# PHARMACEUTICAL SERVICES



An

Object-Oriented Programming through Java Course Project Report in partial fulfilment of the degree

**Bachelor of Technology**

in

**Computer Science & Engineering**

## By

CH.Rakshitha 2103A51105

A.Harshhitha Reddy 2203A51L02

D.Neela Lohitha 2203A51L08

Under the guidance of

## Mr. K. Balakrishna

Assistant Professor, Department of CSE.

**Submitted to**



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**CERTIFICATE**

This is to certify that the **Object Oriented Programming through Java - Course Project** Report entitled **“PHARMACEUTICAL SERVICES”** is a record of bonafide work carried out by the students CH. Rakshitha, A. Harshhitha Reddy, D. NeelaLohitha bearing Roll No(s) 2103A51105,2203A51L02,2203A51L08 during the academic year 2021-25 in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Computer Science & Engineering** by the Jawaharlal Nehru Technological University, Hyderabad.

**Lab In-charge Head of the Department**

# TABLE OF CONTENTS

|  |  |
| --- | --- |
| 1. ABSTRACT | 5 |
| 1. OBJECTIVE OF PROJECT | 6 |
| 1. DEFINATIONS OF THE ELEMENTS USED IN THE PROJECT. | 7-8 |
| 1. DESIGN | 9 |
| 1. IMPLEMENTATION OF CODE | 10-14 |
| 1. RESULT SCREENS | 15-18 |
| 1. CONCLUSION | 19 |

# ACKNOWLEDGEMENT

We express our thanks to Course co-coordinator **Mr., K. Ravi Chythanya, Asst. Prof.** for guiding us from the beginning through the end of the Course Project. We express our gratitude to Head of the department CS&AI, **Dr. M. Sheshikala, Associate Professor** for encouragement, support and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, School of Computer Science and Artificial Intelligence, **Dr C. V. Guru Rao**, for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

# ABSTRACT

Pharmacy was always more than just a place to get your prescription. Patients treated pharmacists as consultants, someone to help them choose an over-the-counter medicine or make sense of a prescription’s dosage and instructions. Always happy to oblige, they rarely had enough information about a person’s medical history, allergies, or treatment plans to give more complete advice. This, however, is changing. The healthcare industry is experiencing transformations and pharmacies are no exception.

Patients have access to their medical data and ability to securely share it. Hospitals are encouraged — if not forced — to become interoperable and connect with all the other players in the field. Valuable data is collected and transformed into insights that help make life or death decisions with more confidence. Basically, it’s much easier to abandon manual processes and participate in patient care instead, all the while growing as a business and staying competitive on the growing market.

Pharmaceutical Services (PS) and how it helps achieve two business goals: 1) having a more efficient, automated pharmacy workflow, and 2) staying above the competition by improving patient outcomes and providing a better customer experience.

.

# OBJECTIVE

This new pharmaceutical services is quite different from other pharmacy software. This pharmacy software has been built for team collaboration for the development team and all its members who will be a part of development team. It’s the user friendly software with customization feature that makes your work easier. As we all know, pharmaceutical services is one of the most complex software to build and keep all consistent information to exist in this competitive market and this software having all feature which will meet your requirements.

# DEFINITIONS OF THE ELEMENTS USED IN THE PROJECT

* CLASS AND OBJECT IN JAVA:

**An object is an instance of a class.** A class is a template or blueprint from which objects are created. So, an object is the instance(result) of a class.

1. An object is a real-world entity.
2. An object is a run time entity.
3. The object is an entity which has state and behavior.
4. The object is an instance of a class.

* CONSTRUCTOR

Constructor in Java is a special method that is used to initialize objects. The constructor is called when an object of a class is created. We used Parameterized Constructor

* METHODS

Method is a block of code which only runs when it is called. You can pass data, known as parameters, into a method. Methods are used to perform certain actions, and they are also known as functions. Predefined Methods. As the name gives it, predefined methods in Java are the ones that the Java class libraries already define.

* KEYWORDS

Java keyword is one of 50 reserved terms that have a special function and a set definition in the Java programming language. The fact that the terms are reserved means that they cannot be used as identifiers for any other program elements, including classes, subclasses, variables, methods and objects.

# The four access modifiers in Java are public and private.

* ACCESS MODIFIERS

There are two types of modifiers in Java: **access modifiers** and **non-access modifiers**.

The access modifiers in Java specifies the accessibility or scope of a field, method, constructor, or class. We can change the access level of fields, constructors, methods, and class by applying the access modifier on it.

.

1. **Private**: The access level of a private modifier is only within the class. It cannot be accessed from outside the class.
2. **Public**: The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.

# METHOD OVERRIDING IN JAVA

If subclass (child class) has the same method as declared in the parent class, it is known as **method overriding in Java**. In other words, if a subclass provides the specific implementation of the method that has been declared by one of its parent classes, it is known as method overriding.

# INHERITANCE

# Inheritance in Java is a concept that acquires the properties from one class to other classes; for example, the relationship between father and son. Inheritance in Java is a process of acquiring all the behaviours of a parent object.

# Single inheritance.

# PACKAGES

A package in Java is used to group related classes. Think of it as **a folder in a file directory**. We use packages to avoid name conflicts, and to write a better maintainable code. Packages are divided into two categories:

* User-defined Packages (create your own packages)

### **User-defined packages**

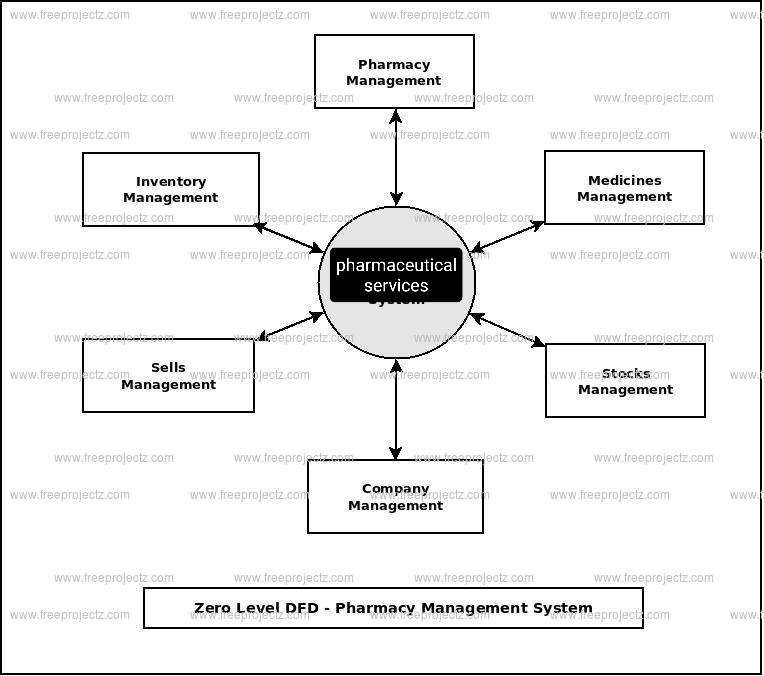
User-defined packages are those that developers create to incorporate different needs of applications. In simple terms, User-defined packages are those that the users define. Inside a package, you can have Java files like classes, interfaces, and a package as well (called a sub-package).

* EXCEPTION

The **Exception Handling in Java** is one of the powerful mechanisms to handle the run-time errors so that the normal flow of the application can be maintained.

8

# DESIGN



**IMPLEMENTATION**

**Code:**

import billing.Product;

import java.text.SimpleDateFormat;

import java.util.\*;

class displaypopup{ //class

displaypopup(){ //constructor

System.out.println("WELCOME TO PHARMACY");

}

void print1(int a){ //method-overloading

System.out.println("MEDICINE ADDED SUCCESSFUL:)"); //Method overloading

}

void print1(char b){ //final

System.out.println("HAVE A NICE DAY:)");

}

}

class ShoppingBill extends billing.Product //inharitance

{

public ShoppingBill(String id, String pname, int qty, double price, double totalPrice) {

super(id, pname, qty, price, totalPrice);

}

public static void main(String args[])

{

displaypopup ds = new displaypopup( );

// variables

String id = null;

String productName = null;

int quantity = 0;

double price = 0.0;

double totalPrice = 0.0;

double overAllPrice = 0.0;

double cgst, sgst, subtotal=0.0, discount=0.0;

char choice = '\0';

final int sgstp=12;

final int cgstp=12;

System.out.println("\t\t\t\t------------------------------------------------------------------");

//format of date and time

SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");

Date date = new Date(); //object

Calendar calendar = Calendar.getInstance();

String[] days = new String[] { "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday" };

//prints current date and time

System.out.println("Date: "+formatter.format(date)+" "+days[calendar.get(Calendar.DAY\_OF\_WEEK) - 1]+"\t\t\t\t\t\t (+91) 9998887770");

Scanner scan = new Scanner(System.in);

System.out.print("Enter GroupMember Name: ");

String customername=scan.nextLine();

//create Scanner class object

//creating an ArrayList to store the product

List<Product> product = new ArrayList<Product>();

do

{

// read input values

System.out.println("Enter the medicine details: ");

System.out.print("medicine ID: ");

id = scan.nextLine();

System.out.print("Medicine Name: ");

productName = scan.nextLine();

System.out.print("Quantity: ");

quantity = scan.nextInt();

System.out.print("Price (per unit): ");

price = scan.nextDouble();

ds.print1(1);

//calculate total price for a particular product

totalPrice = price \* quantity;

//calculates overall price

overAllPrice = overAllPrice + totalPrice;

//creates Product class object and add it to the List

product.add( new Product(id, productName, quantity, price, totalPrice) );

// ask for continue shopping?

System.out.print("Want to add more medicines? (y or n): ");

//reads a character Y or N

choice = scan.next().charAt(0);

//read remaining characters, don't store (no use)

scan.nextLine();

}

while (choice == 'y' || choice == 'Y');

//display all product with its properties

Product.displayFormat();

for (Product p : product)

{

p.display();

}

//price calculation

System.out.println("\n\t\t\t\t\t\t\t\t\t\tTotal Amount (Rs.) " +overAllPrice);

//calculating discount

discount = overAllPrice\*2/100;

System.out.println("\n\t\t\t\t\t\t\t\t\t\t Discount (Rs.) " +discount);

//total amount after discount

subtotal = overAllPrice-discount;

System.out.println("\n\t\t\t\t\t\t\t\t\t\t Subtotal "+subtotal);

//calculating tax

sgst=overAllPrice\*sgstp/100;

System.out.println("\n\t\t\t\t\t\t\t\t\t\t SGST (%) "+sgst);

cgst=overAllPrice\*cgstp/100;

System.out.println("\n\t\t\t\t\t\t\t\t\t\t CGST (%) "+cgst);

//calculating amount to be paid by buyer

System.out.println("\n\t\t\t\t\t\t\t\t\t\t Invoice Total " +(subtotal+cgst+sgst));

ds.print1('a');

System.out.println("\t\t\t\t----------------Thank You !!-----------------");

print1();

// close Scanner

scan.close();

}

static void print1(){ //method overriding

System.out.println("See you soon...");

}

}

PACKAGE:

package billing; //package

public class Product //class

{

// properties

private String id; //access modifier

private String pname;

private int qty;

private double price;

private double totalPrice; // constructor

public Product(String id, String pname, int qty, double price, double totalPrice)

{

this.id=id; //this keyword

this.pname = pname;

this.qty = qty;

this.price = price;

this.totalPrice = totalPrice;

}

// getter methods

public String getId()

{

return id;

}

public String getPname()

{

return pname;

}

public int getQty()

{

return qty;

}

public double getPrice()

{

return price;

}

public double getTotalPrice()

{

return totalPrice;

}

//displayFormat

public static void displayFormat() //static

{

System.out.format("-----------------------------------------------------------------------------------------------------------------------------------");

System.out.print("\nProduct ID \t\tName\t\tQuantity\t\tRate \t\t\t\tTotal Price\n");

System.out.format("-----------------------------------------------------------------------------------------------------------------------------------\n");

}

// display

public void display()

{

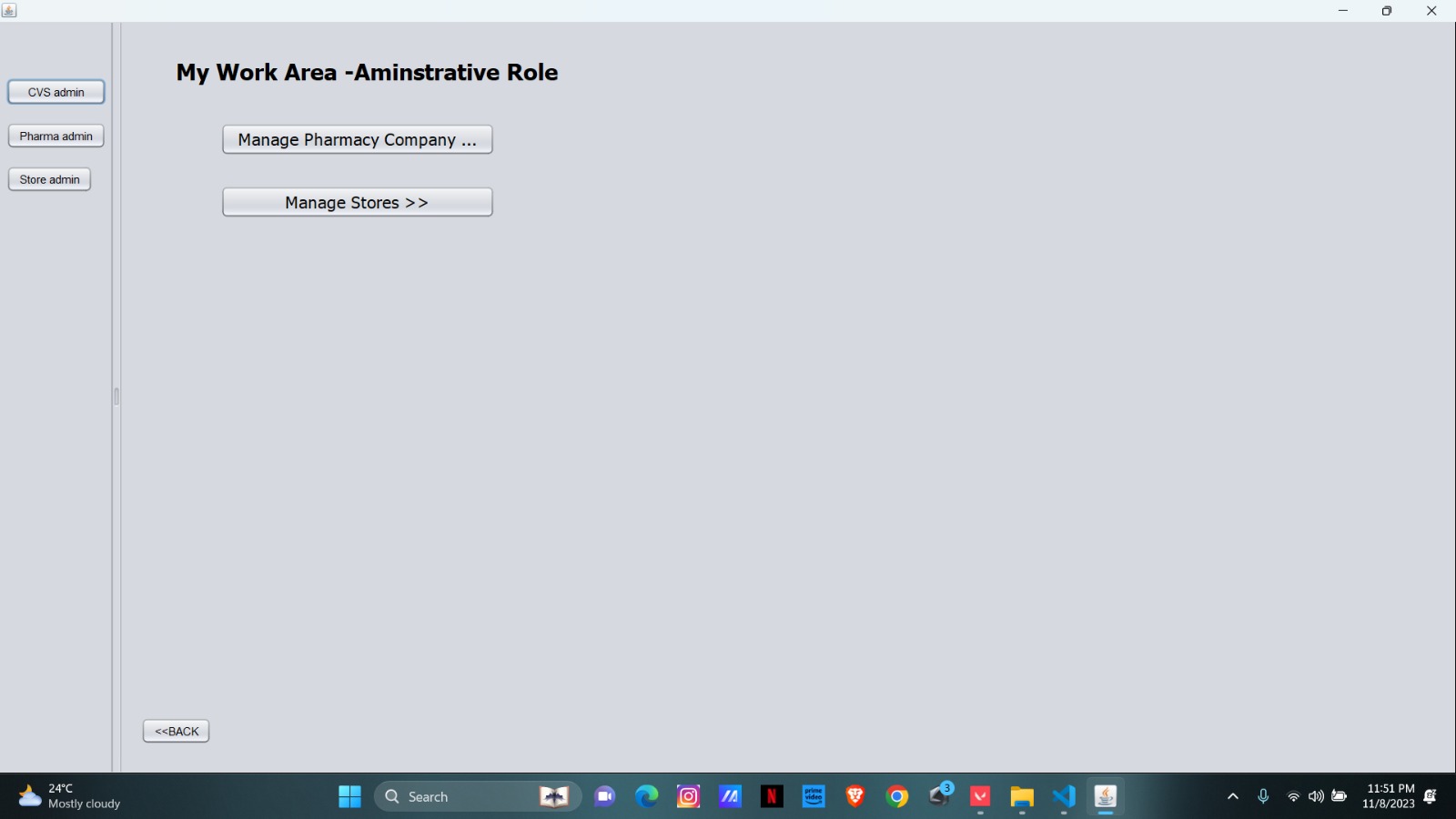
System.out.format(" %-9s %-9s %5d %9.2f %14.2f\n" ,id, pname, qty, price, totalPrice);

}

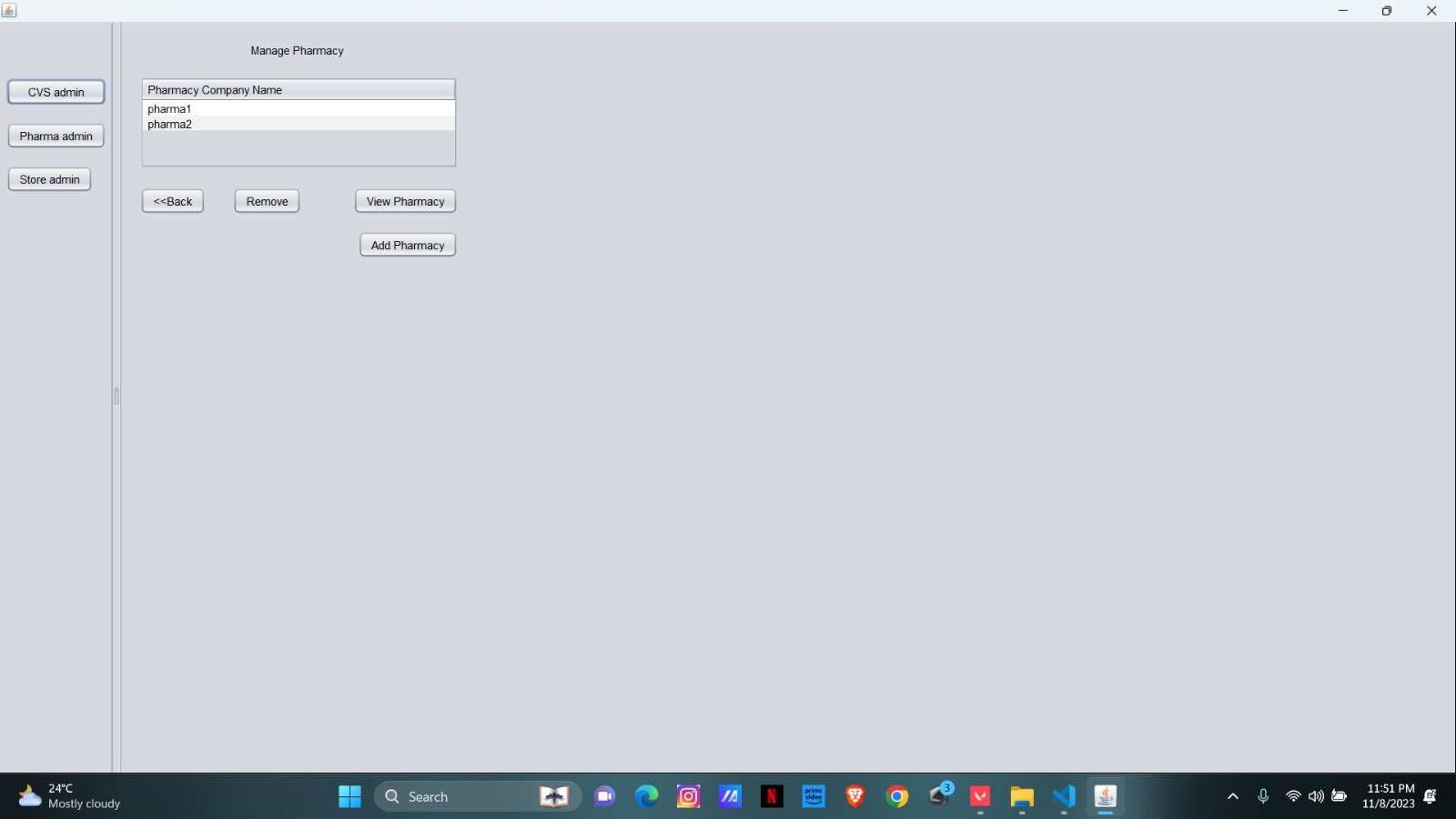
}

# RESULT SCREENS

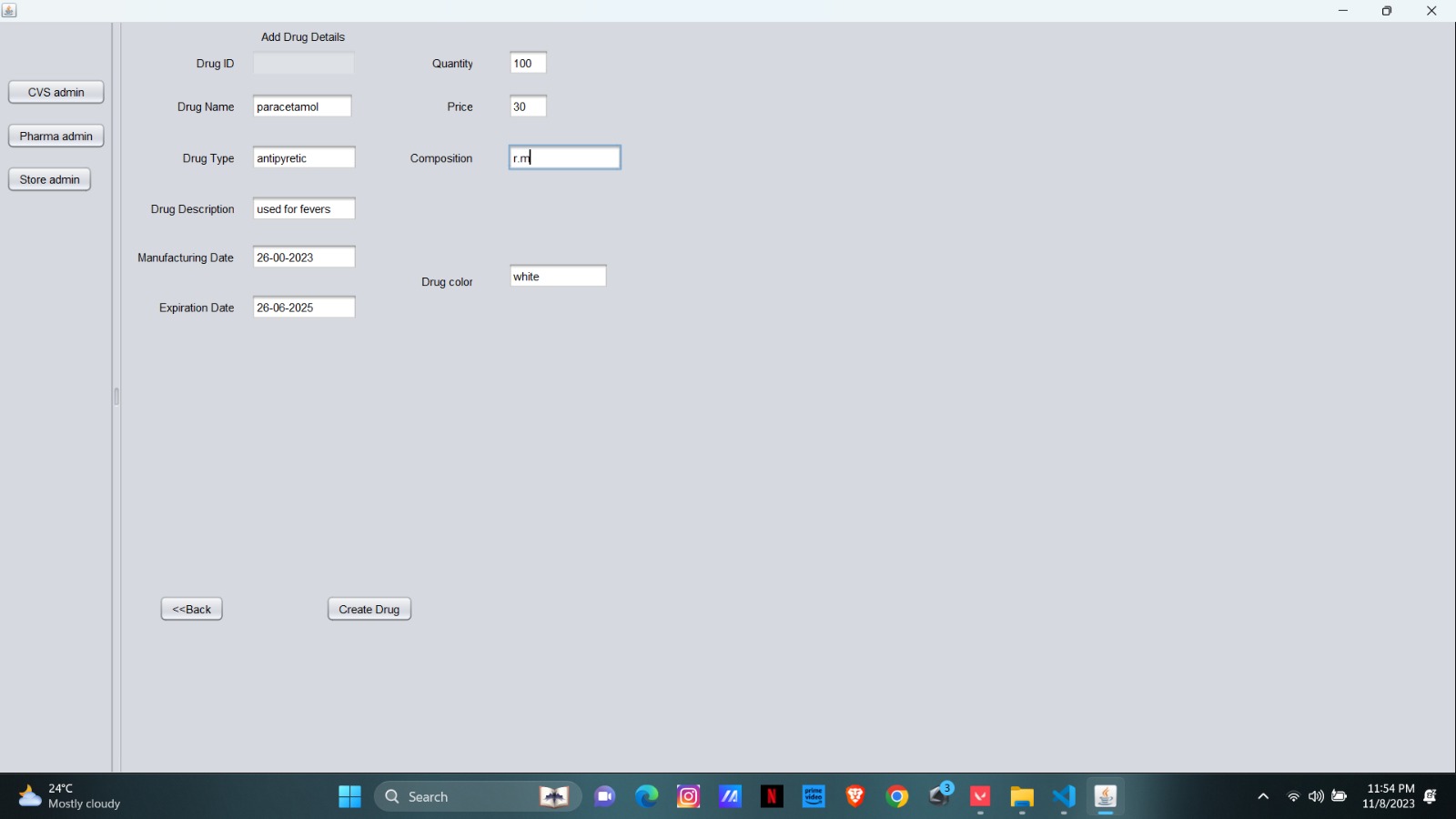
1.GUI Interface



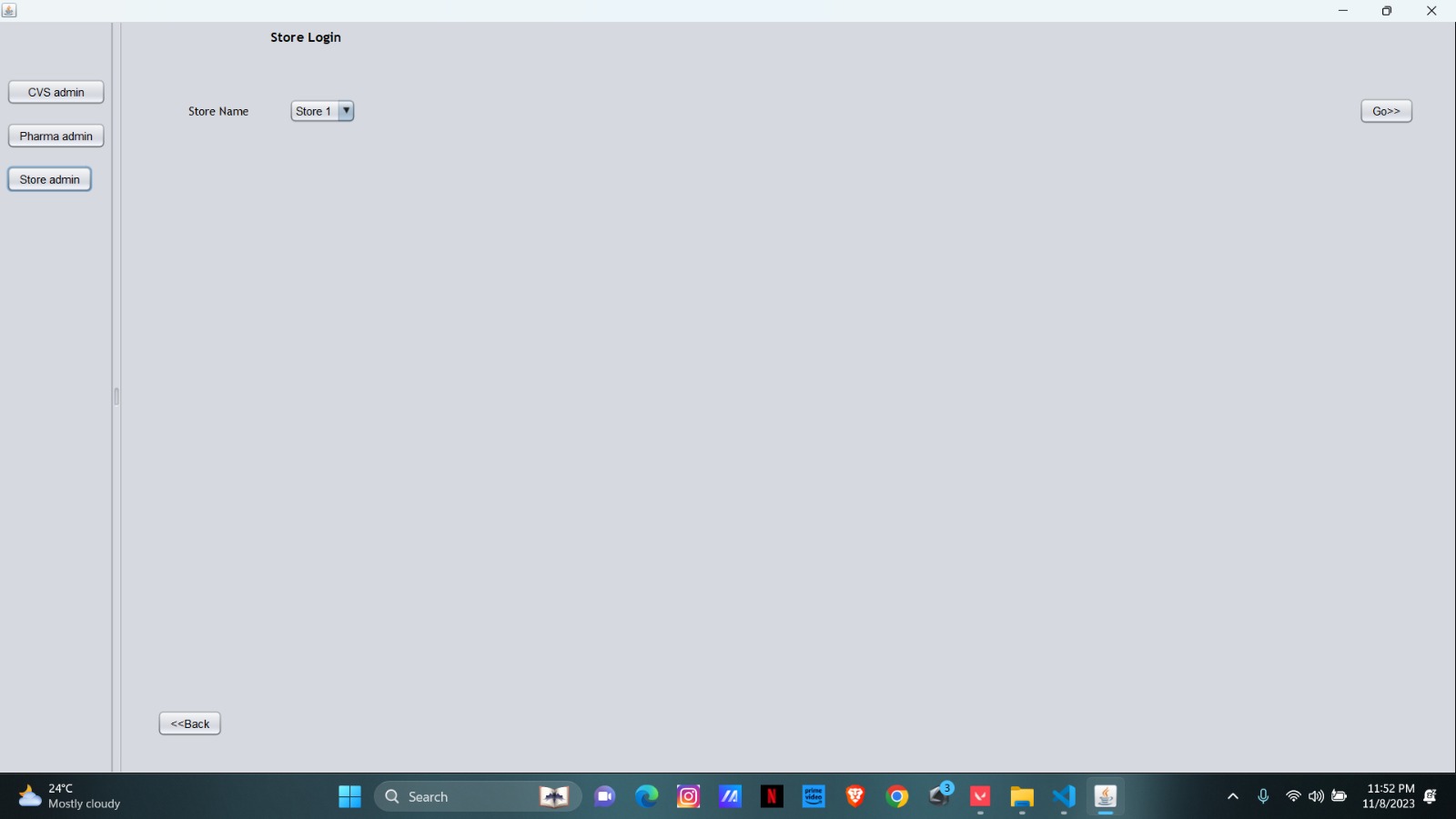
2.Pharmacy management

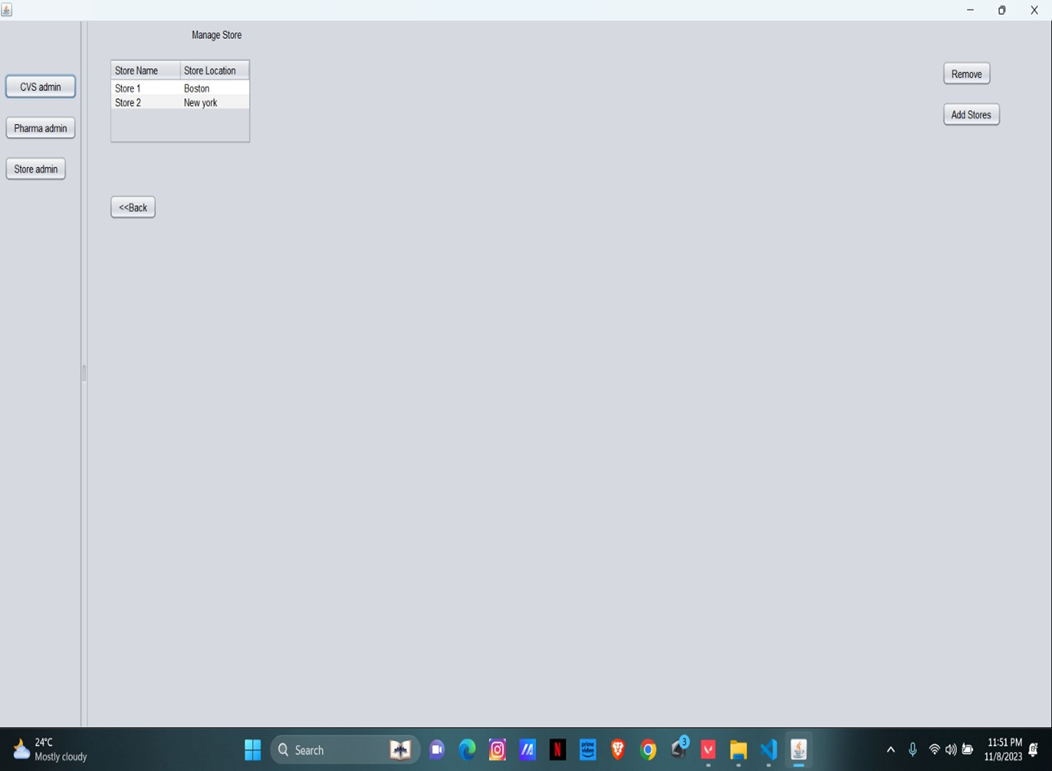


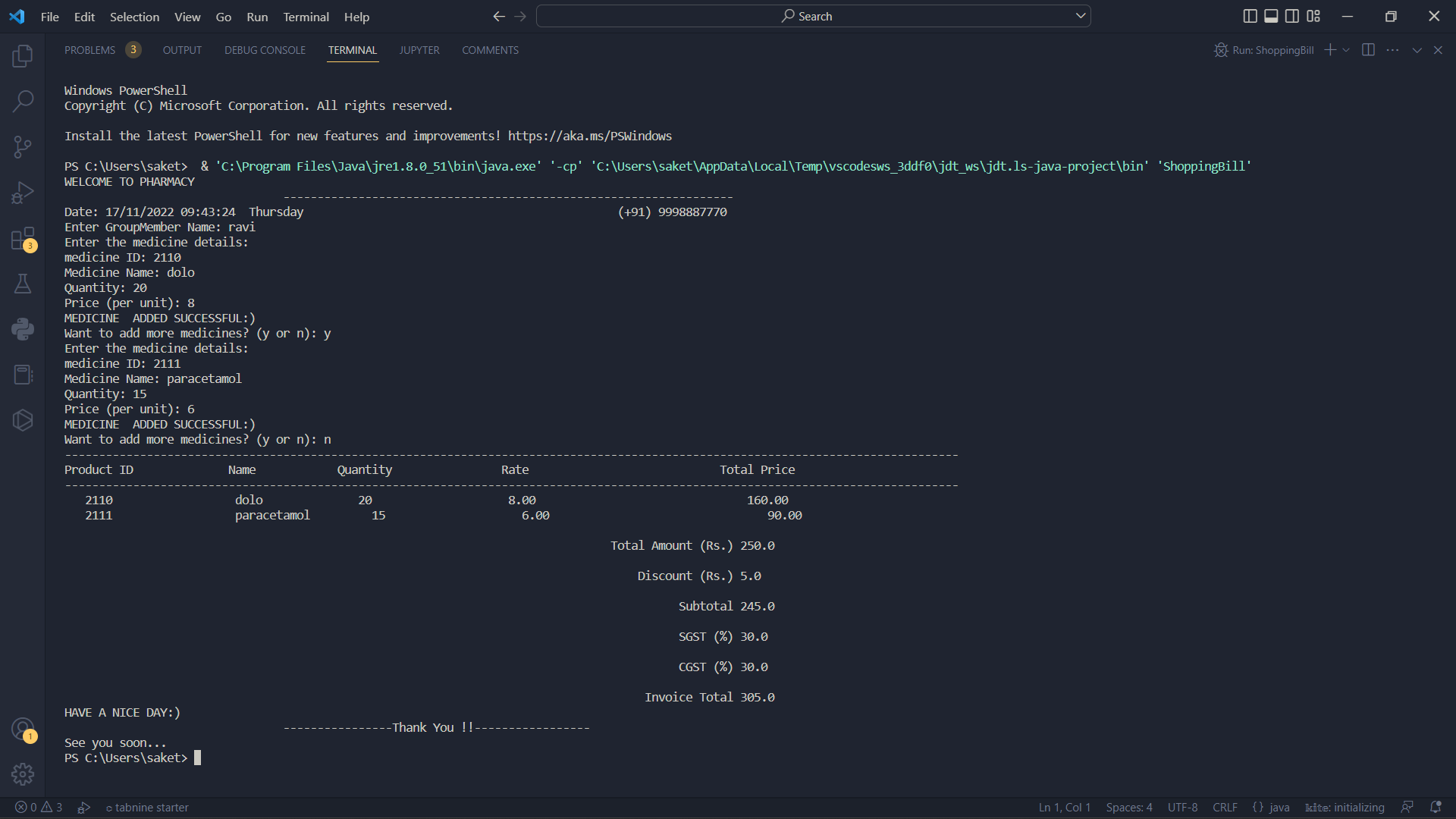
3.Drug details



4.Store details





**RESULT**

# CONCLUSION

This new pharmaceutical services is just like workflow software which can be treated as full version software which has generic modules to prevent underpayment that makes efficient business system. Its automatic batch processing system saves time and money for the user who will use this system through with excellent graphical user interface. Upon all, to meet real time environment working process, this system has automated durable medical equipment billing system. To make non stoppable development process, its full interaction checkup provides the flexibility of drug interaction and status, does checking, allergy checking report and medical reports, duplicate therapy and private patient messages.