A Report

on

Creational Activity Learning

"Pharmacy Management System"

Submitted By

Kolpe Dnyaneshwar Mhalu

Roll No: 91

PRN : UCS22M1078

(TY Computer)

Guided By

Prof. P.M. Dhanrao

Department of Computer Engineering

SANJIVANI COLLEGE OF ENGINEERING

(An Autonomous Institution)

Kopargaon– 423 603, Maharashtra

In the academic year 2024-25

Sanjivani College of Engineering, Kopargaon

Department of Computer Engineering

CERTIFICATE

This is to certify that

Kolpe Dnyaneshwar Mhalu

(TY Computer)

Has successfully completed her CAL report on

" Pharmacy Management System "

Towards the partial fulfillment of Bachelor's degree In Computer Engineering During the academic year 2024-25.

Prof. P.M. Dhanrao
[Subject Teacher]

Dr. M. A. Jawale
[H.O.D. Comp Engg]

ACKNOWLEDGEMENT

I would like to take this opportunity to express my sincere gratitude to all those who helped me directly or indirectly in the successful completion of this project.

First and foremost, I would like to thank my project supervisor, **Prof. P.M. Dhanrao** for their invaluable guidance, timely suggestions, and continuous support throughout the development of the Airline Management System project. Their insights and encouragement inspired me to explore new dimensions of learning and enhanced the quality of my work significantly.

I would also like to extend my heartfelt thanks to the **Dr. M.A.Jawale** of Head of the Computer Department and all the faculty members of the Information Technology Department at Sanjivani College of Engineering for providing the necessary facilities and resources that enabled me to work efficiently on this project.

Special thanks to my family and friends for their constant motivation and emotional support throughout the course of this project. Their encouragement and belief in my capabilities helped me stay focused and confident.

Finally, I would like to thank all those who contributed directly or indirectly in making this project a success. This experience has enriched my knowledge and given me the confidence to take up future technical challenges with greater assurance.

Thank you all.

Mr. Kolpe Dnyaneshwar Mhalu
T.Y. Computer
Roll No. 91

INDEX

Sr. No.	Title	Page No.
1	Abstract	5
2	Introduction	6
3	Problem Statement	7
4	Screenshots	8
5	Project Code	9
6	Conclusion	14
7	References	15

Abstract

The Pharmacy Management System is a console-based application designed to streamline core pharmacy operations such as drug inventory management, customer handling, billing, and order tracking. With the growing need for automation in healthcare and pharmaceutical services, this project aims to replace traditional manual processes with a reliable and efficient digital solution.

The application simplifies pharmacy workflows for both administrators and staff. Admin users can add, update, or remove drug records, manage customer details, and process sales with real-time inventory updates. The system also generates alerts for expired or low-stock drugs and maintains an audit log of critical modifications for accountability.

Developed using Java and connected to a PostgreSQL database via JDBC, the system ensures fast processing, data integrity, and structured management of sensitive pharmaceutical data. The use of modern programming practices and object-oriented design enhances scalability and maintainability.

- Key technologies used include:
  - Backend: Java (Console-based)
  - Database: PostgreSQL
  - Build Tool: Maven
  - IDE: IntelliJ IDEA / Eclipse

This project emphasizes modular design, secure data handling, and automation of repetitive tasks to minimize human error and improve the accuracy and speed of pharmacy services.

Introduction

Pharmacies are a critical component of the healthcare system, responsible for the distribution of medicines and health-related products to the public. As demand for timely and accurate dispensing of medications grows, especially in urban and semi-urban areas, it becomes increasingly important to replace manual, paper-based pharmacy operations with automated, efficient digital solutions. Traditional systems often suffer from human error, stock mismanagement, and poor record-keeping, leading to operational inefficiencies and reduced customer satisfaction.

The Pharmacy Management System is developed to address these challenges by providing a software platform that automates essential pharmacy operations such as drug inventory tracking, sales processing, customer management, and invoice generation. This system is designed to streamline day-to-day pharmacy workflows, reduce administrative burden, and improve accuracy and service speed through real-time data handling and validation.

The project is developed using Java (console-based) for the backend logic and PostgreSQL as the relational database to manage structured data efficiently. JDBC is used to facilitate seamless database connectivity, while Maven manages build dependencies. The development was carried out in the IntelliJ/Eclipse IDE environment, allowing for easy debugging, testing, and modular expansion.

The system is divided into two main functional modules:

- Admin Module: Allows pharmacy managers or administrators to add, update, and delete drug records, manage customer details, track stock levels, and view reports. It also includes alerts for expired or low-stock drugs and maintains an audit trail of all inventory changes.
- Sales Module: Enables the cashier or pharmacist to search for drugs, create invoices, process customer purchases, and update stock in real time.

With its structured design and focus on real-world usability, the Pharmacy Management System is scalable for future integration with barcode scanning, GST-compliant billing, user authentication, and online prescription uploads. Beyond its utility in real pharmacies, the project also serves as a solid educational tool for understanding core concepts in object-oriented programming, JDBC-based database interaction, and CRUD-based application design.

Problem Statement

In today's fast-paced and digitally transforming healthcare environment, pharmacies play a vital role in ensuring timely access to medications and maintaining accurate medical supply chains. However, many small and medium-scale pharmacies still rely on manual or semi-automated systems to manage their operations, which leads to numerous inefficiencies and risks. These include stock mismanagement, outdated inventory records, billing errors, and a lack of real-time tracking for drug availability and expiry.

Manual systems are prone to human error and fail to scale effectively as the volume of sales and customer interactions increases. Tasks such as updating stock after a sale, identifying near-expiry drugs, managing customer records, and generating accurate invoices become cumbersome and error-prone. As a result, customers may face delays, receive incorrect prescriptions, or experience dissatisfaction due to unavailability of medicines or pricing discrepancies.

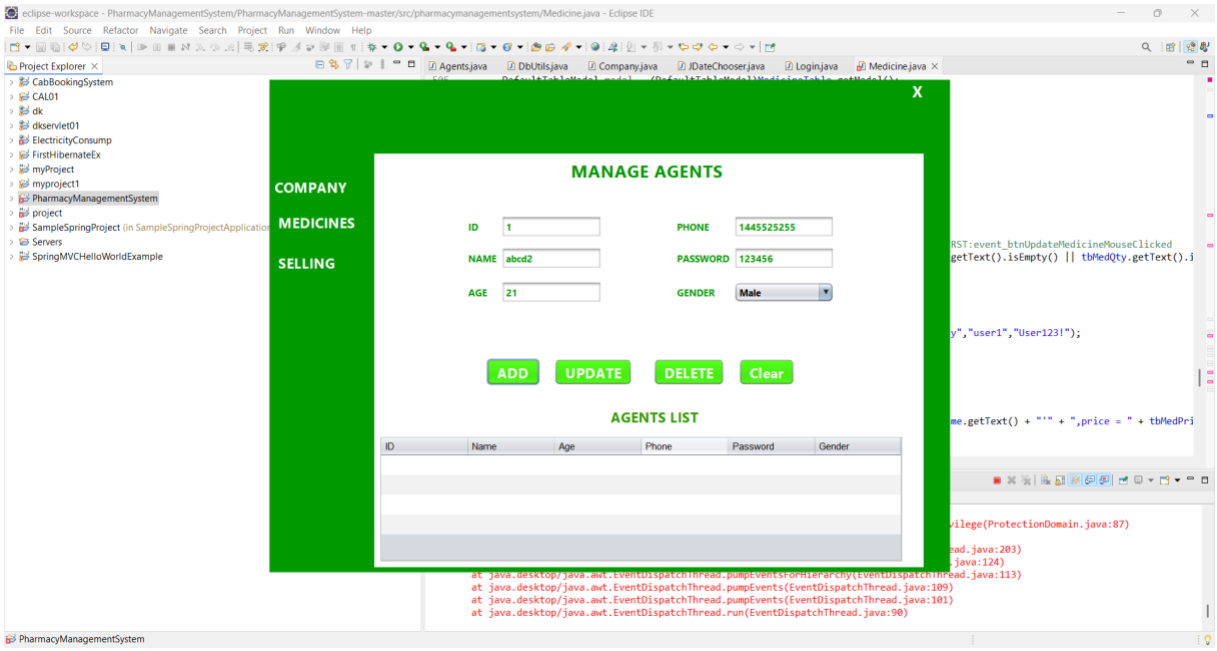
The absence of a centralized, automated Pharmacy Management System creates significant challenges for pharmacy administrators in maintaining up-to-date inventory, tracking sales, and ensuring regulatory compliance. It also impedes the ability to analyze data for decision-making, such as identifying top-selling medicines, understanding purchase trends, or optimizing stock levels.

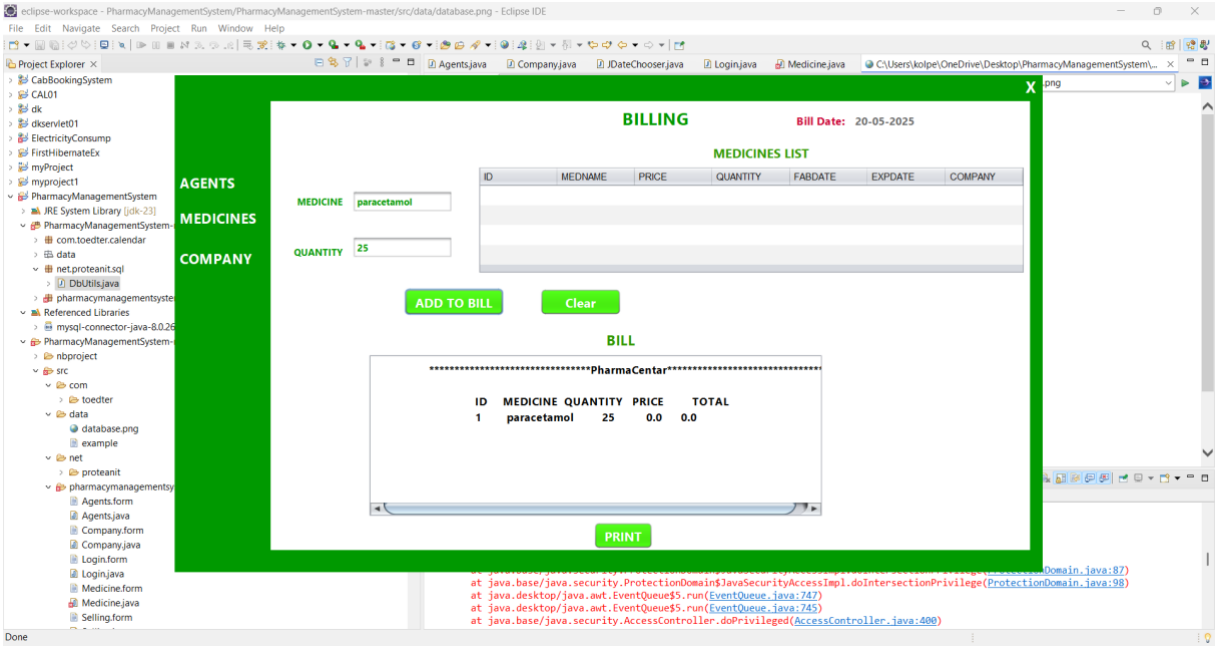
There is a critical need for a robust, efficient, and user-friendly Pharmacy Management System that automates inventory handling, billing, and customer management processes. Such a system should provide a secure and centralized platform for pharmacists to monitor drug stock, update sales records, generate invoices, and maintain customer history in real-time. It should also incorporate alerts for low stock and expired drugs, along with audit logs for sensitive operations.

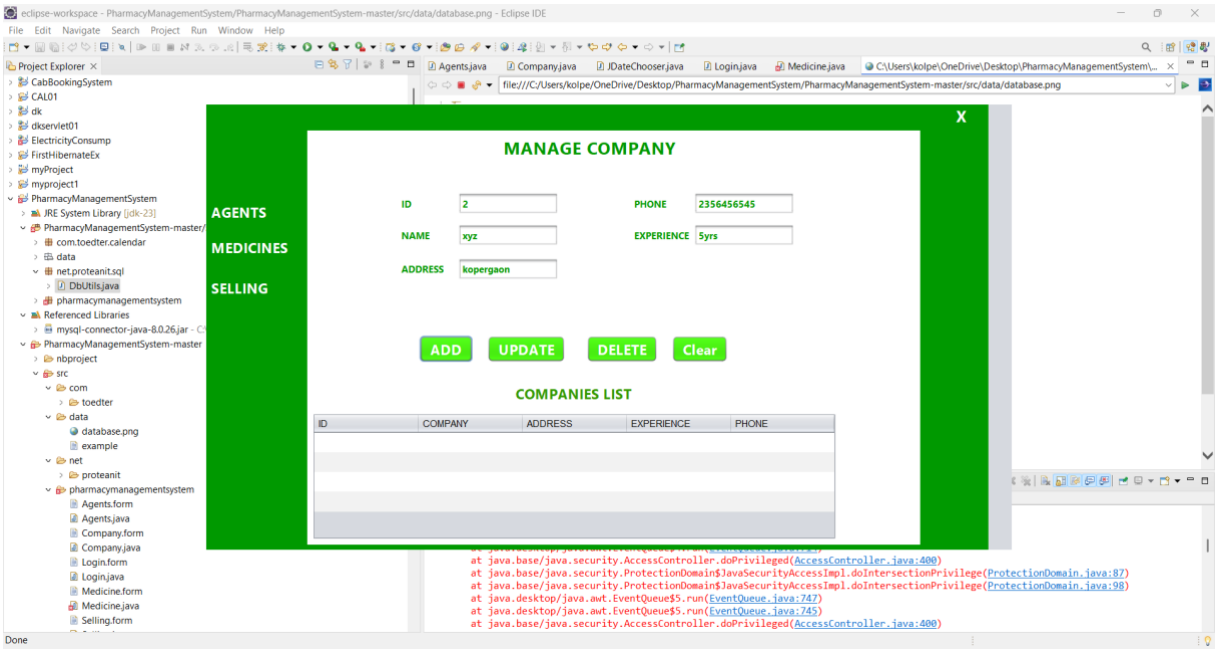
This project seeks to develop a console-based application that integrates these core functionalities using Java and PostgreSQL. The aim is to eliminate the inefficiencies of traditional systems, improve pharmacy service quality, reduce operational risks, and contribute to the digital modernization of pharmacy operations.

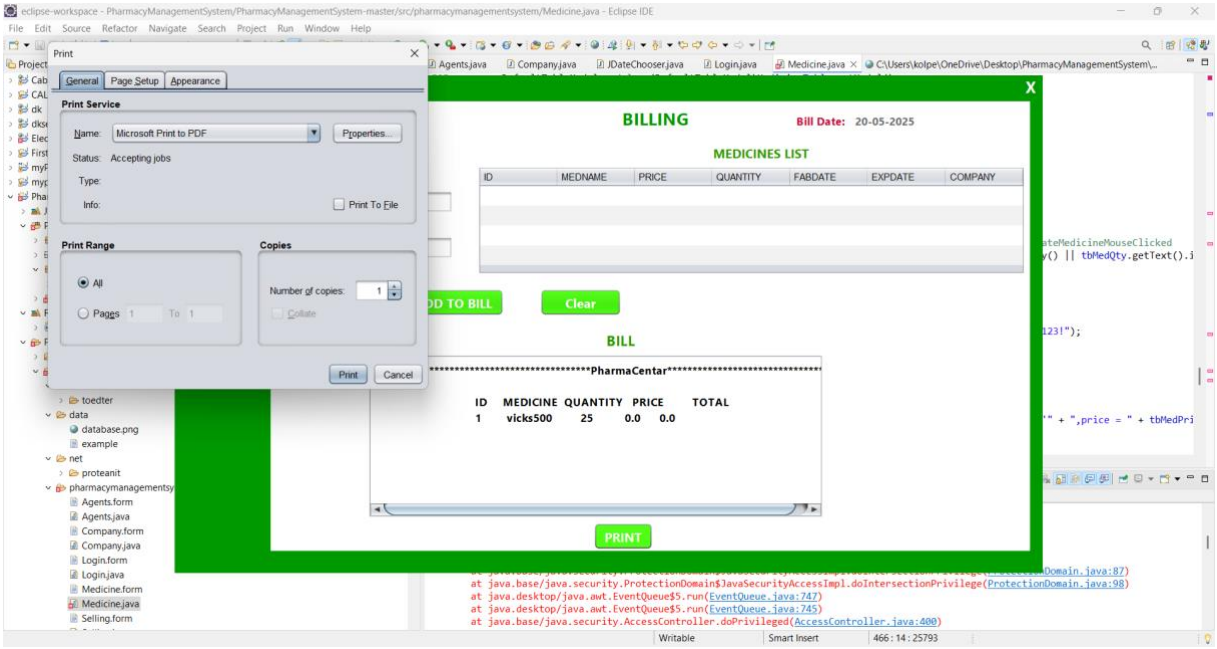
Screenshots

Home Page ➔






MANAGE COMPANY
AGENTS
MEDICINES
SELLING
ID
2
PHONE
2356456545
NAME
xyz
EXPERIENCE
5yrs
ADDRESS
kopergaon
ADD
UPDATE
DELETE
Clear
COMPANIES LIST
ID
COMPANY
ADDRESS
EXPERIENCE
PHONE


Print
General
Page Setup
Appearance
Print Service
Name:
Microsoft Print to PDF
Properties...
Status:
Accepting jobs
Type:
Info:
Print To File
Print Range
All
Pages
Copies
Number of copies:
Print
Cancel
BILLING
Bill Date:
20-05-2025
MEDICINES LIST
ID
MEDNAME
PRICE
QUANTITY
FABDATE
EXPDATE
COMPANY
Clear
BILL
PharmaCentar
ID MEDICINE QUANTITY PRICE TOTAL
1 vicks500 25 0.0 0.0
PRINT

Project Code

```
import java.util.*;
public class Main {

	public static void ordermedicine(Passenger p) {
		Medicinebooker=new CustBooker();
		if(CustBooker.avaiWait==0) {
			System.out.println("No Medicine Avilable");
			return;
		}
		if((p.berthPref.equals("L") && TicketBooker.avaiLow>0) ||
		  (p.berthPref.equals("M") && TicketBooker.avaiMid>0) ||
		  (p.berthPref.equals("U") && TicketBooker.avaiUp>0)) {

//			System.out.println("Your berth preference available..");
			//lower
			if(p.berthPref.equals("L")) {
				System.out.println("Lower Berth Given..");
				booker.bookTicket(p, TicketBooker.lbPos.get(0), "L");
				TicketBooker.lbPos.remove(0);
				TicketBooker.avaiLow--;
			}
			//middle
			else if(p.berthPref.equals("M")) {
				System.out.println("Middle Berth Given..");
				booker.bookTicket(p, TicketBooker.mbPos.get(0), "M");
				TicketBooker.mbPos.remove(0);
				TicketBooker.avaiMid--;
			}
```

```
                //upper
                else if(p.berthPref.equals("U")) {
                        System.out.println("Upper Berth Given..");
                        booker.bookTicket(p, TicketBooker.ubPos.get(0), "U");
                        TicketBooker.ubPos.remove(0);
                        TicketBooker.avaiUp--;
                }

        }
        //if prefered berth not available..
        else if(TicketBooker.avaiLow>0) {
//              System.out.println("Your preference not available");
                System.out.println("Lower Berth Given..");
                booker.bookTicket(p, TicketBooker.lbPos.get(0), "L");
                TicketBooker.lbPos.remove(0);
                TicketBooker.avaiLow--;

        }
        else if(TicketBooker.avaiMid>0) {
//              System.out.println("Your preference not available");
                System.out.println("Middle Berth Given..");
                booker.bookTicket(p, TicketBooker.mbPos.get(0), "M");
                TicketBooker.mbPos.remove(0);
                TicketBooker.avaiMid--;

        }
        else if(TicketBooker.avaiUp>0) {
//              System.out.println("Your preference not available");
```

```
                System.out.println("Upper Berth Given..");
                booker.bookTicket(p, TicketBooker.ubPos.get(0), "U");
                TicketBooker.ubPos.remove(0);
                TicketBooker.avaiUp--;
            }

            //if no berth available...book the rac ticket..
            else if(TicketBooker.avaiRac>0) {
                System.out.println("Berth Not Available");
                System.out.println("RAC Tickets Available");
                booker.bookRAC(p, TicketBooker.racPos.get(0), "RAC");
                TicketBooker.racPos.remove(0);
                TicketBooker.avaiRac--;
            }

            //if no berth and rac avilable ...book into waiting list
            else if(TicketBooker.avaiWait>0) {
                System.out.println("No Berth and RAC are Available..");
                System.out.println("Waiting List Available...");
                booker.addToWait(p,TicketBooker.waitPos.get(0),"WL");
                TicketBooker.waitPos.remove(0);
                TicketBooker.avaiWait--;
            }
        }

        public static void cancelTicket(int id) {
            TicketBooker booker=new TicketBooker();
            if(!TicketBooker.passengers.containsKey(id)) ///checking in the map for
passengers details..
```

```
                System.out.println("Invalid Passenger ID...please enter the valid ID");
            else
                booker.cancelTicket(id);
        }

        public static void main(String[] args) {
            try (Scanner s = new Scanner(System.in)) {
                boolean loop=true;
                while(loop) {
                    System.out.println("1. Book Ticket\n2. Cancel Ticket\n3. Available ticket\n4. Booked Tickets\n5. Exit");
                    int choice =s.nextInt();
                    switch(choice) {
                        case 1:
                            System.out.println("Enter Passenger name,age and berthPref (L,M or U)");
                            String name=s.next();
                            int age=s.nextInt();
                            String berthPref=s.next();

                            Passenger p=new Passenger(name,age,berthPref); //instance of Passenger class..
                            bookTicket(p);
                            break;

                        case 2:
                            System.out.println("Enter the passenger ID to cancel");
                            int id=s.nextInt();
                            cancelTicket(id);
                            break;
```

```
                    case 3:
                        System.out.println("Tickets Available..");
        System.out.println("_____");
                        TicketBooker booker=new TicketBooker();
                        booker.printAvailable();
                        break;

                    case 4:
                    {
                        System.out.println("Booked Tickets");
        System.out.println("_____");
                        TicketBooker booker1=new TicketBooker();
                        booker1.printBooked();
                        break;
                    }

                    case 5:
                        loop=false;
                        break;

                }
            }
        }

    }

}
```

Conclusion

The development of the Pharmacy Management System represents a vital advancement in the automation and streamlining of pharmacy operations. By replacing outdated manual methods with a centralized digital platform, the system significantly improves the efficiency, accuracy, and reliability of managing medication inventories, customer records, prescriptions, and sales transactions.

Built using modern technologies such as Java (JSP and Servlets) and MySQL, the system ensures smooth backend processing and an intuitive user interface for pharmacists, staff, and administrators. Its modular, three-tier architecture enhances scalability and maintainability, allowing for seamless integration of future features such as online prescription uploads, automated restocking alerts, and integration with healthcare systems.

Throughout the development lifecycle, the application of sound software engineering practices, thorough testing, and attention to usability has resulted in a robust, secure, and user-friendly solution. The system not only meets the functional needs of a pharmacy but also serves as a comprehensive learning experience in full-stack application development, database integration, and healthcare software design.

In conclusion, the Pharmacy Management System successfully addresses the core challenges of pharmacy operations, offering a practical and effective solution that improves service quality, reduces human error, and paves the way for digital transformation in the pharmaceutical sector

References

- **Java The Complete Reference** – Herbert Schildt, McGraw-Hill Education.

  (For understanding Java fundamentals, Servlets, and JSP technologies.)

- **Head First Servlets and JSP** – Bryan Basham, Kathy Sierra, Bert Bates, O'Reilly Media.

  (For in-depth knowledge of building dynamic web applications using Servlets and JSP.)

- **MySQL Documentation** – https://dev.mysql.com/doc/

  (For database design, queries, and integration with Java.)

- **Apache Tomcat Documentation** – https://tomcat.apache.org/tomcat-10.0-doc/

  (For configuration and deployment of Java-based web applications.)

- **W3Schools HTML, CSS, JavaScript Tutorials** – https://www.w3schools.com/

  (For front-end design and implementation references.)

- **GeeksforGeeks Tutorials** – https://www.geeksforgeeks.org/

  (For algorithms, Java examples, and system design patterns.)

- **Stack Overflow** – https://stackoverflow.com/

  (For troubleshooting coding issues and reviewing implementation best practices.)

- **MDN Web Docs** – https://developer.mozilla.org/

  (For JavaScript, HTML, and CSS documentation and examples.)