PHARMIO
ONLINE PHARMACY STORE

Mini Project Report

Submitted by

Fathima Hazbin R

Reg. No.: AJC22MCA-2044

In Partial Fulfillment for the Award of the Degree of

MASTER OF COMPUTER APPLICATIONS
(MCA TWO YEAR)
[Accredited by NBA]

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2023-2024

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**PHARMIO"** is the bona fide work of **FATHIMA HAZBIN R (Regno: AJC22MCA-2044)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2023-24.

Ms. Anit James

Internal Guide

Ms. Meera Rose Mathew

Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

DECLARATION

I hereby declare that the project report **"PHARMIO"** is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2023-2024.

Date: 05-12-2023 **FATHIMA HAZBIN R**

KANJIRAPPALLY **Reg: AJC22MCA-2044**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ms. Anit James** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

FATHIMA HAZBIN R

ABSTRACT

This project report delves into the development and implementation of an Online Pharmacy Management System, a web-based application designed to revolutionize the way customers access pharmaceutical services. The system facilitates the seamless ordering and purchase of a diverse range of medicines, including prescription drugs, over-the-counter medications, supplements, vitamins, and personal care items. Beyond its transactional capabilities, the online pharmacy management system serves as an informative resource, offering guidance on various health topics and conditions.

A key objective of the system is to enhance the accessibility, affordability, and overall convenience of pharmacy services for customers. The platform not only allows users to conveniently order medications from the comfort of their homes but also ensures the safe and timely delivery of orders to their doorstep through reliable shipping partners. This report explores the features, functionalities, and the broader impact of the Online Pharmacy Management System, highlighting its potential to transform and improve the pharmaceutical service landscape.

CONTENT

List of Abbreviation

HTML	:	Hyper Text Markup Language
UML	-	Unified Modelling Language
CSS	-	Cascading Style Sheet
GB	-	GigaBytes
SSD	-	Solid-State Drive
JS	-	JavaScript
AJAX	-	Asynchronous JavaScript and XML
PHP	-	PHP:Hypertext Preprocessor

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

An online pharmacy management system is a web-based application that allows customers to order and purchase medicines from the comfort of their homes. An online pharmacy management system offers a wide range of medicines, such as prescription drugs, over-the counter drugs, supplements, vitamins, personal care items, and more. An online pharmacy management system also provides useful information and guidance on various health topics and conditions. It also delivers the orders to the customers' doorstep through reliable shipping partners. An online pharmacy management system aims to improve the accessibility, affordability, and convenience of pharmacy services for the customers.

1.2 PROJECT SPECIFICATION

The store can have 4 modules.

1. Admin: The admin is the owner or manager of the online pharmacy business. The admin user can perform various tasks and functions to manage and monitor the online pharmacy operations, such as,
    - Adding, editing, and deleting medicines.
    - Manage orders, customers, suppliers, staff, and reports.
    - Customize the settings and preferences of the online pharmacy website, such as logo, tax rate, etc.
    - Manage customer queries using chatbot
2. Staff: The staff is the employee or worker of the online pharmacy business who handles the payment transactions of the customers. The cashier user can use the website to enter the
    - Details of the medicines or services purchased by the customers, such as the name, quantity, price, and discount.
    - Calculate the total amount.
    - Accept different payment methods, such as cash, card, or online.
    - Generate and print invoices and receipts for the customers.
3. Customer: The customer is the user who orders and purchases medicines and health medicines from the online pharmacy website. The customer user can perform various tasks and functions to access and use the online pharmacy services, such as
    - Searching for medicines
    - Placing orders ,Cancel orders and Making Payments
    - Managing accounts and profiles

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

The Online Pharmacy Management System is a web-based application developed to facilitate the seamless ordering and purchase of medicines, providing customers with the convenience of accessing pharmaceutical services from the comfort of their homes. This system not only offers a comprehensive range of medicines, including prescription drugs, over-the-counter medications, supplements, and personal care items but also serves as an informative resource for health-related guidance.

2.2 EXISTING SYSTEM

2.2.1 NATURAL SYSTEM STUDIED

In Pharmio Medical Store, the usual way of managing things is pretty old-fashioned. They rely a lot on doing things by hand, like writing things down on paper and dealing with customers face-to-face. Without using a special computer system, it can lead to mistakes and make it harder to keep everything organized. This old-fashioned way might slow down how the pharmacy works and could make it tough to coordinate different tasks. If Pharmio doesn't use a modern computer system, it might be harder for them to work well, collaborate, and reach more people who need their services. So, having a good computer system in Pharmio is really important to make everything smoother, more organized, and easier for everyone.

2.2.2 DESIGNED SYSTEM STUDIED

The newly designed system for Pharmio Medical Store is like a smart and user-friendly assistant for managing the online pharmacy. It introduces a comprehensive digital framework that makes buying medicines and healthcare products super easy. With this system, customers can quickly browse, order, and complete transactions securely. It also keeps track of the inventory, making sure there's always enough stock and reducing the chances of products running out. The design focuses on simplicity, with easy navigation and intuitive features for a smooth and hassle-free experience. Pharmio's new system is all about making online pharmacy activities efficient and user-friendly, providing a reliable and secure platform for customers to access their healthcare needs.

2.3 DRAWBACKS OF EXISTING SYSTEM

- **Limited Accessibility:** Traditional pharmacies may not be easily accessible for individuals in remote areas or those with mobility challenges.
- **Fixed Operating Hours:** Brick-and-mortar pharmacies typically have set operating hours, which may not align with the varied schedules of customers.

- **Product Availability:** Traditional pharmacies might face challenges in maintaining a diverse and extensive stock of medications and healthcare products.
- **Time and Travel Constraints:** Physically visiting a traditional pharmacy can be time-consuming and inconvenient, requiring travel to the physical location
- **Privacy Concerns:** Some individuals may feel uncomfortable discussing their medical conditions or purchasing specific medications in person at a traditional pharmacy.
- **Possibly Higher Costs:** Operating physical stores can lead to higher overhead costs for traditional pharmacies, which may result in slightly higher prices for medications compared to online pharmacies.

2.4 PROPOSED SYSTEM

The envisioned system for Pharmio is poised to transform the landscape of online pharmacy services, introducing a robust platform designed to streamline the medication procurement process and strengthen connections within the healthcare community. This system aims to provide a user-friendly interface catering to customers seeking pharmaceutical products, pharmacists managing inventory, and healthcare providers collaborating on prescription verification. Incorporating the power of PHP in the backend and HTML, CSS, and Bootstrap in the frontend, the system ensures responsiveness and dynamism for an intuitive and efficient user experience. It facilitates secure communication, prescription uploads, and seamless payment processing. By deploying an effective and reliable system, Pharmio aspires to redefine the online pharmacy domain, enhancing accessibility, security, and efficiency for users across the healthcare spectrum.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- **Enhanced Accessibility:** The proposed system for Pharmio improves accessibility by allowing customers to order medicines from anywhere with an internet connection, addressing the limitation of traditional pharmacies being confined to specific locations.
- **Extended Operating Hours and Convenience:** The online system operates 24/7, offering customers the convenience of purchasing medicines at any time, overcoming the fixed operating hours constraint of traditional pharmacies.
- **Wider Product Availability:** Pharmio's digital platform ensures a broader range of available products compared to traditional pharmacies, which might face challenges in maintaining a diverse stock.
- **Time and Travel Savings:** With the proposed system, customers can save time and effort as they can order medicine from the comfort of their homes, eliminating the need for physical travel to a traditional pharmacy.
- **Privacy and Discretion:** The online system provides a discreet option for customers who

prefer privacy in their medical transactions, addressing the discomfort some individuals may feel discussing their conditions in person at a traditional pharmacy.

- **Cost Savings:** Online pharmacies often offer competitive pricing due to reduced overhead costs, potentially resulting in cost savings for customers compared to traditional brick-and-mortar pharmacies.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

A feasibility study is conducted to determine whether the project will, upon completion, fulfil the objectives of the organization in relation to the work, effort, and time invested in it. A feasibility study enables the developer to predict the project's usefulness and potential future. The premise for a feasibility study is the system proposal's viability, which includes Ohe impact on the organisation, ability to meet user needs, and effective use of resources. AO a result, before a new application is accepted for development, it often undergoes a feasibility assessment.

The document outlines the project's viability and contains a number of factors that were carefully considered throughout this project's feasibility study, including its technical, economic, and operational viabilities. It has the following characteristics:

3.1.1 Economical Feasibility

Economic feasibility analysis assesses whether a project aligns with an organization's financial objectives. It includes cost estimation, revenue projection, profitability assessment, risk evaluation, and comparison with alternatives. The analysis determines if a project will yield positive returns, when they will be realized, and how it responds to changes in key variables. Ultimately, it informs decisions about project viability, funding, and adjustments to enhance financial feasibility.

Based on these findings, it is recommended to proceed with the development and implementation of the Online Pharmacy Management System. The project not only promises to enhance operational efficiency and customer satisfaction but also make it economically feasible and beneficial. In this project: Online Pharmacy Management System the cost is divided into; system costs, development costs and hosting costs. All collections indicate that the project was developed at a modest cost. As open-source software was used to develop it entirely.

1. Do the resources needed the exist?

   Yes. The availability of economic resources for the pharmacy management system project should align with the project's financial requirements and the organization's capacity to fund and sustain it. By the conducted economic feasibility assessment the existing necessary resources for the development of the project is determined.

2. Will the proposed health service or intiative lead to better use of resources to improve health outcomes, when compared with the options?

   Yes. To determine whether the proposed health service or initiative will lead to better use of resources and improved health outcomes compared to other options, a comprehensive cost-effectiveness analysis and outcome evaluation are essential which is

done through economical feasibility.

3.1.2 Technical Feasibility

Technical feasibility analysis evaluates whether a proposed project can be successfully implemented from a technological perspective. It assesses whether the required technology, resources, and expertise are available or can be feasibly acquired. This analysis considers factors like system compatibility, infrastructure, software and hardware requirements, and the availability of skilled personnel. It helps determine if the project can be technically executed within constraints and if it aligns with the organization's technical capabilities and objectives.

Technical feasibility is a critical assessment of whether the Online Pharmacy Management System can be successfully developed, implemented, and operated from a technical perspective. These infrastructure components meet the system's requirements for hosting, data storage, and network connectivity. Python, the selected programming language, is robust and widely used for web application development. The required technical infrastructure and technology stack are readily available and suitable for the project's needs. The project's technical feasibility ensures that it can be effectively developed, deployed, and maintained to meet its objectives.

1. Do stakeholders have the expertise needed?

   Yes. The expertise of stakeholders is a critical factor. It is essential to assess whether the stakeholders possess the required expertise for various aspects of the project.

2. Are the additional resources needed in the health system including infrastructure, skills-sets, or job aids?

   Yes. The need for additional resources will depend on the complexity of the project, the existing capabilities of the healthcare system, and the project's specific goals. The conducted assessment of the factors to determine the precise resource requirements for the successful implementation of the pharmacy management system.

3. Is the health system ready in terms of requirements?

   Yes, it's ready. Readiness for implementing a pharmacy management system is a multidimensional assessment that encompasses technical, regulatory, organizational, and human factors. Addressing the considerations will help ensure a smoother and more successful implementation process.

3.1.3 Operational Feasibility

Operational feasibility assesses whether a project can be effectively operated and maintained once implemented. It involves evaluating resource availability, skill and expertise, process integration, system reliability, ongoing costs, user adoption, and compliance. This analysis ensures that the

organization has the necessary resources, capabilities, and processes to support the project's long-term operation and align it with its goals.

Operational feasibility assesses whether a project can be effectively operated and maintained once implemented. It involves evaluating resource availability, skill and expertise, process integration, system reliability, ongoing costs, user adoption, and compliance. This analysis ensures that the organization has the necessary resources, capabilities, and processes to support the project's long-term operation and align it with its goals.

1. Do existing health system procedures and protocols support the new service or initiative?

   Yes. The rules and steps in the existing system support the new service or if changes are needed. This ensures that the new service doesn't disrupt the normal work and follows all the health rules. If changes are needed, they're made in a way that doesn't cause problems for patients and staff.

2. How will key collaborators be involved?

   For an online pharmacy management system, key collaborators, such as pharmacists, healthcare providers, and IT experts, are crucial team members who work together to make sure the system benefits both patients and healthcare professionals. They help design the system to fit the pharmacy's needs and workflow, and they test it to make sure it works well. By working together, they aim to improve the way the online pharmacy operates and provides care to customers.

3.1.4 Feasibility Study Questionnaire

1. **Project Overview.**

   An online pharmacy management system is a software application designed to streamline and enhance the operations of a pharmacy. The main goal is to make pharmacy management operations easier, improve customer service, and enhance the overall experience for both customers and staff. It provides a digital platform for pharmacy management, sales, and customer interaction.

   There are four modules in the project.

   i. Admin Module: Create, modify, and delete user accounts (staff, customers, and delivery team members). Add, update, and delete medicines in the inventory. Monitor sales data, generate reports, and manage orders.

   ii. Staff Module: Update stock levels, add new medicines, and manage product information. Process customer orders, including order verification and fulfilment. Assist customers with inquiries and issues.

iii. Customer Module: User registration and login: Customers can create accounts, log in, and manage their profiles. Browse medicines, view product details, add items to the cart, and place orders. Track the status of orders, including estimated delivery times.

iv. Delivery Team Module: Assign orders to delivery team members. Update order delivery status, including successful deliveries and exceptions. Enable communication between the delivery team and customers for smoother deliveries.

Some of benefits are Streamline pharmacy operations, reducing manual processes and paperwork. Provide customers with a user-friendly platform to browse and order medicines. Keep track of medicine stock levels in real-time, minimizing stockouts. Generate reports for sales analysis, customer trends, and inventory management. Ensure data security and compliance with healthcare regulations.

Technical Stacks: PHP, HTML, CSS, JavaScript (Frontend, with potential frontend frameworks), MySQL (or similar Database)

1. **To what extend the system is proposed for?**

The proposed Online Pharmacy Management System is designed to comprehensively address the diverse needs of pharmacy operations, making it a robust and versatile solution. The admin empowers administrators to efficiently manage user accounts, medicines, and sales data, giving them complete control over the system's functionalities. Staff members can manage order processing, and customer support, enhancing their daily tasks. Customers gain access to user-friendly features for browsing medicines, placing orders, and tracking deliveries, and improves their shopping experience. In addition to these, the Delivery Team can optimize order assignments, delivery tracking, and communication, ensuring that the delivery process runs smoothly.

This comprehensive approach addresses the entire spectrum of pharmacy operations, from backend administration to frontend customer interactions and delivery logistics. The system's scalability and adaptability allow it to meet the evolving needs of a pharmacy, making it an extensive solution that improve efficiency, customer satisfaction, and growth within the pharmaceutical domain.

2. **Specify the Viewers/Public which is to be involved in the System.**

The Online Pharmacy Management System is designed to serve various user roles, each with specific access and interaction capabilities. The viewers/public involved are,

a. Admins: Admin users are responsible for overseeing the entire system. They have full access to all modules and functionalities and ensure the system operates smoothly.
b. Staff Members: Pharmacy staff, including pharmacists, sales representatives interact with the system. They use the System to manage inventory, process orders, and provide customer support.
c. Customers: Customers are the end-users of the system who access the System to browse medicines, place orders, track deliveries, and manage their profiles.
d. Delivery Team: Delivery team members are responsible for delivering orders to customers, receive order assignments, update delivery statuses, and communicate with customers.
e. Pharmacy Management: This includes the owners and managers of the pharmacy. They may have admin-level access or specific reporting access to monitor the pharmacy's performance.
f. Developers and IT Support: The development and IT support teams are involved in maintaining, updating, and troubleshooting the system. They ensure the system's technical aspects run smoothly.
g. General Public: While not directly involved in system usage, the general public can indirectly benefit from the system's availability if they become customers of the pharmacy.

By accommodating these different viewers and user roles, the System optimizes pharmacy operations, enhances customer experiences, and ensures regulatory compliance.

3. List the Modules included in your System?

i. Admin
ii. Staff
iii. Customer
iv. Delivery Team

4. Identify the users in your project?

i. Admin user
ii. Staff members
iii. Customers
iv. Delivery team members
v. General public

5. **Who owns the system?**

   The ownership of a project, such as the development and management of an online pharmacy system, can vary depending on the organization or individuals involved. Ownership can be for,

   a) Individual Entrepreneur or Business Owner: In the case of a privately owned pharmacy, the owner of the pharmacy often takes ownership of the project. They are responsible for funding, decision-making, and overseeing the development and management of the online pharmacy system.
   b) Development Team or Software Development Company**: The ownership of the project during development may be with the client (the pharmacy owner or organization). However, the development team or company is responsible for the technical aspects of the project.

   It's crucial to establish clear ownership and governance structures from the outset to ensure the project's success, accountability, and effective decision-making throughout its development and operation.

6. **System is related to which firm/industry/organization?**

   The System can be used by independent pharmacy stores that operate as standalone businesses or small chains can benefit from an online pharmacy management system to streamline their operations, improve customer service, and expand their reach. system can span various segments of the healthcare and pharmaceutical industry, from independent pharmacies to large healthcare organizations and pharmaceutical manufacturers. Its purpose is to enhance the efficiency, accuracy, and convenience of pharmacy operations, regardless of the specific context in which it is employed.

7. **Details of person that you have contacted for data collection?**

   Medical shop: Kaleeckal Medicals,
   Danapady Road,
   Haripad

   Owner: Bukhari AK

   Phone: 9895994960

8. **Questionnaire to collect details about the project? (min 10 questions, include descriptive answers, attach additional docs (e.g. Bill receipts, certificate models), if any?)**

   a. Are you currently using any software or systems for managing your physical store's operations?

Yes, we use point-of-sale (POS) software for inventory management and sales tracking, but we do not have an online platform for customer orders.

b. What features or functionalities do you expect from the Online Pharmacy Store to meet the needs of your customers and your business?

   We expect features such as secure online ordering, real-time inventory updates, prescription uploads, medication reminders, and a user-friendly interface to enhance customer experience.

c. How do you plan to train your staff to manage the online pharmacy store effectively, including order processing and customer support?

   Staff members will undergo training on using the online platform, order processing, and customer support.

d. How will you handle payment processing and ensure secure transactions for online orders?

   Payment processing will be conducted through secure payment gateways, ensuring encrypted and safe transactions.

3.1 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - i3

RAM - 4 G B

Hard disk - 1 T B

3.2.2 Software Specification

Front End	-	HTML,CSS
Back End	-	PHP
Database	-	MySQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP stands as a versatile server-side scripting language, serving as a cornerstone in web development. Widely recognized for its ease of use, PHP allows developers to create dynamic and interactive websites effortlessly. It seamlessly integrates with HTML, offering a flexible environment for crafting both simple web pages and complex web applications. As an open-source language, PHP is freely available, empowering developers with a cost-effective solution for their projects. Its extensive community support and comprehensive documentation make it accessible for developers of varying skill levels. PHP excels in database integration, supporting various databases like MySQL and PostgreSQL, making it a robust choice for backend development. With its continuous evolution and widespread adoption, PHP remains a reliable and adaptable tool for building a wide array of web-based solutions.

3.3.2 MySQL

MySQL stands as a prominent choice in web development, serving as a robust, open-source relational database management system. Widely embraced for its reliability and efficiency, MySQL is particularly well-suited for web applications that demand scalable and secure database solutions. Unlike SQLite, MySQL typically operates with a dedicated server process, offering a powerful backend for dynamic websites and applications. Its versatility shines through in supporting various data types and offering advanced features like transactions and stored procedures. MySQL is renowned for its seamless integration with web technologies, providing a stable foundation for managing and retrieving data. With a strong reputation for performance, data integrity, and community support, MySQL stands as a trusted and widely adopted solution in the realm of relational databases for web development.

CHAPTER 4
SYSTEM DESIGN

4.1 INTRODUCTION

In the realm of product and system development, the design phase marks the initial and critical stage. This creative process is instrumental in ensuring the seamless and effective functioning of any system. Design, in this context, refers to the meticulous application of diverse techniques and principles to meticulously outline a process or system, facilitating its tangible realization. It involves a comprehensive description of the intricacies of a machine or system to enable its accurate implementation. In software development, the design phase holds exceptional significance, serving as the cornerstone of creating efficient and precise software solutions. System design, specifically, involves meticulous planning for the construction of a well-optimized and reliable software framework. It signifies a shift from user-focused documentation to a more programmer-oriented approach, outlining the logical and physical design aspects of the system. This meticulous software design process ensures the seamless integration of functionality and precision, laying the foundation for a robust and user-friendly final product.

4.2 UML DIAGRAM

UML, the Unified Modeling Language, is an industry-standard visual language utilized for specifying, constructing, and documenting software system artifacts. It was established by the Object Management Group (OMG), with the initial UML 1.0 draft presented in January 1997. Unlike traditional programming languages such as C++, Java, or COBOL, UML serves as a graphical language, facilitating the creation of software blueprints. Its primary function is to provide a versatile and comprehensive modeling approach, enabling the visualization, specification, construction, and documentation of software systems. While UML finds extensive use in modeling software systems, its applications are not confined to this domain exclusively, extending to areas like process flows in manufacturing units and beyond. While UML itself is not a programming language, it supports code generation in various programming languages through UML diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Activity diagram
- State chart diagram
- Deployment diagram

- Component diagram

4.2.1 USE CASE DIAGRAM

A use case is instrumental in identifying and organizing the essential requirements of a system, facilitating a structured approach to manage and prioritize these elements. Applied within the Unified Modeling Language (UML), use case diagrams serve as visual representations of real-world systems. They enable the comprehensive modeling of diverse systems, outlining their key functionalities and interactions. A use case diagram generally comprises four fundamental components, contributing to a clear and concise depiction of system requirements and behavior.

Use case diagrams aid in the documentation of a system's functional specifications. They demarcate the system's boundaries, delineating its distinctions from surrounding elements. Actors represent the individuals involved in distinct roles within the system. These diagrams facilitate a comprehensive understanding of the interactions between various entities or components within a specific context or problem. Constructing a proficient use case diagram necessitates adherence to certain rules. Ensuring appropriate nomenclature for use cases and actors is vital. The diagram should clearly illustrate relationships and dependencies while only displaying the essential connections for optimal diagram functionality. Supplementary notes may be utilized when necessary to elucidate critical points.

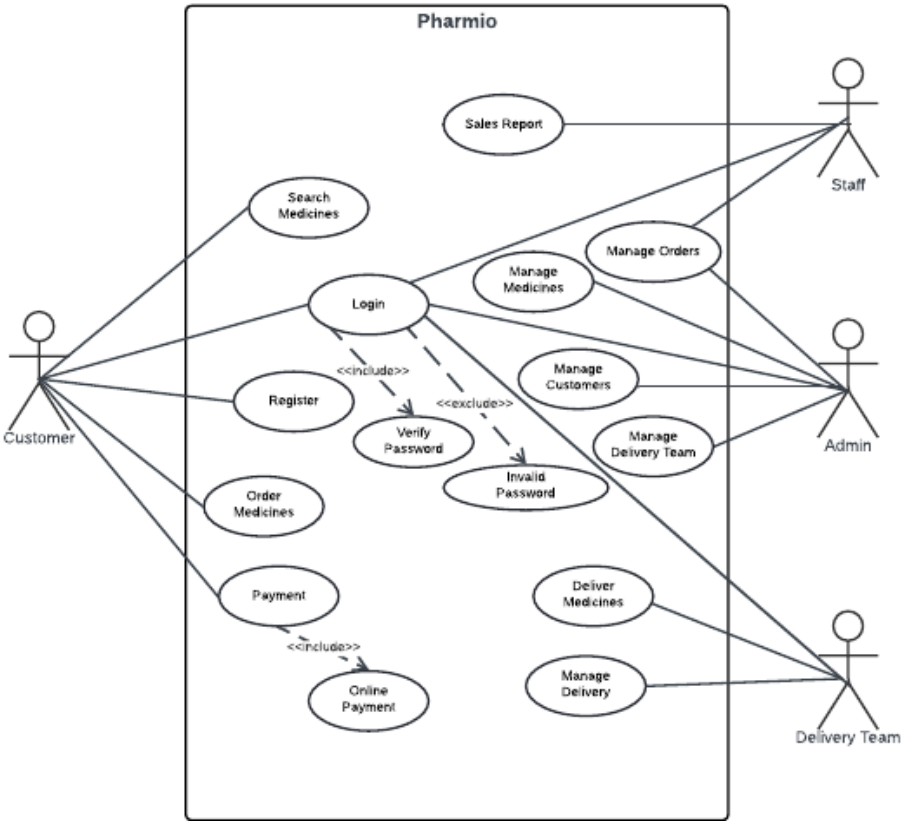


Fig1:Use Case Diagram

4.2.1 SEQUENCE DIAGRAM

A sequence diagram serves to illustrate the systematic interaction between objects in a predetermined order. It elucidates the chronological flow of activities within a system. Referred to as event diagrams or event scenarios, sequence diagrams provide a comprehensive overview of the interplay among various system components, showcasing the specific sequence of their actions. These diagrams are valuable tools for both business professionals and software developers, enabling them to comprehend and depict the requirements of both new and existing systems effectively.

Notations in Sequence Diagrams:

i. **Actors**: Represented as non-system individuals who utilize the system, actors play distinct roles within the diagram, depicted as simple stick figures.

ii. **Lifelines**: Each component within the system is denoted as a lifeline at the top of the sequence diagram.

iii. **Messages**: Objects communicate through the exchange of messages, depicted as arrows following the order of the lifeline. These messages form the core components of the sequence diagram.

iv. **Guards:** Utilized to define various conditions, guards restrict message flow based on specific conditions. They provide guidelines for software developers, outlining the rules and processes within the system..

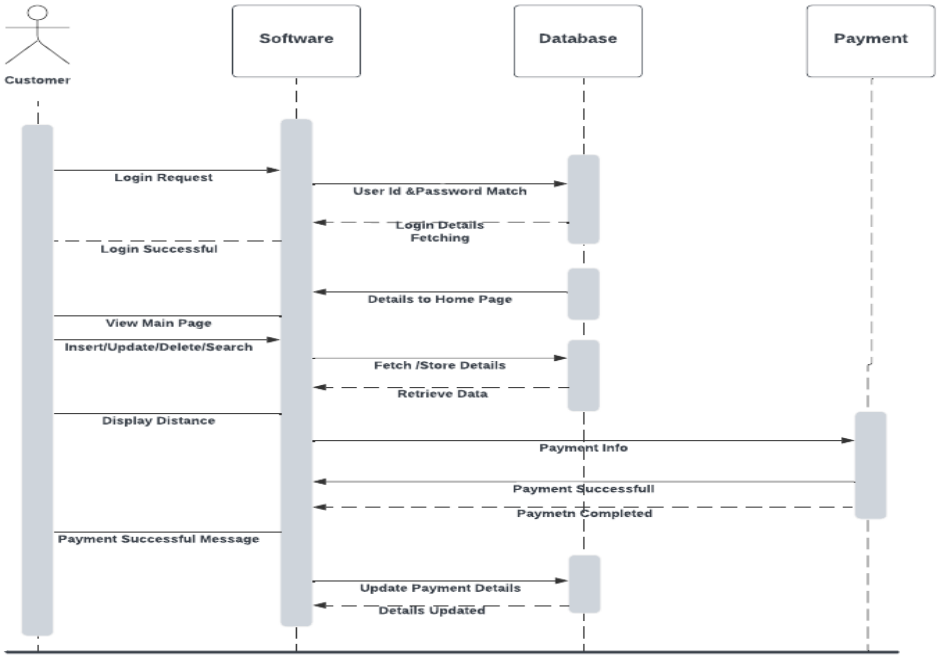


Fig2:Sequence Diagram

2.2 State Chart Diagram

The state machine diagram, also known as a state chart, effectively represents the various states of an object within a system and their sequential transitions. It provides a comprehensive overview of the system's behavioral patterns, delineating the object's progression through different states. It is instrumental in illustrating the interworking dynamics of various elements within a system, be it a group of individuals, a team, a larger collective, or an entire organization. The state machine diagram serves as a strategic blueprint, elucidating how objects evolve during different events and encapsulating the specific states that each element assumes within the system.

Notations in a State Machine Diagram:

• **Initial** state: Signifying the system's initiation, this notation is depicted as a black circle.

• **Final state**: Denoting the system's culmination, this notation is depicted as a filled circle within another circle.

• **Decision box**: Shaped like a diamond, it aids in making decisions based on the evaluation of a guard.

• **Transition**: Represents the shift of power or control from one state to another, illustrated as an arrow with a label indicating the cause of the transition.

• **State box**: Reflects the current status of an element within a group, represented as a rectangular shape with rounded corners.

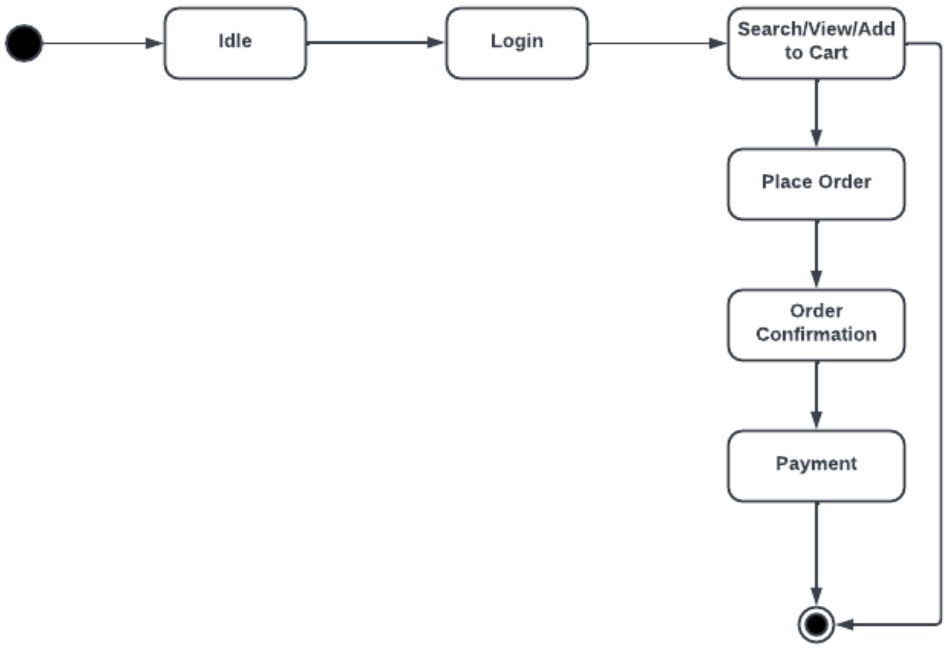


Fig3:State Chart Diagram

4.2.2 Activity Diagram

The activity diagram visually portrays the sequential or simultaneous occurrence of events and activities, effectively illustrating how tasks unfold in a predetermined order. It serves as a comprehensive representation of the flow of activities, demonstrating the interconnectedness of different processes. The activity diagram showcases a variety of flows, encompassing actions occurring sequentially, in parallel, or along distinct pathways. To facilitate these dynamic flows, activity diagrams are equipped with tools such as fork and join, enabling the depiction of multiple simultaneous activities. This type of diagram is often referred to as an object-focused illustration, emphasizing the systematic portrayal of a process's workflow and operational intricacies.

Components of an Activity Diagram:

a) Activities: These refer to the grouping of behaviors into a sequence of actions, visually represented as interconnected nodes. The actions progress from the initial point to the end point, outlining the various tasks, control mechanisms, and resources utilized throughout the process.

b) Activity partition / swim lane: This organizational tool groups similar tasks into distinct categories, enhancing the modularity and comprehensibility of the activity diagram. It aids in the categorization and visualization of different activity sets, offering a clear representation of their interdependencies and interactions.

c) Forks: Fork nodes enable the concurrent execution of various components of a task, allowing the simultaneous progression of multiple activities. They facilitate the divergent flow of information, ensuring that data is dispersed efficiently across different pathways.

d) Join Nodes: Distinguished from fork nodes, join nodes enable the convergence of data from multiple sources, consolidating the incoming data streams into a unified output. These nodes serve to synchronize and coordinate the flow of information within the activity diagram.

Notations in an Activity Diagram:

Initial State: Signifies the commencement or initial step of a process.

Final State: Represents the conclusion or endpoint of a process, indicating the completion of all associated activities.

Decision Box: Ensures that the progression of activities adheres to a predetermined course.

Action Box: Denotes the specific tasks or actions that need to be executed as part of the overall process.

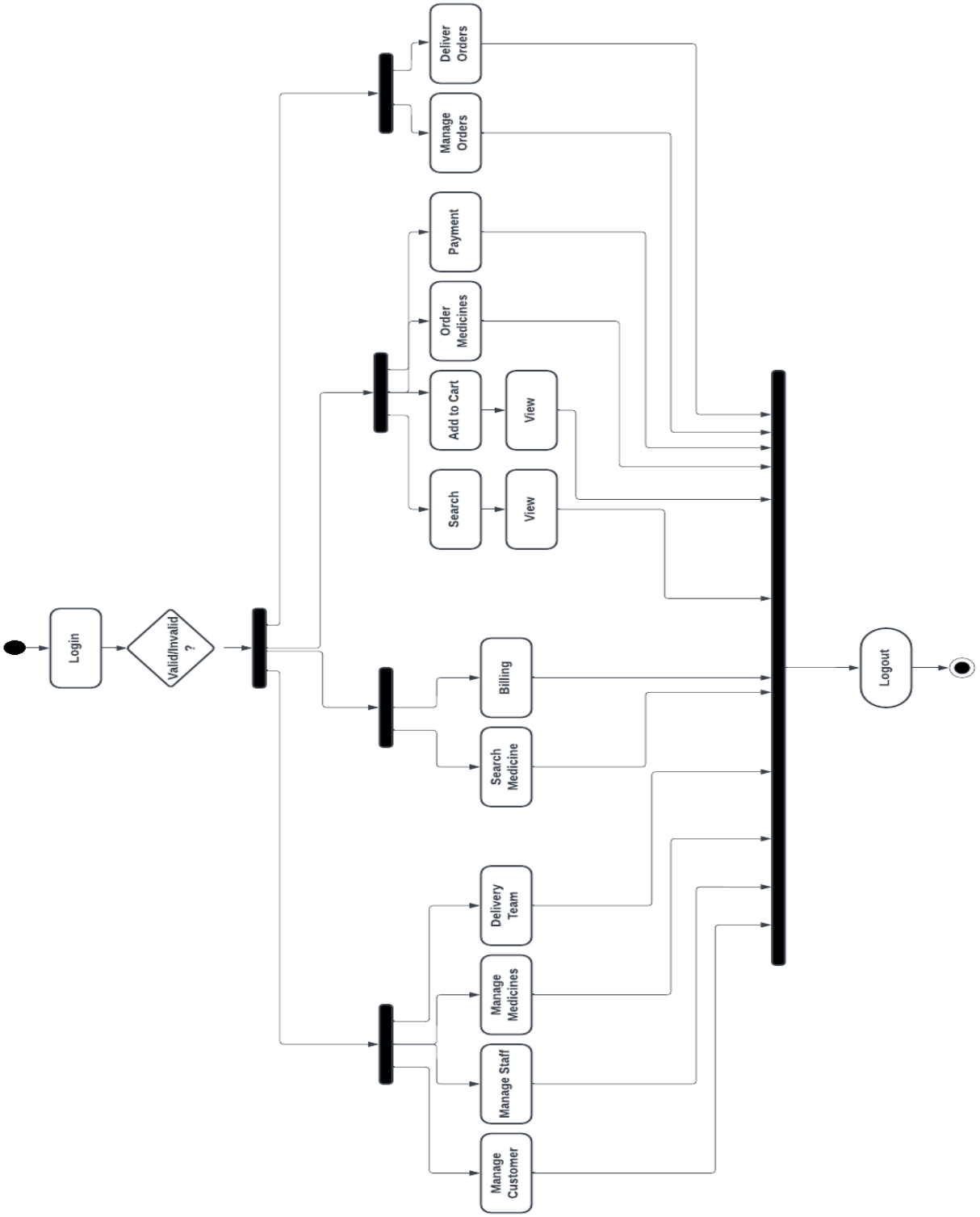


Fig4:Activity Diagram

4.2.3 Class Diagram

The class diagram is a static representation that illustrates the constituent elements and their associations within the system, providing a visual overview of its structure and components. It serves as a blueprint for organizing various attributes, relationships, and actions, streamlining the software development process by encapsulating critical information about the system's design and functionality. Essentially, a class diagram presents a holistic depiction of the system, offering insights into its composition, interdependencies, and behavior through a structured representation.

Components of a Class Diagram:

- **Upper Section**: This segment entails the class name, serving as a distinct identifier for a group of similar entities within the system. The class name is highlighted in bold letters, positioned centrally at the top. For an abstract class, the title is presented in slanted writing style, emphasizing its abstract nature.
- **Middle Section**: This section delineates the class attributes, demonstrating its properties and visibility factors. Attributes are depicted alongside their corresponding visibility indicators, such as public (+), private (-), protected (#), and package (~), reflecting the accessibility levels of each attribute.
- **Lower Section**: The lower segment encompasses the class methods or operations, outlined in a comprehensive list format. Each method is depicted as a discrete line item, emphasizing the interactions between the class and its associated data.
- Within the context of UML, the relationships depicted in the class diagram adhere to specific conventions:
- **Dependency**: Signifies the impact of one element's changes on another.
- **Generalization**: Illustrates the hierarchical relationship between classes, where one class acts as a parent while another assumes the role of the child.
- **Association**: Indicates the connections between different elements within the system.
- **Multiplicity**: Sets constraints on the permissible quantities associated with specific attributes or relationships.
- **Aggregation**: Represents a collection or group forming part of an association.
- **Composition**: Describes a specialized form of aggregation, emphasizing the interdependence between parent and child elements.

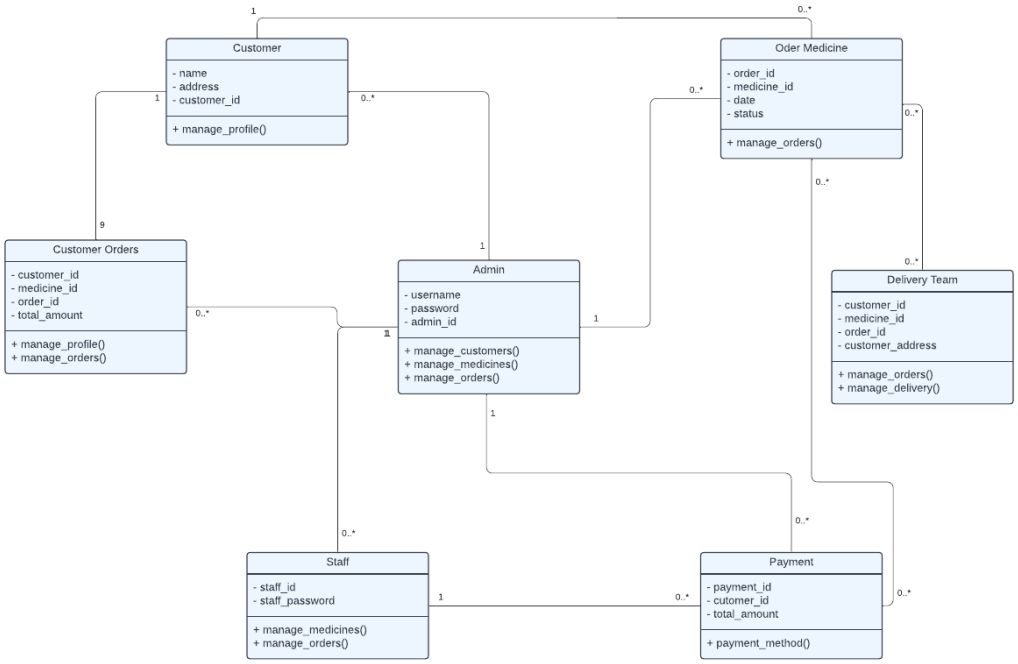


Fig5:Class Diagram

4.2.4 Object Diagram

Object diagrams are closely related to class diagrams and are derived from them. They offer a snapshot of a specific moment within an object-based system, illustrating a group of elements represented by a class. While they share similarities with class diagrams, object diagrams provide a more concrete depiction by showcasing specific instances of objects at a particular instant, enhancing the comprehension of the system's functionality and structure.

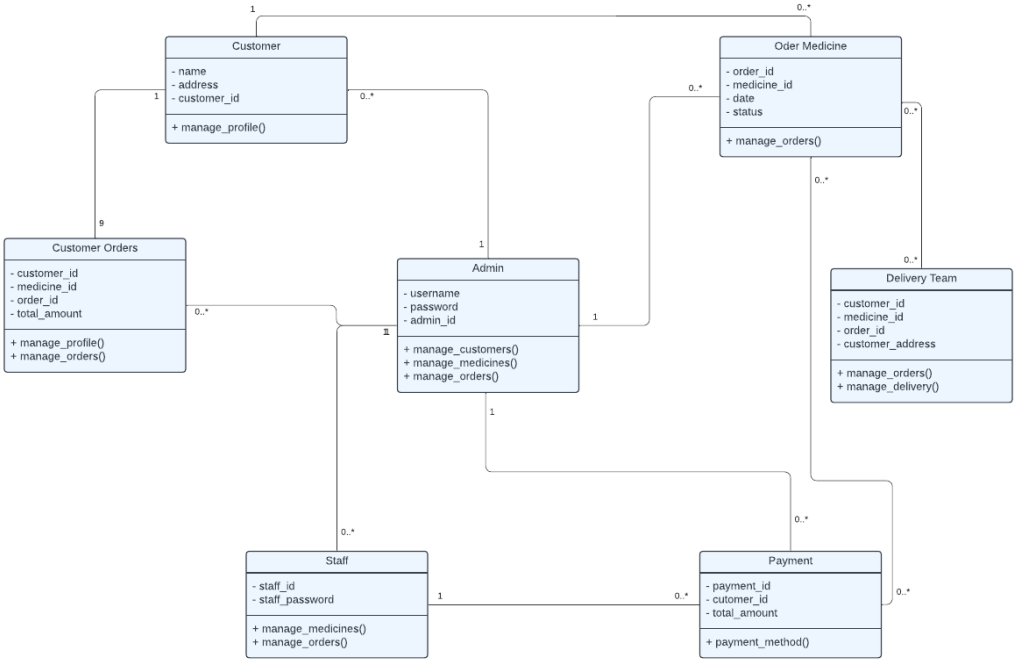


Fig6:Object Diagram

4.2.5 Component Diagram

A component diagram is instrumental in partitioning a complex system composed of objects into more manageable segments. It provides an overview of the internal components such as programs, documents, and tools within the node. By illustrating the connections and organization of elements in the system, it facilitates the creation of a usable system. Components are discreet sections of a system capable of autonomous function and conceal their internal operations, akin to a confidential box that operates only when used correctly.

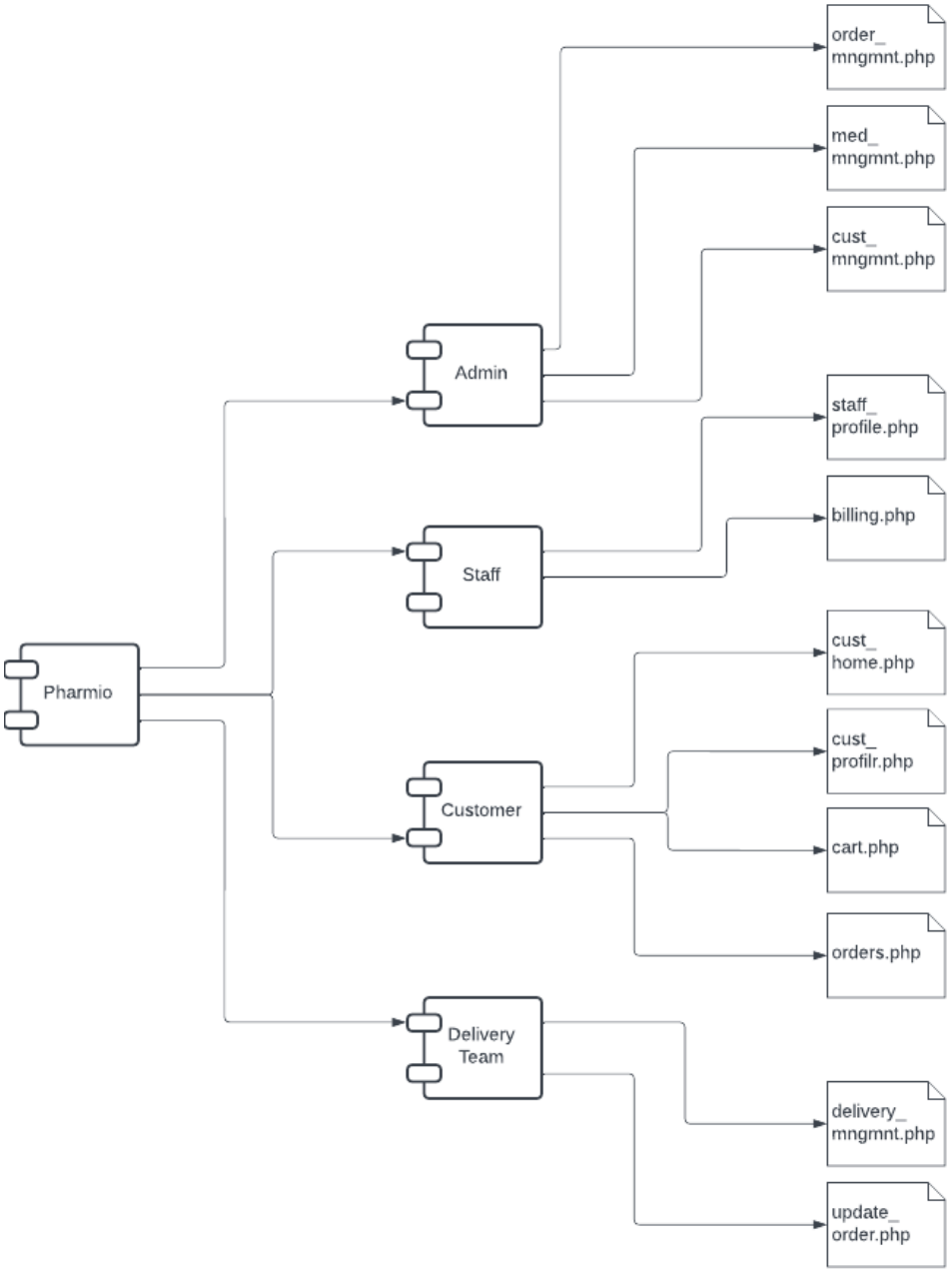
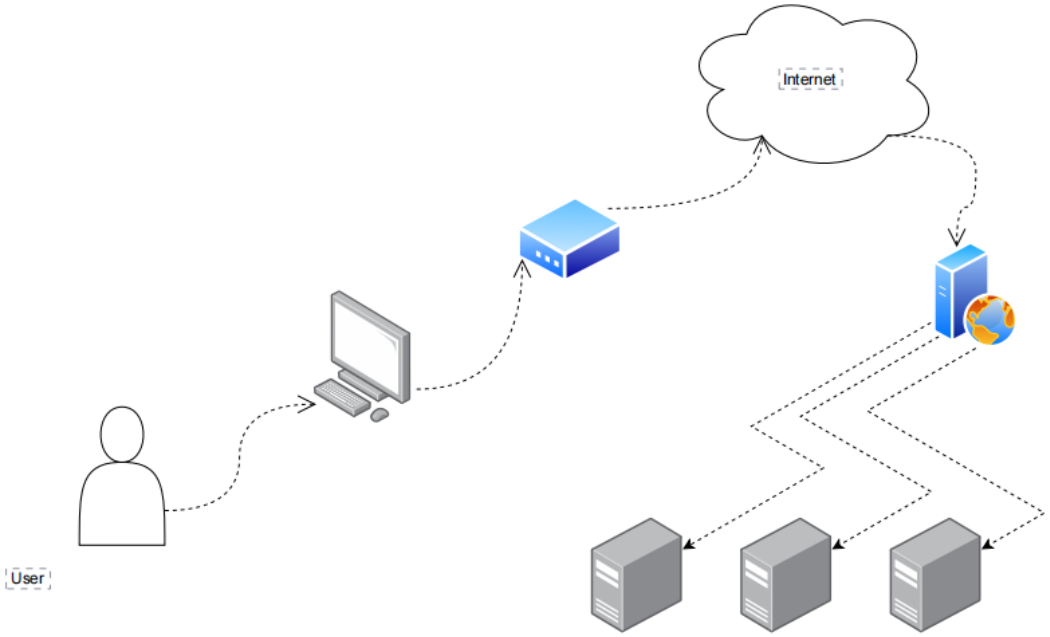


Fig7:Component Diagram

4.2.8 Deployment Diagram

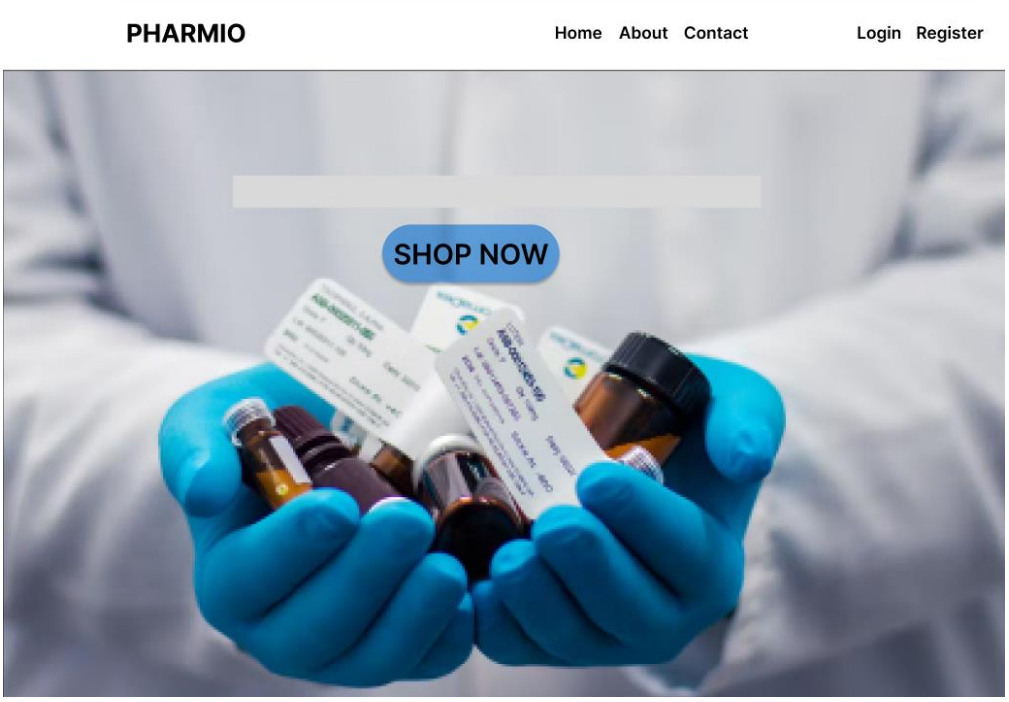
The deployment diagram visually represents the placement of software components within the physical computer or server. It provides insights into the static arrangement of the system's elements, including nodes and their interconnections. The diagram outlines the implementation of software on the computer system, detailing the architecture's composition and organization.



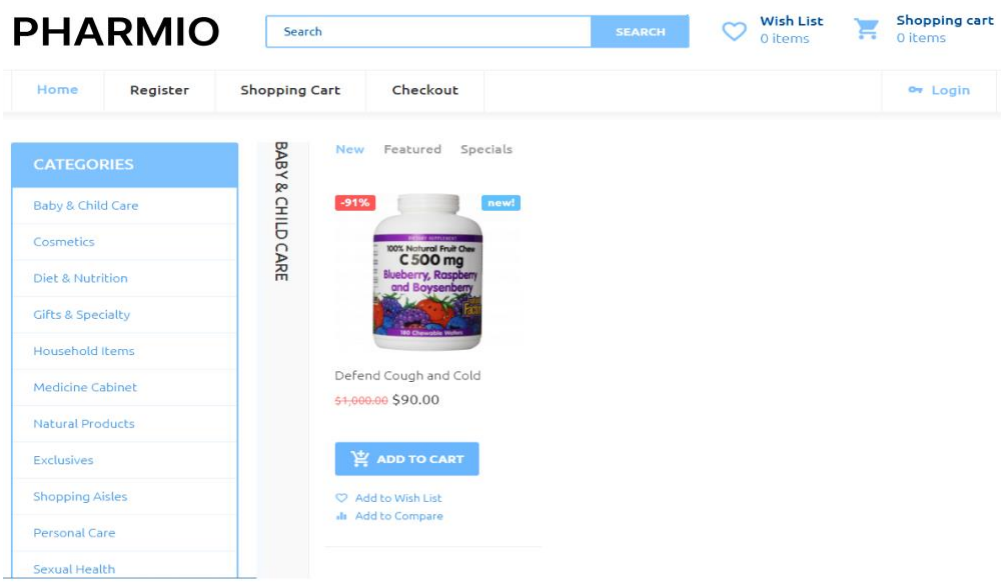
4.3 USER INTERFACE DESIGN USING FIGMA

Login page

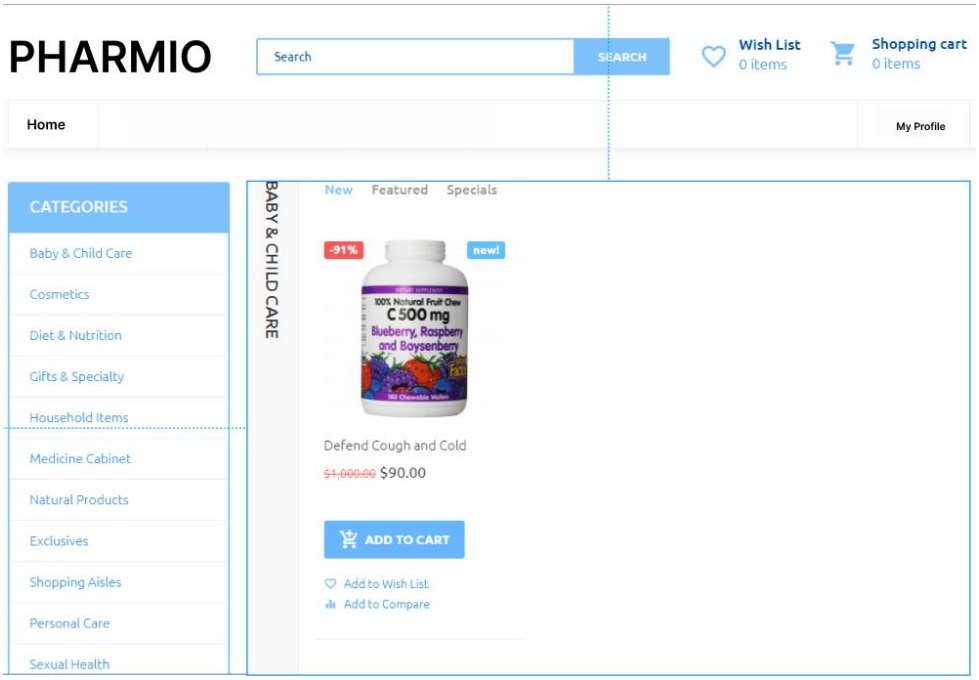
Home



Customer Page



Shopping Page logout



4.4 DATABASE DESIGN

4.4.1 Relational Database Management System (RDBMS)

- A relational database management system (RDBMS) is a structured system designed to efficiently store and retrieve information, ensuring its accessibility and security for users.
- Developing a database involves understanding user requirements and tailoring the database system to meet their specific needs.
- The process begins with creating an organizational plan for the data, independent of any particular computer program.
- Subsequently, this plan is translated into a specific design for the chosen computer program, focusing on the effective storage of information within the database system.
- Key aspects in this process include ensuring data integrity, maintaining the accuracy and consistency of data, and achieving data independence, allowing data to be accessed without being affected by changes in the database schema or structure.

4.4.2 Normalization

Normalization is the process of organizing a database into tables and ensuring that the data is efficiently stored and free from redundancy. It involves structuring the tables to minimize data duplication and dependency. Each table consists of rows and columns, where a row, also known as a tuple, represents a specific data set, and a column header, termed an attribute, identifies the type of data being stored. In the context of the relational model, a group of interrelated tables constitutes a relational database, facilitating effective management and utilization of data.

4.4.3 Sanitization

Sanitization in PHP refers to the process of cleansing and validating input data to ensure that it meets the expected format and is free from potentially harmful or unintended content. This is a crucial security measure to prevent various types of attacks, such as SQL injection, cross-site scripting (XSS), and other forms of code injection. PHP provides built-in functions and filters for sanitizing input, including filter_input(), filter_var(), and htmlspecialchars(). These tools help developers validate and clean user input by removing or encoding malicious characters, ensuring that data processed by the PHP application is safe and adheres to specified constraints. Implementing proper sanitization practices is essential for building robust and secure PHP applications, reducing the risk of vulnerabilities arising from untrusted input.

4.4.4 Indexing

Indexing involves organizing specific data or data groups in a database in a structured order based on their values. This ordered arrangement of index entries facilitates the quick and efficient retrieval of exact matches or data falling within a particular range. By using indexes, users can easily access information in a database without the need to search through every record during database operations. An index acts as a navigational tool for locating information within a database, enabling swift data lookup and facilitating the identification of records based on specific ordering. Additionally, an index can be established based on one or multiple columns within the table.

4.5 TABLE DESIGN

1.Table name: Role

Primary key: Role_id

Field Name	Datatype	Key Constraints	Description
Role_id	Int	Primary Key	Role id
Role	Varchar(10)	NOT NULL	Role of user

2.Table name: Customer_details

Primary key: Customer_id

Foreign key: Address_id references table address_details

Foreign key: Role_id references table Role

Field Name	Datatype	Key Constraints	Description
Customer_id	Varchar(10)	Primary Key	user id
Username	Varchar(10)	NOT NULL	Username
Customer_name	Varchar(200)	NOT NULL	Customer name
Dob	Varchar(20)	NOT NULL	Date of birth
Gender	Varchar(5)	NOT NULL	gender
Address_id	Varchar(25)	Foreign Key	Address
Password	Varchar(20)	NOT NULL	Password
Role_id	Int	Foreign Key	Role of user
Status	Binary	NOT NULL	Active or not

3.Table name: Address_details

Primary key: Address_id

Field Name	Datatype	Key Constraints	Description
Address_id	Varchar(10)	Primary Key	User id
Building/house	Varchar(100)	NOT NULL	Address line 1
Street	Varchar(100)	NOT NULL	Street
City	Varchar(100)	NOT NULL	City
District	Varchar(100)	NOT NULL	District
Pincode	**N**umber	Foreign Key	Pincode
Phone	**N**umber	NOT NULL	Phone number
Status	Binary	Foreign Key	Active or not

4.Table name: Brand_details

Primary key: Brand_id

Field Name	Datatype	Key Constraints	Description
Brand_id	Varchar(10)	Primary Key	Brand id
Brand_name	Varchar(50)	NOT NULL	Brand name
Brand_details	Varchar(200)	NOT NULL	Brand details
Status	Varchar(5)	NOT NULL	Status

5.Table name: Distributer_details

Primary key: Dis_id

Field Name	Datatype	Key Constraints	Description
Dis_id	Varchar(10)	Primary Key	Distributer id
Dis_name	Varchar(50)	NOT NULL	Distributer name
Address	Varchar(225)	Foreign Key	Address
Dis_email	Varchar(255)	NOT NULL	Distributer Email id

6.Table name: Medicine_details

Primary key: Med_id

Foreign key: Brand_id references table brand_details

Foreign key: Category_id references table category_details

Field Name	Datatype	Key Constraints	Description
Med_id	Varchar(10)	Primary Key	Medicine id
Med_name	Varchar(200)	NOT NULL	Medicine name
Brand_id	Varchar(10)	Foreign Key	Brand name
Batchno	Varchar(10)	NOT NULL	Batch number
Manuf_date	Date	NOT NULL	Manufacturing date
Exp_date	Date	NOT NULL	Date of expiry
Specification	Varchar(350)	NOT NULL	Specifications
Category_id	Varchar(10)	Foreign Key	Category id
Stock	Integer	NOT NULL	Available Stock
Price	Double	NOT NULL	Price
Status	Varchar(5)	NOT NULL	Status

7.Table name: Category_details

Primary key: Category_id

Field Name	Datatype	Key Constraints	Description
Category_id	Varchar(10)	Primary Key	Category id
Category_name	Varchar(100)	NOT NULL	Category name
Category_description	Varchar(350)	NOT NULL	Description

8.Table name: Sales_details

Primary key: Sales_id

Foreign key: Customer_id references table customer_details

Foreign key: Med_id references table medicine_details

Foreign key: Address_id references table address_details

Field Name	Datatype	Key Constraints	Description
Sales_id	Varchar(10)	Primary Key	Sales id
Customer_id	Varchar(10)	Foreign Key	Customer id
Address_id	Varchar(10)	Foreign Key	Address id
Med_id	Varchar(10)	Foreign Key	Medicine id
Quantity	Integer	NOT NULL	Quantity
Mode_of_sale	Binary	NOT NULL	Online(1)/Offline(0)
Unit_price	Double	NOT NULL	Unit price

Sales_date	Date	NOT NULL	Date of sale
Status	Varchar(5)	NOT NULL	Status

9.Table name: Wishlist_details

Primary key: Wish_id

Foreign key: Customer_id references table customer_details

Foreign key: Med_id references table medicine_details

Field Name	Datatype	Key Constraints	Description
Wish_id	Int	Primary Key	Wish id
Customer_id	Varchar(10)	Foreign Key	Customer id
Med_id	Varchar(10)	Foreign Key	Medicine id
Status	Binary	NOT NULL	Active or not

10.Table name: Cart_details

Primary key: Cart_id

Foreign key: Customer_id references table customer_details

Foreign key: Med_id references table medicine_details

Field Name	Datatype	Key Constraints	Description
Cart_id	Varchar(10)	Primary Key	Cart id
Customer_id	Varchar(10)	Foreign Key	Customer id
Med_id	Varchar(10)	Foreign Key	Medicine id
Quantity	Varchar(10)	NOT NULL	Quantity
Date	Date	NOTNULL	Date
Status	Binary	NOT NULL	Active or not

11.Table name: Payment_details

Primary key: Payment_id

Foreign key: Sales_id references table sales_details

Field Name	Datatype	Key Constraints	Description
Payment_id	Varchar(10)	Foriegn Key	Payment id
Sales_id	Varchar(10)	Foreign Key	Order id
Method	Varchar(30)	NOT NULL	Mode of payment
Payment_date	Date	NOT NULL	Date of payment

Status	Varchar(10)	NOT NULL	Status of payment

12.Table name: Delivery_details

Primary key: Delivery_id

Foreign key: Sales_id references table sales_details

Payment_id references table payment_details

Field Name	Datatype	Key Constraints	Description
Delivery_id	Varchar(10)	Primary Key	Delivery id
Sales_id	Varchar(10)	Foreign Key	Sales id
Payment_id	Date	Foreign Key	Payment id
Delivery_date	Date	NOT NULL	Date of sale
Status	Varchar(10)	NOT NULL	Status of delivery

CHAPTER 5
SYSTEM TESTING

5.1 INTRODUCTION

Software testing serves as a critical means of ensuring that a computer program operates in accordance with its intended functionality. It is employed to verify that the software performs as expected, meeting the specified requirements and standards. Validation, in this context, involves the thorough examination and testing of software to confirm that it aligns with the user's desired outcomes. Software testing is a comprehensive process that, alongside methods such as inspection and program walkthroughs, aims to identify potential errors and discrepancies within the software. The objectives of testing can be guided by several principles, including the execution of a program with the aim of uncovering errors, the creation of effective test cases, and the successful identification of previously undiscovered errors. When conducted successfully, testing can pinpoint flaws in the software, ensuring that the computer program operates efficiently and as intended. Additionally, three fundamental methods are employed for assessing a computer program: correctness, implementation efficiency, and computational complexity.

5.2 TEST PLAN

A test plan serves as a comprehensive set of guidelines for conducting various types of tests, functioning as a navigational tool that outlines specific steps to follow. Software developers create instructions for both program usage and the organization of essential information to ensure the program's seamless operation. They meticulously verify each component of the program to confirm its intended functionality. The responsibility of thoroughly testing the built software lies with the ITG (Information Technology Group), ensuring comprehensive and rigorous testing beyond the initial development phase. The objectives of testing should be well-defined and quantifiable, with the test plan encompassing details about the frequency of malfunctions, associated costs for rectification, occurrence rates, and the time required for comprehensive reevaluation.

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

Software components or modules, which are the smallest units of a software design, are tested through unit testing. Using the design guidance, test key control pathways in a module to identify issues. This implies the level of difficulty of the tests for each minor component of a program and the components that haven't been put to the test. One kind of testing called unit testing

examines the internal workings of the code and can include completed concurrently for several program components.

We must first determine whether data is correctly flowing between the various components of the computer software before beginning any more tests. All other checks are useless if the data isn't entering and exiting the system appropriately. When creating anything, it's critical to consider potential difficulties and develop a plan to address them. This could entail rerouting the procedure or ending it altogether.

5.2.2 Integration Testing

Integration testing is a technique for developing a program while simultaneously searching for errors in the interoperability of its many components. Using tested components to develop a software according to plan is the goal. To ensure proper operation, the entire software is tested. Errors are corrected, but then mistakes occur again, and so on. Following the inspection of every component of the system, the components were assembled to ensure optimal performance. Furthermore, they created uniform programs rather than varying them.

5.2.3 Validation Testing or System Testing

This concludes the testing phase. During this test, the entire system was examined to ensure that all of the various building blocks and instructions interacted as intended. This type of testing is known as system testing or black box testing. One technique to make sure the software performs as intended is to use black box testing. Through the use of several input formats, black box testing assists software engineers in identifying every issue present in a program. Black box testing examines code for errors in startup and termination, performance, data access, functions, and interfaces.

5.2.4 Output Testing or User Acceptance Testing

The system is being evaluated to see if users like it and if it satisfies the business' requirements. While it is being developed or updated, the computer program must remain connected to the user.

The following factors are taken into consideration:

- Input screen designs
- Output screen designs

The aforementioned is tested using various types of data. For system testing, having the test data ready is crucial. After gathering test data, they use that information to check the system they are

researching. We look for errors in the system and use the procedures we are already familiar with to find and correct them. In order to use these fixes in the future, we keep a record of them.

5.2.5 Automation Testing

Before a piece of software is officially used, automated testing determines whether it functions properly and complies with standards. This testing technique makes use of tools that execute written instructions. When a computer system employs a specialized tool to test things automatically, this is known as UI automation testing. We write scripts that perform this task automatically for each different test instead of relying on users to click around the application to ensure everything is in working order. When you verify information and then add it, there are a few things you should do. When you need to run the same test simultaneously on numerous computers, automatic testing is required.

5.2.6 Selenium Testing

Selenium is a free and practical tool that automatically tests websites. It's critical that web developers are aware of it. Selenium automation testing is the practice of testing things using the Selenium tool. Selenium is a collection of tools that each perform unique tasks for automation testing, not just one single tool. Although manual testing is a crucial step in the development of an application, it has flaws. Its potential to be monotonous and repetitive is a major issue. Jason Huggins, a ThoughtWorks employee, developed a method to automatically test things rather than doing it by hand to make things simpler. He created a program called the JavaScriptTestRunner to aid in automatically testing websites. The program's name was changed to Selenium in 2004.

Test Case 1

```
package definition;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import io.cucumber.java.en.*;
public class loginsteps {
   WebDriver driver;
   @Given("^the user is on the login page$")
   public void userIsOnLoginPage()throws Exception {
      // Provide the path to your ChromeDriver executable
      System.setProperty("webdriver.chrome.driver",
     "src/test/resources/driver/chromedriver.exe");
```

```
        driver = new ChromeDriver();
        // Navigate to the login page
        driver.get("http://localhost/project/login.php");
        Thread.sleep(2000);
    }
    @When("^the user enters valid credentials$")
    public void userEntersValidCredentials() throws Throwable{
        // Enter valid credentials
            System.out.println("Given: the user is on the login page");
              System.out.println("When: the user enters valid credentials");
        driver.findElement(By.id("username")).sendKeys("fathimahazbinr2024a@mca.ajce.in");
        driver.findElement(By.id("password")).sendKeys("Hazbi@12345");
    }
    @And("^clicks the login button$")
    public void userClicksLoginButton() {
        // Click the login button
        driver.findElement(By.id("login-btn")).click();
    }
    @Then("^the user should be redirected to the dashboard page$")
    public void userIsRedirectedToDashboardPage()throws Exception {
            System.out.println("Then: the user should be redirected to the dashboard page");
            System.out.println("Test Passed");
        // Verify redirection to the dashboard page
        // You might use assertions or other verification techniques here
        // Example: Assert.assertTrue(driver.getCurrentUrl().contains("dashboard"));
            Thread.sleep(2000);
            driver.close();
            driver.quit();
    }
}
```

Screenshot

```
  Then the user should be redirected to the dashboard page # definition.loginsteps.userIsRedirectedToDashboardPage()

1 Scenarios (1 passed)
4 Steps (4 passed)
0m20.919s
```

Test Case 1					
Project Name: Pharmio					
Login Test Case					
Test Case ID: Test_1			**Test Designed By: Fathima Hazbin R**		
Test Priority(Low/Medium/High):High			**Test Designed Date: 03/12/2023**		
Module Name:Login Screen			**Test Executed By: Ms. Anit James**		
Test Title: Login			**Test Execution Date: 04/12/2023**		
Description: Testing the login Page					
Pre-Condition : User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigate to login page		User should be able to login	User is navigated to home page with successful login.	Passs
2	Provide Valid email	fathimahazbinr2024a@mca.ajce.in			
3	Provide valid password	Hazbi@12345			
4	Click on Login button				
Post-Condition: User is navigated to Home page.					

Test Case 2

package definition;

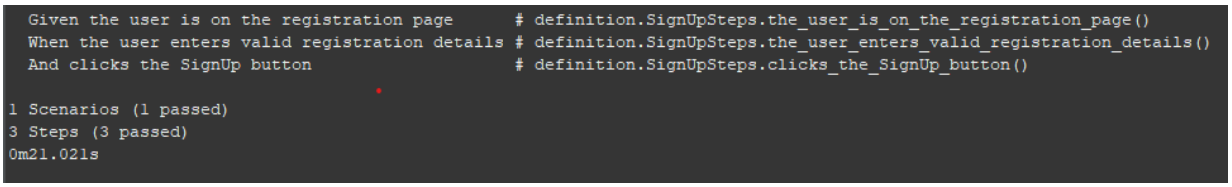
import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

```
import org.openqa.selenium.chrome.ChromeDriver;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
public class SignUpSteps {
   private WebDriver driver = new ChromeDriver();
   @Given("^the user is on the registration page$")
   public void the_user_is_on_the_registration_page() {
         // Provide the path to your ChromeDriver executable
   System.setProperty("webdriver.chrome.driver",
"src/test/resources/driver/chromedriver.exe");
   //Initialize the WebDriver
    driver = new ChromeDriver();
      driver.get("http://localhost/project/register.php");
   }
   @When("^the user enters valid registration details$")
   public void the_user_enters_valid_registration_details() {
      enterText("email", "test@example.com");
      enterText("fname", "John");
      enterText("lname", "Doe");
      enterText("password", "password123");
      enterText("confirm_password", "password123");
   }
   @When("^clicks the SignUp button$")
   public void clicks_the_SignUp_button() {
      WebElement submitButton = driver.findElement(By.name("submit"));
      submitButton.click();
   }
   private void enterText(String elementId, String text) {
      WebElement element = driver.findElement(By.id(elementId));
      element.sendKeys(text);
   }
}
```

Screenshot



Test Case 2					
Project Name: Pharmio					
SignUp Test Case					
Test Case ID: Test_2			**Test Designed By: Fathima Hazbin R**		
Test Priority(Low/Medium/High):High			**Test Designed Date: 03/12/2023**		
Module Name: Registration			**Test Executed By: Ms. Anit James**		
Test Title: SignUp			**Test Execution Date: 04/12/2023**		
Description: Testing the registration process					
Pre-Condition : User should have valid email.					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigate to SignUp page		User should be able to Register	User is navigated to login after successful registration.	Passs
2	User Enters Vald registration details.	Email:fathimahazbinr@gmail.com Fname:Fathima Lname:Hazbin Password:password@123 Confirm password:password@123			
3	Click on Signup Button				
Post-Condition: User registered successfully and redirected to login page.					

Test Case 3

package definition;

import org.openqa.selenium.WebElement;

```
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import static org.junit.Assert.assertTrue;
import java.io.File;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
import io.cucumber.java.en.Then;
public class imageupload {
    private WebDriver driver;
    @Given("^the admin is on the login page$")
    public void the_admin_is_on_the_login_page() {
        System.setProperty("webdriver.chrome.driver", "src/test/resources/driver/chromedriver.exe");
        driver = new ChromeDriver();
        driver.get("http://localhost/project/login.php");
        System.out.println("Given: the user is on the login page");
         }
    @When("^the admin logs in with valid credentials$")
    public void the_admin_logs_in_with_valid_credentials() {
        WebElement usernameInput = driver.findElement(By.id("username"));
        WebElement passwordInput = driver.findElement(By.id("password"));
        usernameInput.sendKeys("admin");
        passwordInput.sendKeys("Admin@12345");
        WebElement loginButton = driver.findElement(By.id("login-btn"));
        loginButton.click();
    @When("^the admin is on the add medicine page$")
    public void the_admin_is_on_the_add_medicine_page() {
        driver.get("http://localhost/project/add_medicine.php");
        System.out.println("And: the admin is on the add medicine page");}
    @When("^the admin fills out the medicine form$")
    public void the_admin_fills_out_the_medicine_form() {
        WebElement medNameInput = driver.findElement(By.id("med_id"));
        medNameInput.sendKeys("Sample Medicine");
        WebElement genericNameInput = driver.findElement(By.id("generic_name"));
```

```
    genericNameInput.sendKeys("Generic Sample");
    WebElement brandIdDropdown = driver.findElement(By.id("brand_id"));
    selectDropdownOptionByVisibleText(brandIdDropdown, "Panadol");
    WebElement batchNoInput = driver.findElement(By.id("batchno"));
    batchNoInput.sendKeys("12345");
    WebElement manufDateInput = driver.findElement(By.id("manuf_date"));
    manufDateInput.sendKeys("2023-01-01");
    WebElement expDateInput = driver.findElement(By.id("exp_date"));
    expDateInput.sendKeys("2023-12-31");
    WebElement specificationInput = driver.findElement(By.id("specification"));
    specificationInput.sendKeys("Sample specifications");
    WebElement categoryIdDropdown = driver.findElement(By.id("category_id"));
    selectDropdownOptionByVisibleText(categoryIdDropdown, "Pain Relief");
    WebElement stockInput = driver.findElement(By.id("stock"));
    stockInput.sendKeys("100");
    WebElement priceInput = driver.findElement(By.id("price"));
    priceInput.sendKeys("19.99");
    WebElement imagesInput = driver.findElement(By.id("images"));
    String relativePath = "src/test/resources/images/Picture1.png";
    String absolutePath = new File(relativePath).getAbsolutePath();
    imagesInput.sendKeys(absolutePath);
    System.out.println("And :the admin fills out the medicine form");}
@When("^the admin submits the form$")
public void the_admin_submits_the_form() {
    WebElement submitButton = driver.findElement(By.id("submit"));
    submitButton.click();
    System.out.println("And :the admin submits the form");}
@Then("^verify that the medicine is added successfully$")
public void verify_that_the_medicine_is_added_successfully() {
    WebElement successMessage = driver.findElement(By.id("successMessage"));
    assertTrue(successMessage.isDisplayed());
    driver.quit();
    System.out.println("Then :verify that the medicine is added successfully");
private void selectDropdownOptionByVisibleText(WebElement dropdown, String optionText) {
    new WebDriverWait(driver, 10)
            .until(ExpectedConditions.elementToBeClickable(dropdown));
    Select select = new Select(dropdown);
```

```
        System.out.println("Dropdown options: " + select.getOptions());
        new WebDriverWait(driver, 10)
                .until(ExpectedConditions.numberOfElementsToBeMoreThan(
                        By.tagName("option"), 1));
        select.selectByVisibleText(optionText);
}
```

Screenshot

```
WARNING: Unable to find an exact match for CDP version 120, returning the closest version; found: 119;
When: the user enters valid credentials and logs in
And: the admin is on the add medicine page
And :the admin fills out the medicine form
And :the admin submits the form
Then :verify that the medicine is added successfully
Then: the medicine added successfully
Test Passed
```

Test Case 3					
Project Name: Pharmio					
Medicine Adding Test Case					
Test Case ID: Test_3			**Test Designed By: Fathima Hazbin R**		
Test Priority(Low/Medium/High):High			**Test Designed Date: 03/12/2023**		
Module Name: Admin			**Test Executed By: Ms. Anit James**		
Test Title: Add Medicine			**Test Execution Date: 04/12/2023**		
Description: Testing the add medicine process					
Pre-Condition : User should have valid email.					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigate to login page		User should be able to login	User is navigated to home page with successful login.	Pass
2	Provide Valid email	Username: admin password: Admin@12345			
3	Admin navigates to medicine form	Sample Medicine Generic Sample			

		Panadol 12345 2023-01-01 2023-12-31 Sample specifications Pain Relief 100 19.99			
4	Admin clicks the submit button.				
Post-Condition: Admin added medicine Succesfully					

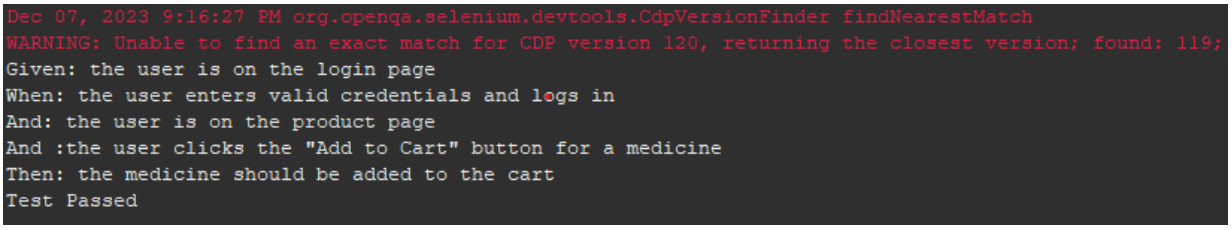
Test Case 4

```
package definition;
import static org.junit.Assert.assertTrue;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
import io.cucumber.java.en.Then;
public class AddToCartSteps {
    private WebDriver driver;
    @Given("^the user is on the login page for Add to Cart$")
    public void userIsInLoginPageForAddToCart() {
        setupWebDriver();
        driver.get("http://localhost/project/login.php");
        System.out.println("Given: the user is on the login page");
```

```
    }
    @Given("^the user enters valid credentials and logs in for Add to Cart$")
    public void givenTheUserIsInTheLoginPageForAddToCart() {
        WebElement usernameInput = driver.findElement(By.id("username"));
        WebElement passwordInput = driver.findElement(By.id("password"));
        usernameInput.sendKeys("fathimahazbinr2024a@mca.ajce.in");
        passwordInput.sendKeys("Hazbi@12345");
        WebElement loginButton = driver.findElement(By.id("login-btn"));
        loginButton.click();
            System.out.println("When: the user enters valid credentials and logs in");
    }
    @Given("^the user is on the product page for Add to Cart$")
    public void userIsInCutomerpageForAddToCart() {
        driver.get("http://localhost/project/customer.php");
        System.out.println("And: the user is on the product page");
    }
    @When("^the user clicks the \"Add to Cart\" button for a medicine for Add to Cart$")
    public void whenUserClicksAddToCartButton() {
        WebElement addToCartButton = driver.findElement(By.cssSelector(".cart-button"));
        addToCartButton.click();
        System.out.println("And: the user clicks the \"Add to Cart\" button for a medicine");
    }
    @Then("^the medicine should be added to the cart for Add to Cart$")
    public void thenMedicineShouldBeAddedToTheCart() {
        WebElement successMessage = driver.findElement(By.cssSelector(".success-message"));
        assertTrue(successMessage.isDisplayed());
        System.out.println("Then: the medicine should be added to the cart");
        System.out.println("Test Passed");
    }
    private void setupWebDriver() {
        System.setProperty("webdriver.chrome.driver",
"src/test/resources/driver/chromedriver.exe");
        driver = new ChromeDriver();
    }
```

```
public void tearDown() {
    if (driver != null) {
        driver.quit();
    }}}
```

Screenshot



Test Case 4					
Project Name: Pharmio					
Add to Cart Test Case					
Test Case ID: Test_4			**Test Designed By: Fathima Hazbin R**		
Test Priority(Low/Medium/High):High			**Test Designed Date: 03/12/2023**		
Module Name: Customer			**Test Executed By: Ms. Anit James**		
Test Title: Add to cart.			**Test Execution Date: 04/12/2023**		
Description: Testing whether medicine add to cart.					
Pre-Condition : User should have valid email.					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Navigate to login page	fathimahazbinr2024a@mca.ajce.in password: Hazbin@1234	User should be able to login	User is navigated to home page with successful login.	Pass
2	User navigates to shopping page.				
3	User clicks Add to Cart.		Message added should be alerted	Message added alert box appears.	
4	Medicine added to cart.	Medicine name:Panadol Quantity : 1			
Post-Condition: Medicine added to cart.					

CHAPTER 6

IMPLEMENTATION

6.1INTRODUCTION

Implementation is the crucial phase during which the planned system materializes into a tangible, operational entity. Building user trust and confidence in the newly developed system is vital for its successful integration. The focus lies on user training and the creation of supportive resources. Conversion typically occurs during or after user training. In essence, implementation signifies the transition from a conceptual system design to its practical, functional existence. It involves the process of replacing or modifying existing systems to ensure a seamless shift towards the new system. Thorough planning, comprehensive system analysis, and the formulation of effective transition strategies are integral to the implementation process.

6.2 IMPLEMENTATION PROCEDURES

During the software implementation process, the software is installed in its intended environment and functions as expected. In certain organizations, an individual not directly involved in software usage oversees and greenlights the project's development. Initial hesitations may arise, and it is crucial to address any uncertainties before they escalate into formidable resistance. It is essential to demonstrate to users why the new system is superior to its predecessor and to instill a sense of trust in the software. Users should receive comprehensive training to ensure their comfort and confidence with the application. To evaluate the outcome, one must ensure that the server program is operational on the server before assessing the system's functionality.

6.2.1 User Training

User training is essential for imparting the necessary skills to individuals in utilizing and adapting to the system. It is imperative that users feel at ease and confident in their ability to navigate and operate the computer system effectively. Particularly when faced with complex functionalities, the significance of comprehensive user training becomes even more pronounced. This training equips users with the ability to input information, manage errors, query the database, and utilize various tools for generating reports and accomplishing other critical tasks.

6.2.2 Training on the Application Software

Training on the application software involves an in-depth understanding of how to operate a new program after familiarizing oneself with the fundamentals of computer usage. This training aims to provide comprehensive guidance on navigating the new system, encompassing the utilization of various screens, accessing support resources, troubleshooting mistakes, and rectifying errors. Its primary objective is to equip users with the requisite knowledge and skills to effectively engage with the system, tailored to the specific needs and roles of different user groups.

6.2.3 System Maintenance

System maintenance is a complex challenge within the sphere of system development. It encompasses vital tasks and ensures the effective functioning of software during the maintenance phase of the software life cycle. Once a system is operational, it requires ongoing attention to ensure its continued smooth operation. Prioritizing software maintenance throughout the development process is essential to enable the system to adapt to environmental changes. The practice of software maintenance extends beyond mere error detection within the code.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

Pharmio emerges as a pioneering online pharmacy store, offering a transformative solution that redefines the traditional pharmacy experience. With a user-friendly interface and a secure system design, Pharmio facilitates convenient access to pharmaceuticals and healthcare essentials, transcending the limitations of brick-and-mortar establishments. Rigorous testing ensures compliance with industry standards, safeguarding user data and privacy. The platform's commitment to ongoing enhancements underscores its adaptability and responsiveness to the ever-evolving landscape of online healthcare services. Pharmio not only represents a significant leap in accessibility and efficiency but also serves as a beacon of innovation, promising a continuous evolution to meet the dynamic demands of the digital pharmacy market.

7.2 FUTURE SCOPE

Looking forward, Pharmio aims to improve your experience by exploring additional features such as an expanded product range, including various health and wellness items beyond medicines. The project envisions strategic collaborations with healthcare providers to enhance its services, providing a comprehensive solution for users. Pharmio is committed to maintaining user-friendly interfaces and prioritizing the security of your information. In the future, the focus will be on continuous refinement, adapting to changing healthcare trends to ensure Pharmio remains a reliable and convenient platform for your health needs.

Future initiatives include:

- Diversified Offerring: Introducing a broad spectrum of health and wellness products.
- Strategic Collaboration: Partnering with healthcare providers for enhanced services.
- User-Friendly Design: Maintaining easy-to-use interfaces.
- Data Security: Ensuring the utmost protection of user information.
- Continuous Adaptation: Staying current with evolving healthcare trends for user-centric solutions.

CHAPTER 8
BIBLIOGRAPHY

REFERENCES:

- "Designing Data-Intensive Applications" by Martin Kleppmann: This book covers the design and development of large-scale data systems, including data modeling, storage, and processing.
- "Clean Architecture: A Craftsman's Guide to Software Structure and Design" by Robert C. Martin: This book provides guidance on designing software systems that are modular, maintainable, and scalable.
- “The Pragmatic Programmer: From Journeyman to Master" by Andrew Hunt and David Thomas: This classic book provides practical advice on software development, including design principles, testing, and code organization.
- "Refactoring: Improving the Design of Existing Code" by Martin Fowler: This book provides a systematic approach to improving the design of existing software systems through refactoring.

WEBSITES:

- http://www.w3schools.com/
- www.w3schools.com
- getbootstrap.com
- www.geeksforgeeks.com
- ChatGPT (openai.com)

CHAPTER 9

APPENDIX

9.1 Sample Code

Admin Pages

```
<?php
include('db_config.php');
session_start();

if (!isset($_SESSION['admin'])) {
    header('Location: login.php'); // Redirect to admin login page if not logged in as admin
    exit();
}
?>

<!DOCTYPE html>
<html>
<head>
  <!-- Basic -->
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <!-- Mobile Metas -->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
  <!-- Site Metas -->
  <meta name="keywords" content="" />
  <meta name="description" content="" />
  <meta name="author" content="" />
  <link rel="shortcut icon" href="images/favicon.png" type="">

  <meta http-equiv="Cache-Control" content="no-cache, no-store, must-revalidate">
    <meta http-equiv="Pragma" content="no-cache">
    <meta http-equiv="Expires" content="0">

  <title> Pharmio </title>

  <!-- bootstrap core css -->
  <link rel="stylesheet" type="text/css" href="css/bootstrap.css"/>

  <!-- fonts style -->
  <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500;700;900&display=swap"
rel="stylesheet">

  <!--owl slider stylesheet -->
  <link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.carousel.min.css" />

  <!-- font awesome style -->
  <link href="css/font-awesome.min.css" rel="stylesheet" />

  <!-- Custom styles for this template -->
  <link href="css/style1.css" rel="stylesheet" />
  <!-- responsive style -->
  <link href="css/responsive.css" rel="stylesheet" />
  <style>
     body {
        font-family: Arial, sans-serif;
      }

       .container {
```

```
      max-width: 800px;
      margin: 0 auto;
      padding: 20px;
      margin-left: 320px;
    }

    h1 {
      text-align: center;
    }

    .logout {
      float: right;
    }

/* Sidebar Styles */
.sidebar {
    height: 100%;
    width: 250px;
    position: fixed;
    top: 76px;
    left: 0;
    background-color: #f2f2f2;
    padding-top: 20px;
}

.sidebar h1 {
    text-align: left;
    margin-left: 13px;
    font-weight: bold;
    font-size: 30px;
}

.sidebar a.menu-item {
    padding: 8px 16px;
    text-decoration: none;
    font-size: 18px;
    color: #333;
    display: block;
    transition: 0.3s;
}

.sidebar .submenu {
    padding-left: 20px;
    display: none;
}

.sidebar .submenu a.sub-item {
    padding: 8px 0;
    color: #555;
}

.sidebar a.menu-item.active {
    background-color: #ddd;
}
```

```
  .dashboard-link {
    text-decoration: none;
    color: inherit;
  }

  .logout-button {
    background-color: #007bff;
    color: white;
    border: none;
    padding: 10px 20px;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s ease;
  }

  .logout-button:hover {
    background-color: #0056b3;
  }

  .welcome-message {
      text-align: left;
      margin-left:260px;
      margin-top: 25px;
      font-family: 'Roboto', sans-serif;
  }

  .welcome-message h2 {
      font-size: 36px;
      color: #000;
      margin-bottom: 10px;
  }

  .welcome-message p {
      font-size: 18px;
      color: #333;
  }

</style>

</head>

<body>

  <div class="hero_area">

  <div class="hero_bg_box">
      <div class="bg_img_box">
        <img src="" width="100%" height="100%">
      </div>
    </div>

    <!-- header section strats -->
    <header class="header_section">
      <div class="container-fluid">
        <nav class="navbar navbar-expand-lg custom_nav-container ">
```

```
        <a class="navbar-brand" href="admin.php">
          <span>
            PHARMIO ADMIN
          </span>
        </a>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
          <ul class="navbar-nav  ">
            <li class="nav-item">
              <a class="nav-link" href="logout.php"> <i class="fa fa-user" aria-hidden="true"></i>
LogOut</a>
            </li>

          </ul>
        </div>
      </nav>
    </div>
  </header>
  <!-- Sidebar -->
  <div class="sidebar">
    <h1>Dashboard</h1>
    <a href="#" class="menu-item">Customer</a>
    <div class="submenu">
      <a href="" class="sub-item">View Customers</a><br>
      <a href="customer_view.php" class="sub-item">Manage Customers</a>
    </div>
    <a href="#" class="menu-item">Staff</a>
    <div class="submenu">
      <a href="" class="sub-item">View Staffs</a><br>
      <a href="staff_view.php" class="sub-item">Manage Staff</a>
    </div>
    <a href="#" class="menu-item">Medicines</a>
    <div class="submenu">
      <a href="medicine_view.php" class="sub-item">Manage Medicines</a><br>
      <a href="medicine_category.php" class="sub-item">Manage Categories</a><br>
      <a href="medicine_brands.php" class="sub-item">Manage Brands</a><br>
    </div>
  </div>
  <!-- Content Section -->
  <div class="content">
    <div class="welcome-message">
      <h2>Welcome Admin!</h2>
      <p>Manage your customers, staff, and medicines efficiently.</p>
    </div>
  </div>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script>
    document.addEventListener("DOMContentLoaded", function() {
      var menuItems = document.querySelectorAll(".menu-item");
      menuItems.forEach(function(item) {
        item.addEventListener("click", function() {
          // Toggle submenu for the clicked item
          var submenu = item.nextElementSibling;
          if (submenu) {
            if (submenu.style.display === "block") {
              submenu.style.display = "none";
```

```
                item.classList.remove("active");
            } else {
                // Hide all submenus
                var submenus = document.querySelectorAll(".submenu");
                submenus.forEach(function(submenu) {
                    submenu.style.display = "none";
                });

                submenu.style.display = "block";
                // Toggle active class for menu items
                menuItems.forEach(function(menuItem) {
                    menuItem.classList.remove("active");
                });
                item.classList.add("active");
            }
        }
    });
  });
 });
</script>
</body>
</html>
```

Customer

```
<?php
    include('db_config.php');
    session_start();

    if (!isset($_SESSION['username'])) {
        header('Location: login.php');
        exit();
    }

    // Assuming your customer_details table has a 'first_name' column
    $sql = "SELECT first_name FROM customer_details WHERE username =
'{$_SESSION['username']}'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        $row = $result->fetch_assoc();
        $_SESSION['first_name'] = $row['first_name'];
    }
?>
<!DOCTYPE html>
<html>
<head>
  <!-- Basic -->
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <!-- Mobile Metas -->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
  <!-- Site Metas -->
  <meta name="keywords" content="" />
  <meta name="description" content="" />
```

```
  <meta name="author" content="" />
  <link rel="shortcut icon" href="images/favicon.png" type="">

  <meta http-equiv="Cache-Control" content="no-cache, no-store, must-revalidate">
  <meta http-equiv="Pragma" content="no-cache">
  <meta http-equiv="Expires" content="0">

  <title> Pharmio </title>

  <!-- bootstrap core css -->
  <link rel="stylesheet" type="text/css" href="css/bootstrap.css"/>

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"
/>

  <!-- fonts style -->
  <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500;700;900&display=swap"
rel="stylesheet">

  <!--owl slider stylesheet -->
  <link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.carousel.min.css" />

  <!-- font awesome style -->
  <link href="css/font-awesome.min.css" rel="stylesheet" />

  <!-- Custom styles for this template -->
  <link href="css/style1.css" rel="stylesheet" />
  <!-- responsive style -->
  <link href="css/responsive.css" rel="stylesheet" />
  <style>
    .logout-button {
       background-color: #007bff;
       color: white;
       border: none;
       padding: 10px 20px;
       border-radius: 5px;
       cursor: pointer;
       transition: background-color 0.3s ease;
    }

    .logout-button:hover {
       background-color: #0056b3;
    }

    .product-list {
       display: flex;
       flex-wrap: wrap;
       justify-content: flex-start;
       padding: 10px;
    }

    .product-grid {
       margin-left: 250px; /* Set margin equal to the width of the sidebar */
       display: flex;
```

```
    flex-wrap: wrap;
    justify-content: flex-start;
    padding: 20px;
}

.product-card {
    /* width: 340px; */
    border: 1px solid #ddd;
    border-radius: 8px;
    padding: 30px;
    margin: 10px;
    text-align: center;
    overflow-wrap: break-word;
    white-space: normal;
    position: relative;
}

.product-card img {
    max-width: 100%;
    height: auto;
    margin-bottom: 10px;
}

.product-card h3 {
    font-size: 18px;
    margin-bottom: 10px;
}

.product-card p {
    font-size: 14px;
    margin-bottom: 8px;
}

.product-card button {
    background-color: #000;
    color: #fff;
    border: none;
    padding: 10px 20px;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s ease;
    font-size:14px
}

.product-card button:hover {
    background-color: #333;
}

.product-header {
    position: absolute;
    top: 10px;
    right: 10px;
}

.favorites-icon {
```

```
    font-size: 24px;
    color: #ccc; /* Default color for unfilled heart */
    cursor: pointer;
    top: 10px; /* Adjust the top position as needed */
    right: 20px;
}

.favorites-icon.fas {
    color: black; /* Color for filled heart */
}

.cart-button,
.buy-now-button {
    width: 100%;
    margin-top: 10px;
}

.submenu1 {
    display: none;
    position: absolute;
    background-color: #ffffff;
    box-shadow: 0px 8px 16px 0px rgba(0, 0, 0, 0.2);
    z-index: 1;
    list-style: none;
    padding: 0;
    text-align: left;
}

.nav-item:hover .submenu1 {
    display: block;
}

.submenu1 .nav-item {
    width: 200px;
    padding: 10px;
}

.submenu1 .nav-link {
    color: #000; /* Set text color to black */
    text-decoration: none;
    display: block;
    padding: 8px 16px; /* Add padding for better spacing */
    font-size: 14px; /* Reduce font size */
}

.submenu1 .nav-link:hover {
    background-color: #f2f2f2;
}

.sidebar {
    background-color: #f9f9f9;
    color: #fff;
    width: 250px;
    height: 100vh;
    padding: 20px;
```

```
    position: fixed;
    left: 0;
    top: 76px;
    z-index: 1;
}

.sidebar-heading {
    font-size: 24px;
    margin-bottom: 20px;
    color:black;
}

.category-list {
    list-style: none;
    padding: 0;
}

.category-list li {
    margin-bottom: 10px;
}

.category-list li a {
    color: #000;
    text-decoration: none;
    font-size: 18px;
}

.category-list li a:hover {
    text-decoration: underline;
}
a {
    text-decoration: none; /* Remove underline */
    color: inherit; /* Inherit text color from the parent element */
}

.product-header {
    position: absolute;
    top: 10px;
    right: 10px;
}

.favorites-icon {
    color: black; /* Change the color as desired */
    font-size: 24px;
}

.cart-button,
.buy-now-button {
    width: 100%;
    margin-top: 10px;
}

/* Add this CSS to your existing styles or in the <style> tag in the head section */
.search-container {
    text-align: center;
```

```
    }

    .search-container form {
        display: inline-block;
    }

    .search-container input[type=text] {
        padding: 4px;
        width: 400px;
        border: 1px solid #ccc;
        border-radius: 4px;
        box-sizing: border-box;
    }

    .search-container button {
        padding: 4px;
        background: black;
        color: #fff;
        border: 1px solid black;
        border-radius: 4px;
        cursor: pointer;
    }

    .search-container button:hover {
        background: #333;
        border: 1px solid #333;
    }

    .product-image {
        width: 200px; /* Set the desired width */
        height: 150px; /* Set the desired height */
        object-fit: cover; /* Maintain aspect ratio and cover the container */
        margin-top: 15px;
    }
  </style>
</head>
<body>

<!-- header section strats -->
<header class="header_section">
    <div class="container-fluid">
    <nav class="navbar navbar-expand-lg custom_nav-container ">
        <a class="navbar-brand" href="customer.php">
        <span>
            PHARMIO
        </span>
        </a>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav">
            <li>
            <div class="search-container">
            <form action="" method="GET">
                <input type="text" placeholder="Search..." name="search">
                <button type="submit"onclick=searchMedicine()><i class="fa fa-search"></i></button>
            </form>
```

```
        </div>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#"><i class="fas fa-user"></i> <?php echo
htmlspecialchars($_SESSION['first_name']); ?><i class="fas fa-angle-down"></i></a>
          <ul class="submenu1">
            <li class="nav-item">
              <a class="nav-link" href="cust_profile.php"><i class="fas fa-user"></i> My Profile</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="cart.php"><i class="fas fa-shopping-cart"></i> My Cart</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#"><i class="fas fa-heart"></i> My Wishlist</a>
            </li>
          </ul>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="logout.php"><i class="fas fa-sign-out-alt"></i> Log Out</a>
        </li>
        </ul>
      </div>
    </nav>
    </div>
</header>

<!-- sidebar starts -->
<div class="sidebar">
    <h2 class="sidebar-heading">Categories</h2>
    <ul class="category-list">
        <?php
            // Fetch category names from the database
            $categoryQuery = "SELECT category_name FROM category_details";
            $categoryResult = $conn->query($categoryQuery);

            if ($categoryResult->num_rows > 0) {
                while ($categoryRow = $categoryResult->fetch_assoc()) {
                    echo "<li><a href='#'>{$categoryRow['category_name']}</a></li>";
                }
            }
        ?>
    </ul>
</div>
<!-- sidebar ends -->

<div class="product-grid">
<?php
    // Retrieve medicines from the database
    $sql = "SELECT * FROM medicines";
    $result = $conn->query($sql);

    // Check if there are any medicines
    if ($result->num_rows > 0) {
?>
<div class="product-list">
```

```
<?php
    // Output data of each row
    while ($row = $result->fetch_assoc()) {
        // Assuming $row["Images"] contains the binary image data from the database
        $imageData = $row["Images"];

        // Convert the binary data to base64
        $base64Image = base64_encode($imageData);

        // Initialize $wishlistResult
        $wishlistResult = null;

        // Check if the medicine is in the wishlist for the current customer
        $wishlistQuery = "SELECT * FROM wish_lists WHERE Customer_id =
'{$_SESSION['username']}' AND Med_id = '{$row['Med_id']}'";
        $wishlistResult = $conn->query($wishlistQuery);

        // Determine if the heart icon should be filled or not
        $isInWishlist = ($wishlistResult && $wishlistResult->num_rows > 0);
?>
<div class="product-card" data-medicine-id="<?php echo $row['Med_id']; ?>">
<div class="product-header">
    <?php
        // Output the heart icon based on wishlist status
        if ($isInWishlist) {
            echo '<i class="favorites-icon fas fa-heart"></i>';
        } else {
            echo '<i class="favorites-icon far fa-heart"></i>';
        }
    ?>
</div>
    <img class="product-image" src="<?php echo $base64Image; ?>"
alt="<?php echo $row["Med_name"]; ?>">
    <a href="medicine_details.php"><h3><?php echo $row["Med_name"]; ?></h3></a>
    <p>Generic Name: <?php echo $row["generic_name"]; ?></p>
    <p>Price: Rs.<?php echo $row["Price"]; ?></p>
    <form action="add_to_cart.php" method="post">
        <input type="hidden" name="medicineName" value="<?php echo $row['Med_name']; ?>">
        <button class="cart-button" type="submit">
            <i class="fas fa-cart-plus"></i> Add to Cart
        </button>
    </form>
    <form action="">
    <button class="buy-now-button"><i class="fas fa-shopping-cart"></i> Buy Now</button>
    </form>
</div>
<?php
    }
?>
</div>
<?php
    } else {
        echo "No medicines available.";
    }
    // Close the database connection
```

```
        $conn->close();
    ?>
</div>
<script>
     function addToCart(medicineName) {
        // You can use AJAX to send a request to add_to_cart.php
        // Example using jQuery:
        $.post("add_to_cart.php", { medicineName: medicineName },
        function(data) {
                alert(data); // You can replace this with any other logic
        to handle the response
            });
          }
</script>
</body>
</html>
```

Cart

```
<?php
    include('db_config.php');
    session_start();

    if (!isset($_SESSION['username'])) {
        header('Location: login.php');
        exit();
    }

    // Assuming your customer_details table has a 'first_name' column
    $sql = "SELECT first_name FROM customer_details WHERE username =
'{$_SESSION['username']}'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        $row = $result->fetch_assoc();
        $_SESSION['first_name'] = $row['first_name'];
    }
?>
<!DOCTYPE html>
<html>
<head>
  <!-- Basic -->
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <!-- Mobile Metas -->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
  <!-- Site Metas -->
  <meta name="keywords" content="" />
  <meta name="description" content="" />
  <meta name="author" content="" />
  <link rel="shortcut icon" href="images/favicon.png" type="">

  <meta http-equiv="Cache-Control" content="no-cache, no-store, must-revalidate">
  <meta http-equiv="Pragma" content="no-cache">
  <meta http-equiv="Expires" content="0">
```

```
  <title> Pharmio </title>

  <!-- bootstrap core css -->
  <link rel="stylesheet" type="text/css" href="css/bootstrap.css"/>

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"
/>

  <!-- fonts style -->
  <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500;700;900&display=swap"
rel="stylesheet">

  <!--owl slider stylesheet -->
  <link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.carousel.min.css" />

  <!-- font awesome style -->
  <link href="css/font-awesome.min.css" rel="stylesheet" />

  <!-- Custom styles for this template -->
  <link href="css/style1.css" rel="stylesheet" />
  <!-- responsive style -->
  <link href="css/responsive.css" rel="stylesheet" />
  <style>
    .logout-button {
        background-color: #007bff;
        color: white;
        border: none;
        padding: 10px 20px;
        border-radius: 5px;
        cursor: pointer;
        transition: background-color 0.3s ease;
    }

    .logout-button:hover {
        background-color: #0056b3;
    }

    .product-list {
        display: flex;
        flex-wrap: wrap;
        justify-content: flex-start;
        padding: 20px;
    }

    .product-list {
        margin-left: 250px; /* Set margin equal to the width of the sidebar */
        display: flex;
        flex-wrap: wrap;
        justify-content: flex-start;
        padding: 20px;
    }

    .cart-heading {
        width: 100%;
```

```
    text-align: left;
    margin-bottom: 20px;
}

.cart-heading h2 {
    font-size: 24px;
    font-weight: bold;
    border-bottom: none; /* Remove the underline */
    padding-bottom: 10px;
    display: inline-block; /* Ensures the border only spans the width of the text */
}

.cart-heading h2::before {
    content: '\1F6D2'; /* Unicode for a shopping cart symbol */
    margin-right: 10px;
    font-size: 30px; /* Adjust the size of the cart symbol as needed */
}

.main-content {
    display: flex;
}

.product-list {
    flex-grow: 1;
    margin-right: 20px;
}

.cart-actions {
    width: 200px; /* Adjust the width as needed */
}
.cart-summary {
    width: 350px; /* Adjust the width as needed */
    padding: 20px;
    background-color: #fff;
    border: 0px solid #fff;
    border-radius: 0px;
    margin-left: 30px; /* Adjust the margin to create space between product list and cart summary */
}

.cart-summary p,
.cart-summary button {
    margin-bottom: 10px; /* Add margin to create space between elements */
}

/* Additional styles for the checkout button */
.checkout-button {
    width: 100%;
    padding: 10px;
    background-color: #000;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s ease;
    font-size: 14px;
```

```
}

.checkout-button:hover {
    background-color: #333;
}

.product-card {
    display: flex; /* Make the product card a flex container */
    border: 1px solid #ddd;
    border-radius: 8px;
    padding: 20px;
    margin: 10px;
    text-align: center;
    overflow-wrap: break-word;
    white-space: normal;
    position: relative;
}

.product-card img {
    width: 200px; /* Set the desired width */
    height: 150px; /* Set the desired height */
    object-fit: cover; /* Maintain aspect ratio and cover the container */
    margin-right: 20px; /* Add margin to separate image and details */
}

.product-card h3 {
    font-size: 18px;
    margin-bottom: 10px;
}

.product-card p {
    font-size: 14px;
    margin-bottom: 8px;
}

.product-card button {
    background-color: #000;
    color: #fff;
    border: none;
    padding: 10px 20px;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s ease;
    font-size: 14px;
}

.product-card button:hover {
    background-color: #333;
}

.product-header {
    position: absolute;
    top: 10px;
    right: 10px;
}
```

```
.submenu1 {
    display: none;
    position: absolute;
    background-color: #ffffff;
    box-shadow: 0px 8px 16px 0px rgba(0, 0, 0, 0.2);
    z-index: 1;
    list-style: none;
    padding: 0;
    text-align: left;
}

.nav-item:hover .submenu1 {
    display: block;
}

.submenu1 .nav-item {
    width: 200px;
    padding: 10px;
}

.submenu1 .nav-link {
    color: #000; /* Set text color to black */
    text-decoration: none;
    display: block;
    padding: 8px 16px; /* Add padding for better spacing */
    font-size: 14px; /* Reduce font size */
}

.submenu1 .nav-link:hover {
    background-color: #f2f2f2;
}

.sidebar {
    background-color: #f9f9f9;
    color: #fff;
    width: 250px;
    height: 100vh;
    padding: 20px;
    position: fixed;
    left: 0;
    top: 76px;
    z-index: 1;
}

.sidebar-heading {
    font-size: 24px;
    margin-bottom: 20px;
    color: black;
}

.category-list {
    list-style: none;
    padding: 0;
}
```

```
    .category-list li {
        margin-bottom: 10px;
    }

    .category-list li a {
        color: #000;
        text-decoration: none;
        font-size: 18px;
    }

    .category-list li a:hover {
        text-decoration: underline;
    }

    a {
        text-decoration: none; /* Remove underline */
        color: inherit; /* Inherit text color from the parent element */
    }
  </style>
</head>
<body>

<!-- header section strats -->
<header class="header_section">
    <div class="container-fluid">
    <nav class="navbar navbar-expand-lg custom_nav-container ">
        <a class="navbar-brand" href="customer.php">
        <span>
            PHARMIO
        </span>
        </a>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav">
            <li>
            <div class="search-container">
            <form action="" method="GET">
                <input type="text" placeholder="Search..." name="search">
                <button type="submit" onclick="searchMedicine()"><i class="fa fa-search"></i></button>
            </form>
            </div>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#"><i class="fas fa-user"></i> <?php echo
htmlspecialchars($_SESSION['first_name']); ?><i class="fas fa-angle-down"></i></a>
                <ul class="submenu1">
                    <li class="nav-item">
                        <a class="nav-link" href="edit_profile.php"><i class="fas fa-user"></i> My Profile</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="cart.php"><i class="fas fa-shopping-cart"></i> My Cart</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#"><i class="fas fa-heart"></i> My Wishlist</a>
                    </li>
```

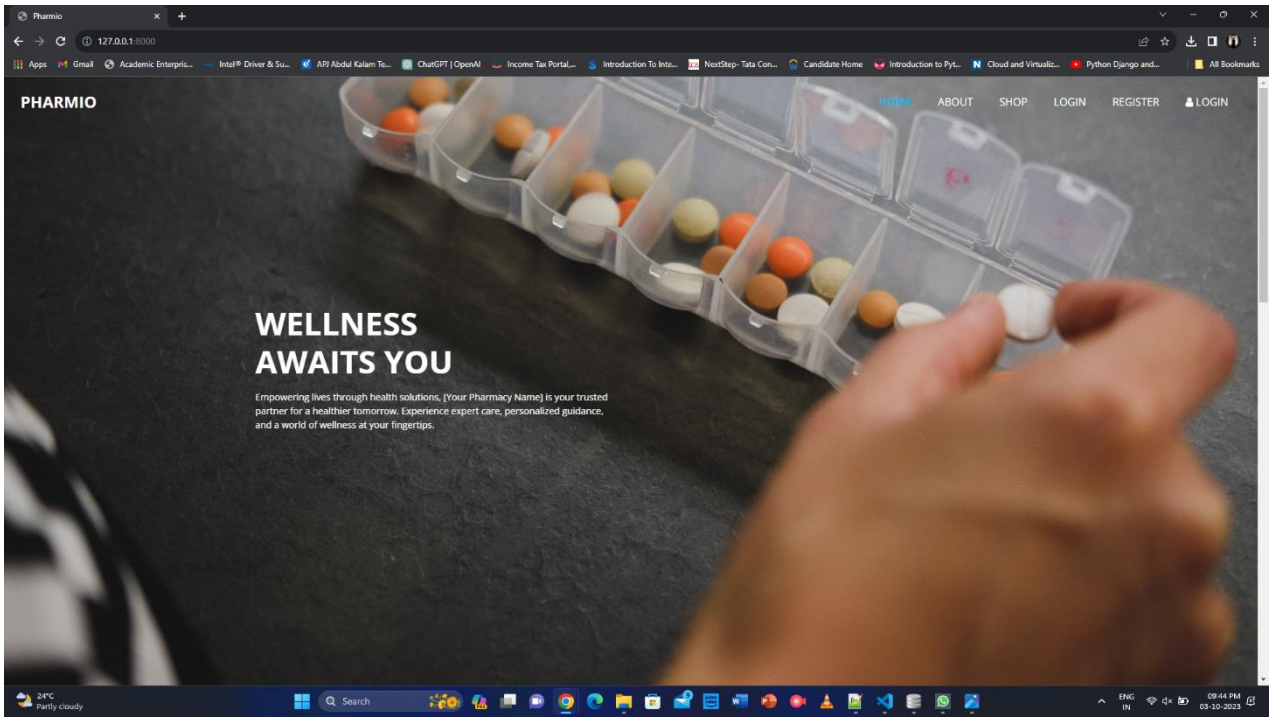
```
            </ul>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="logout.php"><i class="fas fa-sign-out-alt"></i> Log Out</a>
          </li>
          </ul>
        </div>
    </nav>
    </div>
</header>

<!-- sidebar starts -->
<div class="sidebar">
    <h2 class="sidebar-heading">Categories</h2>
    <ul class="category-list">
        <?php
            // Fetch category names from the database
            $categoryQuery = "SELECT category_name FROM category_details";
            $categoryResult = $conn->query($categoryQuery);

            if ($categoryResult->num_rows > 0) {
                while ($categoryRow = $categoryResult->fetch_assoc()) {
                    echo "<li><a href='#'>{$categoryRow['category_name']}</a></li>";
                }
            }
        ?>
    </ul>
</div>
<!-- sidebar ends -->
<div class="main-content">
    <div class="product-list">
        <div class="cart-heading">
            <h2>Your Cart</h2>
        </div>
        <div class="cart-products">
            <?php
            // Fetch cart items from the database
            $cartQuery = "SELECT cart_details.quantity, medicines.Med_name, medicines.Images,
medicines.Price
                    FROM cart_details
                    JOIN medicines ON cart_details.med_id = medicines.med_id
                    WHERE cart_details.customer_id = (SELECT customer_id FROM customer_details
WHERE username = '{$_SESSION['username']}')";

            $cartResult = $conn->query($cartQuery);

            if ($cartResult->num_rows > 0) {
                while ($cartRow = $cartResult->fetch_assoc()) {
                    // Assuming $cartRow["Images"] contains the binary image data from the database
                    $imageData = $cartRow["Images"];

                    // Convert the binary data to base64
                    $base64Image = base64_encode($imageData);

                    // Display the product details
```

```
                echo "<div class='product-card'>";
                echo "<img class='product-image' src='data:image;base64,{$base64Image}'>";
                echo "<div class='product-details'>";
                echo "<h3>{$cartRow['Med_name']}</h3>";

                // Check if 'Price' key exists in the array
                if (isset($cartRow['Price'])) {
                    echo "<p>Price: {$cartRow['Price']}</p>";
                } else {
                    echo "<p>Price: Not available</p>";
                }

                // Check if 'quantity' key exists in the array
                if (isset($cartRow['quantity'])) {
                    echo "<p>Quantity: {$cartRow['quantity']}</p>";
                } else {
                    echo "<p>Quantity: Not available</p>";
                }

                echo "<div class='product-actions'>";
                echo "<button class='remove-button'>Remove from cart</button>";
                echo "</div>";
                echo "</div>";
                echo "</div>";
            }
        } else {
            echo "<p>Your cart is empty.</p>";
        }
        ?>
    </div>
</div>

<div class="cart-summary">
    <!-- Display total amount and checkout button here -->
    <?php
    // Calculate total amount
    $totalQuery = "SELECT SUM(quantity * price) AS total
                FROM cart_details
                JOIN medicines ON cart_details.med_id = medicines.med_id
                WHERE cart_details.customer_id = (SELECT customer_id FROM customer_details
WHERE username = '{$_SESSION['username']}')";

    $totalResult = $conn->query($totalQuery);
    $totalRow = $totalResult->fetch_assoc();
    $totalAmount = $totalRow['total'];
    // After calculating the total amount
    $_SESSION['total_amount'] = $totalAmount;

    echo "<p>Total Amount: $totalAmount</p>";
    echo "<button class='checkout-button' id='checkoutBtn'>Proceed to Checkout</button>";
    ?>
</div>
</div>
</div>
<script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>
```

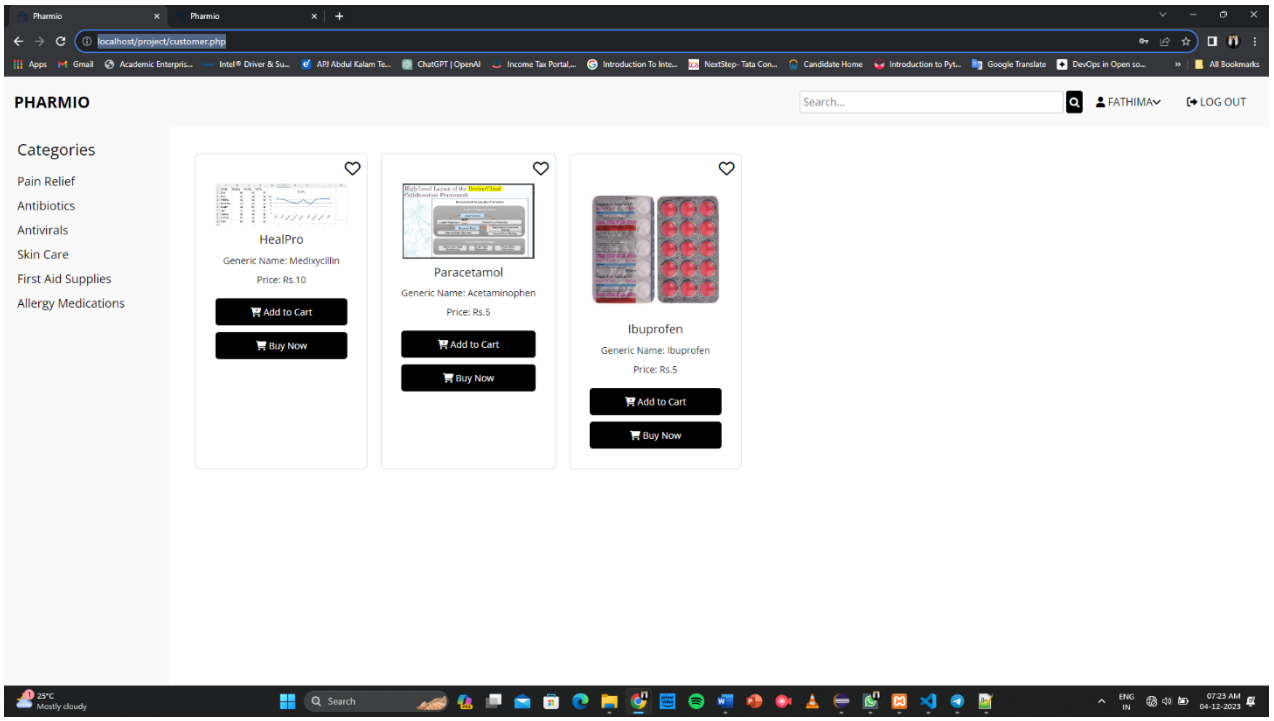
```
<script>
   $(document).ready(function(){
      // Add a click event listener to the button
      $('#checkoutBtn').click(function(){
         // Navigate to payment.php
         window.location.href = 'payment.php';
      });
   });
</script>

</body>
</html>
```

9.1 Screen Shots

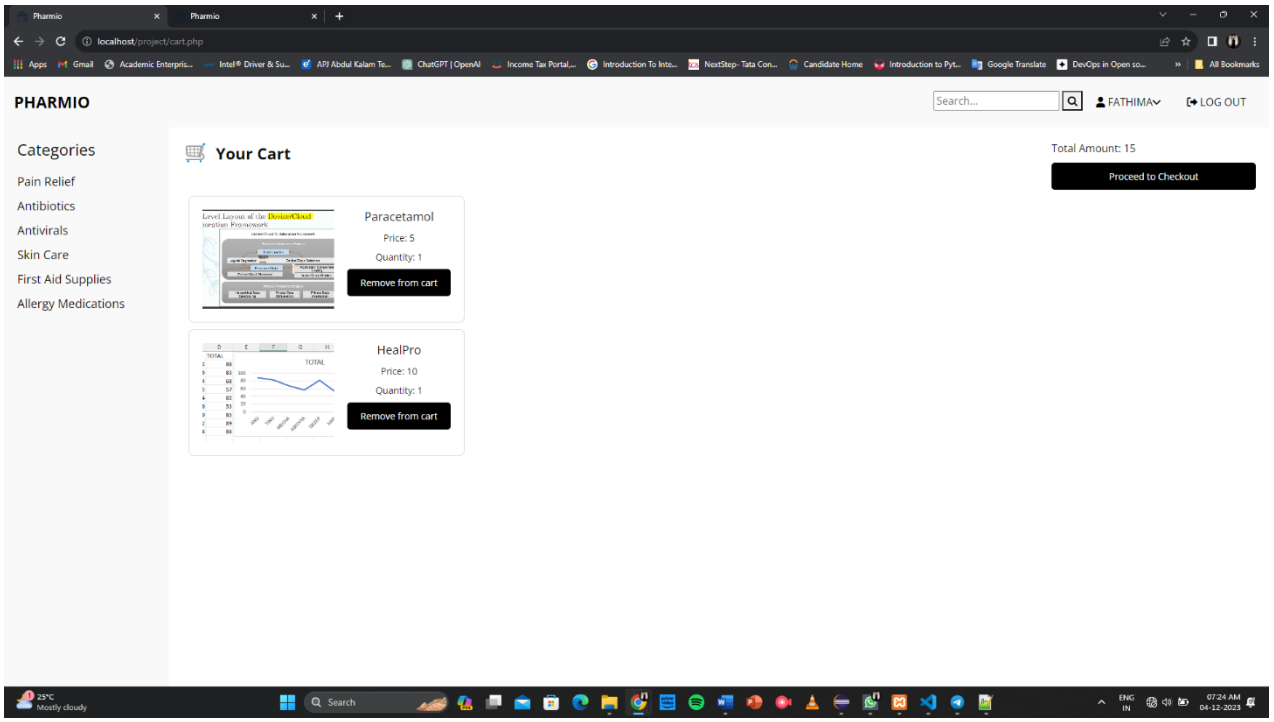
Home



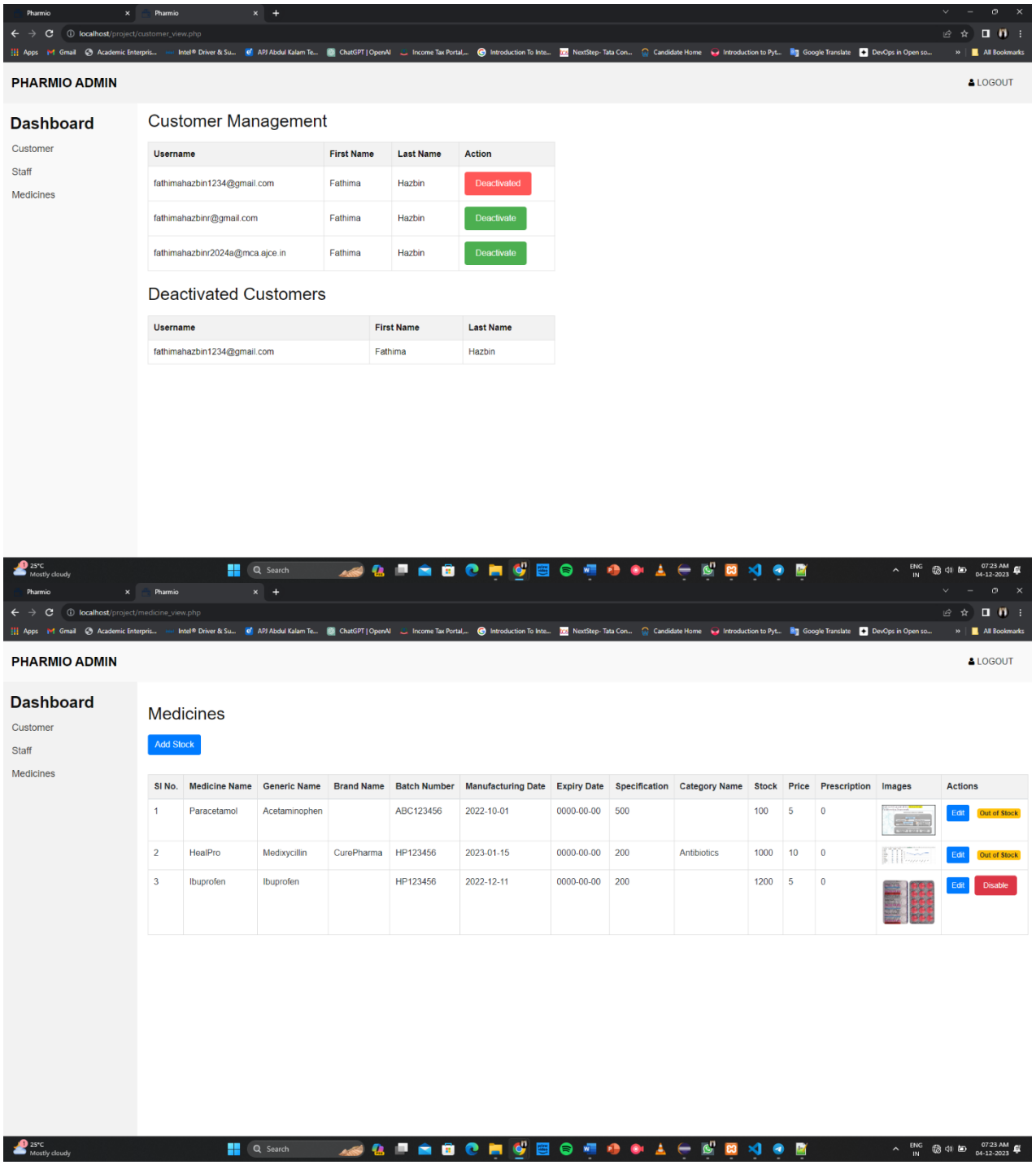
Customer

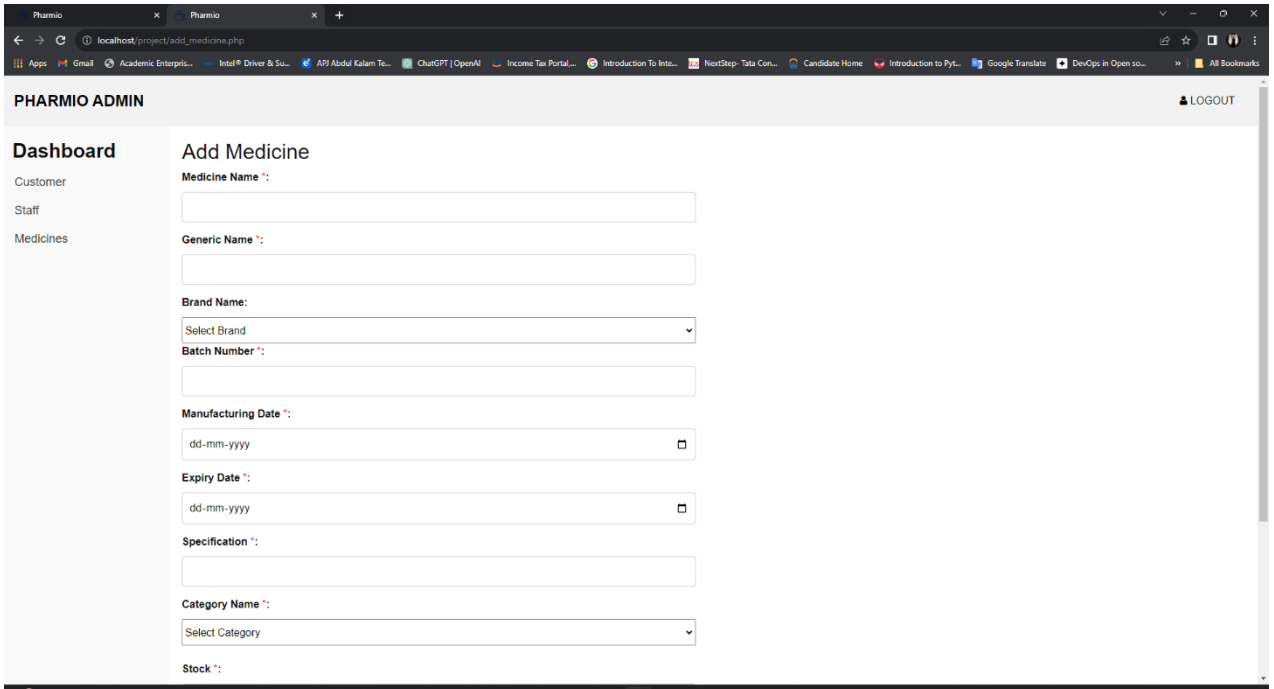


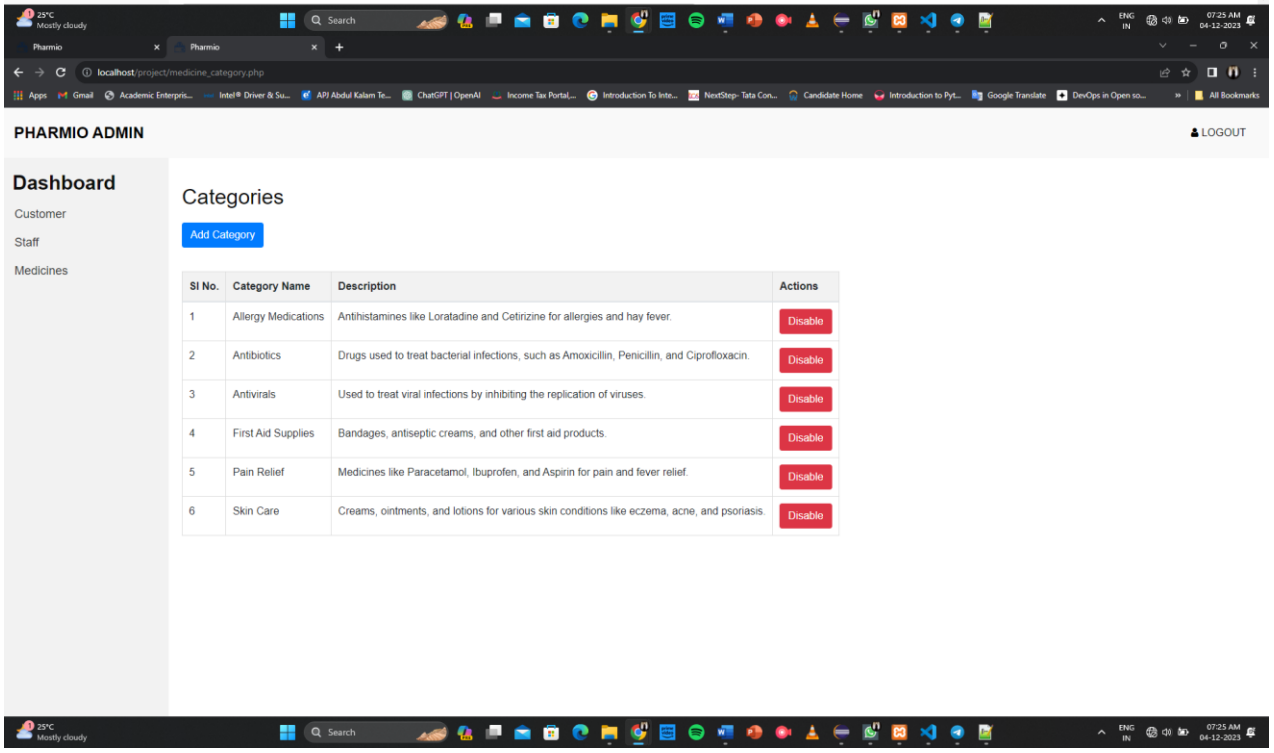
Cart



Admin Pages


PHARMIO ADMIN
LOGOUT
Dashboard
Customer
Staff
Medicines
Customer Management
Username
First Name
Last Name
Action
fathimahazbin1234@gmail.com
Fathima
Hazbin
Deactivated
fathimahazbinr@gmail.com
Fathima
Hazbin
Deactivate
fathimahazbinr2024a@mca.ajce.in
Fathima
Hazbin
Deactivate
Deactivated Customers
Username
First Name
Last Name
fathimahazbin1234@gmail.com
Fathima
Hazbin
PHARMIO ADMIN
LOGOUT
Dashboard
Customer
Staff
Medicines
Medicines
Add Stock
Sl No.
Medicine Name
Generic Name
Brand Name
Batch Number
Manufacturing Date
Expiry Date
Specification
Category Name
Stock
Price
Prescription
Images
Actions
1
Paracetamol
Acetaminophen
ABC123456
2022-10-01
0000-00-00
500
100
5
0
Edit
Out of Stock
2
HealPro
Medixycillin
CurePharma
HP123456
2023-01-15
0000-00-00
200
Antibiotics
1000
10
0
Edit
Out of Stock
3
Ibuprofen
Ibuprofen
HP123456
2022-12-11
0000-00-00
200
1200
5
0
Edit
Disable


PHARMIO ADMIN
LOGOUT
Dashboard
Customer
Staff
Medicines
Add Medicine
Medicine Name *:
Generic Name *:
Brand Name:
Select Brand
Batch Number *:
Manufacturing Date *:
dd-mm-yyyy
Expiry Date *:
dd-mm-yyyy
Specification *:
Category Name *:
Select Category
Stock *:


PHARMIO ADMIN
LOGOUT
Dashboard
Customer
Staff
Medicines
Categories
Add Category
Sl No.
Category Name
Description
Actions
1
Allergy Medications
Antihistamines like Loratadine and Cetirizine for allergies and hay fever.
Disable
2
Antibiotics
Drugs used to treat bacterial infections, such as Amoxicillin, Penicillin, and Ciprofloxacin.
Disable
3
Antivirals
Used to treat viral infections by inhibiting the replication of viruses.
Disable
4
First Aid Supplies
Bandages, antiseptic creams, and other first aid products.
Disable
5
Pain Relief
Medicines like Paracetamol, Ibuprofen, and Aspirin for pain and fever relief.
Disable
6
Skin Care
Creams, ointments, and lotions for various skin conditions like eczema, acne, and psoriasis.
Disable