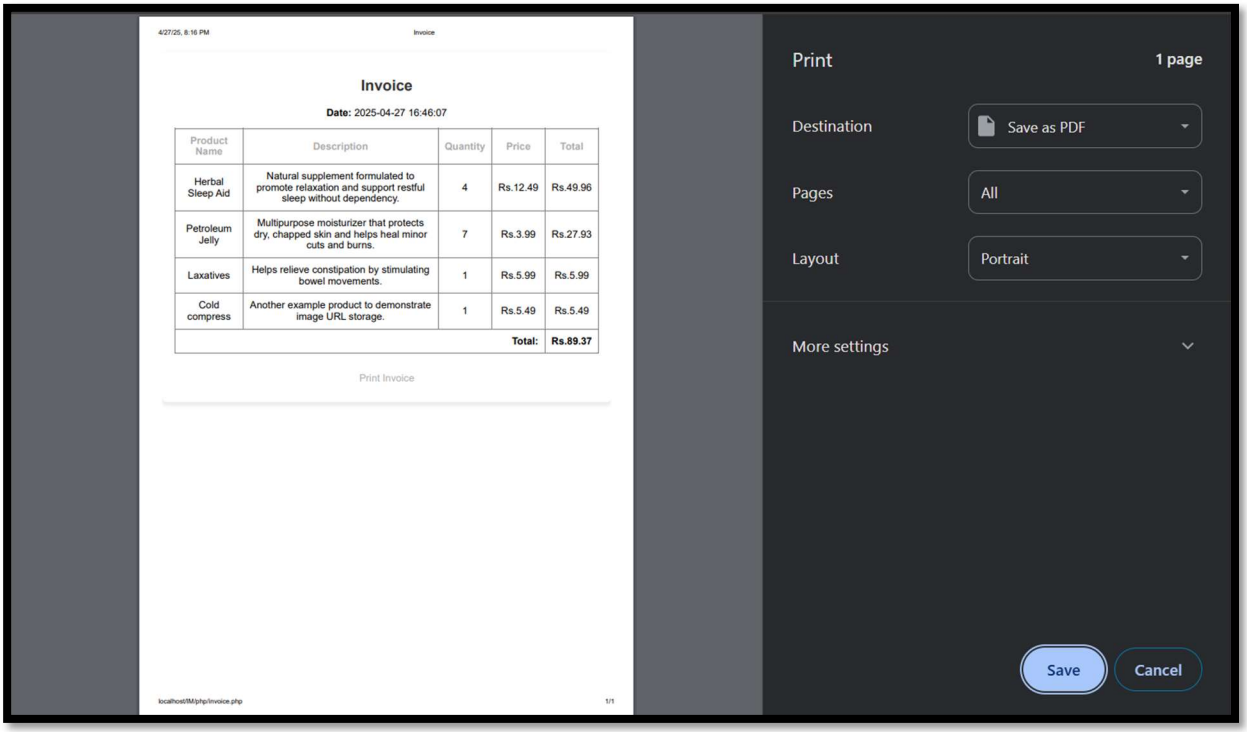


Invoice

Date: 2025-04-27 16:46:07

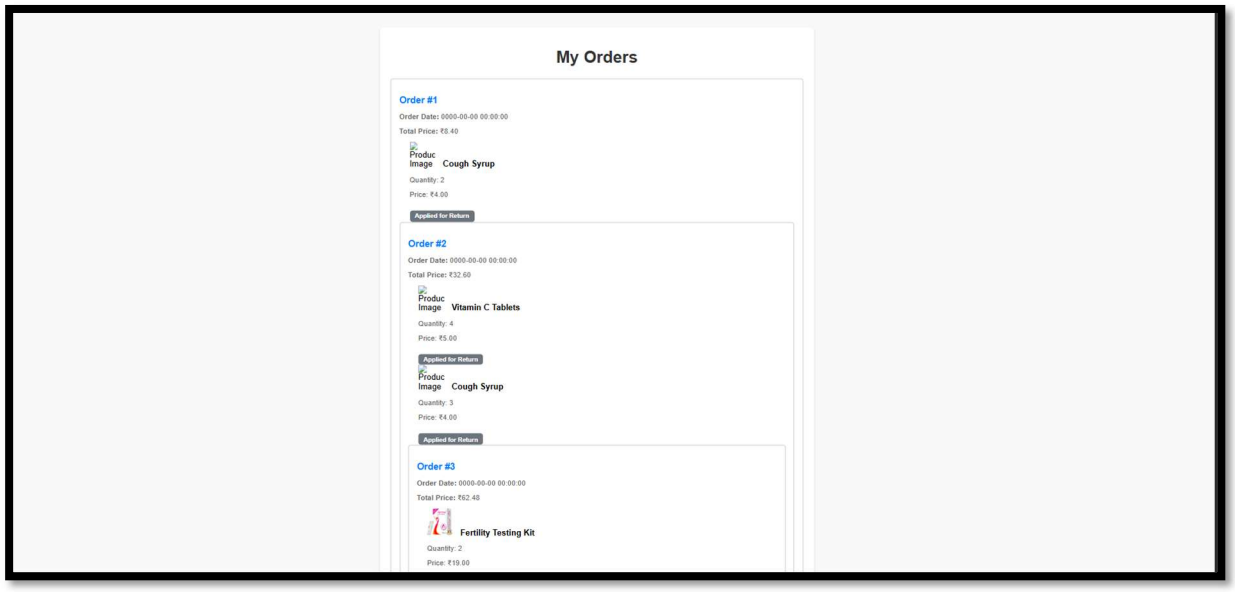
Product Name	Description	Quantity	Price	Total
Herbal Sleep Aid	Natural supplement formulated to promote relaxation and support restful sleep without dependency.	4	Rs.12.49	Rs.49.96
Petroleum Jelly	Multipurpose moisturizer that protects dry, chapped skin and helps heal minor cuts and burns.	7	Rs.3.99	Rs.27.93
Laxatives	Helps relieve constipation by stimulating bowel movements.	1	Rs.5.99	Rs.5.99
Cold compress	Another example product to demonstrate image URL storage.	1	Rs.5.49	Rs.5.49
			Total:	**Rs.89.37**

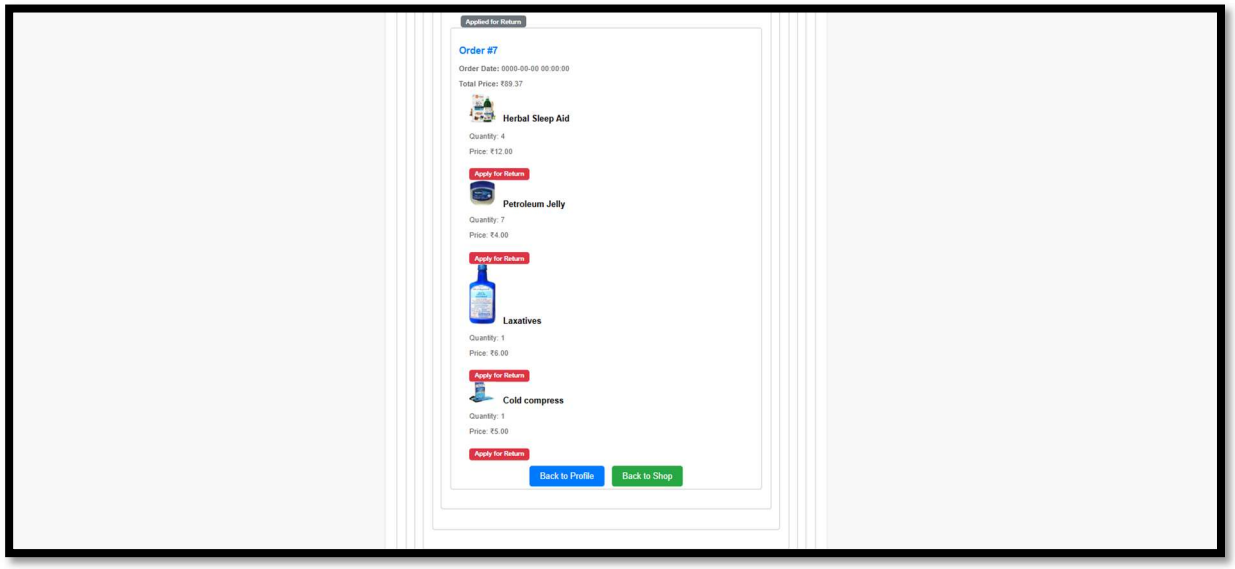
Print Invoice



RETURN PAGE:

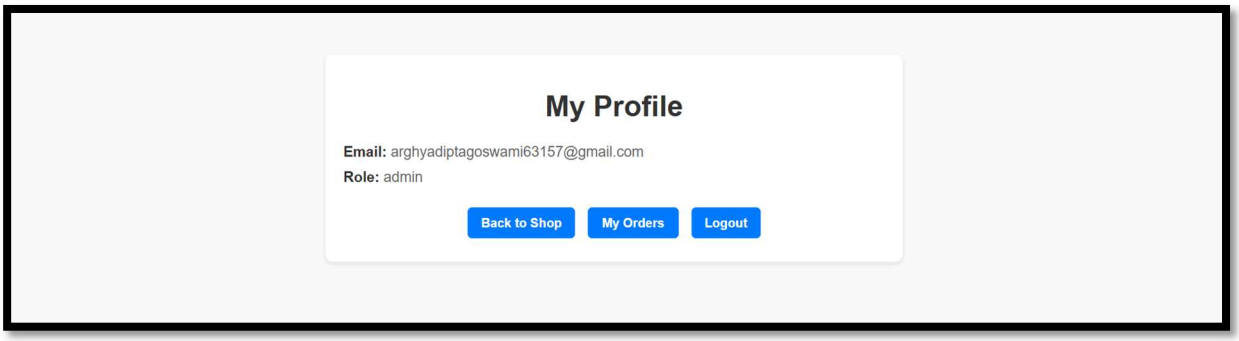
For Order Return Page.



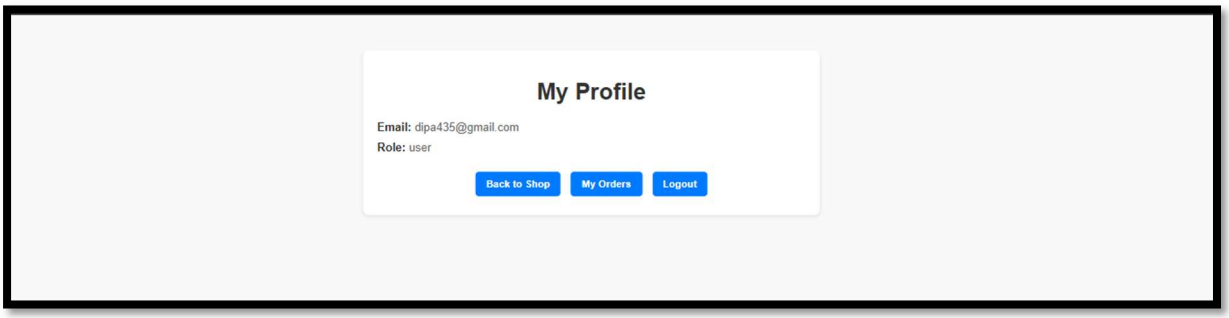


PROFILE:

Admin Profile (For developers and employs)



User Profile (For Customers)



Chapter 6

CONCLUSION

In recent years, the healthcare sector, particularly pharmacies and small medical stores, has faced increasing challenges due to inefficient inventory management practices. Rapid urbanization, growing population demands, supply chain disruptions, and unexpected events like the COVID-19 pandemic have made it even more difficult to maintain proper medicine availability. Manual record-keeping methods, though once sufficient, have now become outdated and error-prone, leading to stock shortages, expired medicines, and financial losses for pharmacies.

The *Medicine Inventory Management System* developed in this project addresses these issues by offering a structured, automated solution using web technologies. The system successfully implements key functionalities:

- **Real-time Monitoring** – Tracks the stock levels and updates them automatically.
- **Expiry Tracking** – Monitors and alerts when medicines are nearing their expiration date.
- **Stock Movement Detection** – Manages the inflow and outflow of medicines efficiently.
- **User Interaction** – Provides an intuitive interface for pharmacy staff to manage inventory easily.

This system helps pharmacies enhance their operational efficiency, reduce human errors, minimize wastage, and ensure the continuous availability of essential medicines. It not only improves day-to-day inventory management but also prepares businesses for future scalability and smarter healthcare service delivery.

FUTURE WORK

The current system successfully handles core features including real-time stock tracking, expiry date monitoring, low stock alerts, and simple user management. However, there is scope for further enhancement and expansion. Future work can focus on integrating more advanced functionalities, such as AI-based demand forecasting to predict future stock needs based on sales trends and seasonal demands. This would help pharmacies prepare better for high-demand periods and minimize wastage.

Currently, the system is scaled for small and medium-sized pharmacies, managing a limited inventory size and user base. Future improvements can include scaling the system to handle multi-branch pharmacy chains with centralized management, multi-user roles with specific access levels, and real-time synchronization across different locations. Additional features like supplier order automation, barcode scanner integration, and mobile app compatibility for remote inventory access can also be developed. Incorporating cloud-based data backup for improved security and reliability, and enhancing the reporting features with customizable analytics dashboards, would further extend the system's capabilities and make it suitable for larger organizations like hospitals, wholesale distributors, and healthcare canter's.

REFERENCES

[1] Yadav, A., & Singh, P. (2020). *Design and Implementation of Web-Based Pharmacy Inventory Management System*. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 6(1), 101-107.

[2] Rajput, M., & Goyal, S. (2021). *Real-Time Inventory Tracking System for Healthcare Facilities Using PHP and SQLite*. International Journal of Advanced Computer Science and Applications, 12(4), 78-83.

[3] Ghobakhloo, M. (2018). *The Future of Healthcare Systems: Integrating Information Systems with Pharmacy Management*. Journal of Health Management, 20(1), 1-15. https://doi.org/10.1177/0972063417744112

[4] Gupta, A., & Bhatia, P. K. (2018). *Enhancing Efficiency of Inventory Management Systems through Web Technologies*. Procedia Computer Science, 132, 1095-1103. https://doi.org/10.1016/j.procs.2018.05.220

[5] Rathore, A. K. S., & Gupta, A. (2019). *Development of Pharmacy Management System using PHP and MySQL*. International Journal of Innovative Research in Computer and Communication Engineering, 7(3), 3056-3060.