



# PHARMACY MANAGEMENT SYSTEM



## A PROJECT REPORT

*Submitted by*

**GAYATHRI M (2303811710422045)**

*in partial fulfillment of requirements for the award of the course*

**CGB1201 - JAVA PROGRAMMING**

*In*

**COMPUTER SCIENCE AND ENGINEERING**

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY**

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

**SAMAYAPURAM – 621 112**

**NOVEMBER- 2024**

# K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621 112

## BONAFIDE CERTIFICATE

Certified that this project report on “**PHARMACY MANAGEMENT SYSTEM**” is the bonafide work of **STUDENT NAME (2303811710422045)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

CGB1201-JAVA PROGRAMMING  
Dr.A.DELPHIN CAROLINA RANI, M.E., Ph.D.,  
HEAD OF THE DEPARTMENT  
PROFESSOR

### SIGNATURE

Dr.A.Delphin Carolina Rani, M.E.,Ph.D.,

### HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology  
(Autonomous)

Samayapuram–621112.

CGB1201-JAVA PROGRAMMING  
Mr. M. SARAVANAN, M.E.,  
SUPERVISOR  
ASSISTANT PROFESSOR

### SIGNATURE

Mr. M. Saravanan, M.E.,

### SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology  
(Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on 02/12/2024

CGB1201-JAVA PROGRAMMING  
Mr. MADHUKANNAN A. M.E.,  
INTERNAL EXAMINER  
ASSISTANT PROFESSOR

INTERNAL EXAMINER

CGB1201-JAVA PROGRAMMING  
Dr. S. SETHAMIL SELVI, M.E., Ph.D.,  
EXTERNAL EXAMINER  
PROFESSOR  
8138-SCE, TRICHY.

EXTERNAL EXAMINER

## DECLARATION

I declare that the project report on “**PHARMACY MANAGEMENT SYSTEM**” is the result of original work done by us and best of our knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201-JAVA PROGRAMMING**.



**Signature**

GAYATHRI M

---

Place: Samayapuram

Date:02/12/2024

## ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution “**K.Ramakrishnan College of Technology (Autonomous)**”, for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN, B.E.**, for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.**, Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. A. DELPHIN CAROLINA RANI, M.E.,Ph.D.**, Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Mr. M. SARAVANAN, M.E.**, Department of **COMPUTER SCIENCE AND ENGINEERING**, for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

## **VISION OF THE INSTITUTION**

To serve the society by offering top-notch technical education on par with global standards

## **MISSION OF THE INSTITUTION**

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities
- Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

## **VISION OF DEPARTMENT**

To be a center of eminence in creating competent software professionals with research and innovative skills.

## **MISSION OF DEPARTMENT**

**M1: Industry Specific:** To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

**M2: Research:** To prepare students for research-oriented activities.

**M3: Society:** To empower students with the required skills to solve complex technological problems of society.

## **PROGRAM EDUCATIONAL OBJECTIVES**

### **1. PEO1: Domain Knowledge**

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

### **2. PEO2: Employability Skills and Research**

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

### **3. PEO3: Ethics and Values**

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

## **PROGRAM SPECIFIC OUTCOMES (PSOs)**

### **PSO 1: Domain Knowledge**

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

### **PSO 2: Quality Software**

To apply software engineering principles and practices for developing quality software for scientific and business applications.

### **PSO 3: Innovation Ideas**

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

## **PROGRAM OUTCOMES (POs)**

Engineering students will be able to:

- 1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- 2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
- 3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
- 4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

- 5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
- 6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- 10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- 11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- 12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## **ABSTRACT**

A Pharmacy Management System is a specialized software solution designed to streamline and optimize the operations of a pharmacy, focusing on efficient handling of prescription processing. The system provides an integrated platform to manage various aspects of pharmacy operations, including prescription verification, medication dispensing, and maintaining accurate records. The primary objective of the system is to enhance workflow efficiency, reduce the likelihood of errors, and ensure compliance with regulatory standards. Key features of the system include automated prescription verification to ensure accuracy and legitimacy, efficient dispensing workflows to reduce wait times, and secure record-keeping for patient and prescription data to safeguard privacy and meet legal requirements. This system also incorporates inventory management to track stock levels and avoid medication shortages, reporting and analytics to provide insights into sales trends and operational performance, and integration with external systems like healthcare providers and insurance databases for seamless communication. By digitizing and automating critical processes, the Pharmacy Management System minimizes manual interventions, reduces operational costs, and improves the quality of service provided to patients. It ultimately supports pharmacists in delivering safe, accurate, and efficient care while enhancing customer satisfaction and regulatory adherence. The Pharmacy Management System (PMS) is a comprehensive solution designed to address the challenges of modern pharmacy operations. It focuses on streamlining the prescription processing workflow, ensuring medication accuracy, and maintaining secure, accessible records. The system offers a blend of automation and intelligent management tools to simplify routine tasks while ensuring regulatory compliance and patient safety. Key components of the system include prescription verification, which ensures that prescriptions are authentic and adhere to established medical guidelines, and dispensing management, which optimizes the dispensing process to minimize errors and enhance speed.



## ABSTRACT WITH POs AND PSOs MAPPING

### CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
<p>A Pharmacy Management System is a specialized software solution designed to streamline and optimize the operations of a pharmacy, focusing on efficient handling of prescription processing. The system provides an integrated platform to manage various aspects of pharmacy operations, including prescription verification, medication dispensing, and maintaining accurate records.</p> <p>The primary objective of the system is to enhance workflow efficiency, reduce the likelihood of errors, and ensure compliance with regulatory standards. Key features of the system include automated prescription verification to ensure accuracy and legitimacy, efficient dispensing workflows to reduce wait times, and secure record-keeping for patient and prescription data to safeguard privacy and meet legal requirements.</p>	<p><b>PO1 -3</b></p> <p><b>PO2 -3</b></p> <p><b>PO3 -3</b></p> <p><b>PO4 -3</b></p> <p><b>PO5 -3</b></p> <p><b>PO6 -3</b></p> <p><b>PO7 -3</b></p> <p><b>PO8 -3</b></p> <p><b>PO9 -3</b></p> <p><b>PO10 -3</b></p> <p><b>PO11-3</b></p> <p><b>PO12 -3</b></p>	<p><b>PSO1 -3</b></p> <p><b>PSO2 -3</b></p> <p><b>PSO3 -3</b></p>

Note: 1- Low, 2-Medium, 3- High

## **TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>ABSTRACT</b>	8
<b>1</b>	<b>INTRODUCTION</b>	1
	1.1 Objective	1
	1.2 Overview	1
	1.3 Java Programming concepts	
<b>2</b>	<b>PROJECT METHODOLOGY</b>	1
	2.1 Proposed Work	2
	2.2 Block Diagram	1
<b>3</b>	<b>MODULE DESCRIPTION</b>	2
	<b>3.1 Prescription Module</b>	1
	3.2 Prescription Management Module	3
	3.3 User Interaction Module	1
	3.4 Data Storage Module	3
<b>4</b>	<b>CONCLUSION &amp; FUTURE SCOPE</b>	1
	4.1 Conclusion	4
	4.2 Future Scope	1
	<b>REFERENCES</b>	4
	<b>APPENDIX A (SOURCE CODE)</b>	1
	<b>APPENDIX B (SCREENSHOTS)</b>	5

## **CHAPTER 1**

# INTRODUCTION

## 1.1 Objective

An objective for a Pharmacy Management System (PMS) can outline the purpose and goals of implementing such a system. Here's a concise objective. To streamline and automate pharmacy operations, ensuring efficient management of inventory, prescriptions, and sales while enhancing patient safety, compliance with regulations, and overall operational productivity.

## 1.2 Overview

The Pharmacy Management System (PMS) is an advanced software application designed to streamline and enhance the operational efficiency of pharmacies. This system aims to address the critical aspects of prescription processing, inventory management, record-keeping, and customer interaction while ensuring regulatory compliance and improving the overall patient experience.

## 1.3 Java Programming Concepts

Object-Oriented Programming (OOP) is a programming paradigm centered around the concept of "objects," which encapsulate data and behavior. Below are the key concepts of OOP and their relevance to the Pharmacy Management System project.

### 1. Class and Object

- **Class:** A blueprint or template for creating objects. It defines the properties (attributes) and behaviors (methods) of the objects.
- **Object:** An instance of a class, representing an entity with specific values for its attributes.

### In the Project:

- Classes can represent entities like Prescription, Pharmacist, Patient, Medicine, and Inventory.
- For example, the Medicine class may have attributes like name, price, expiry\_date, and quantity and methods like updateStock() or checkExpiry().

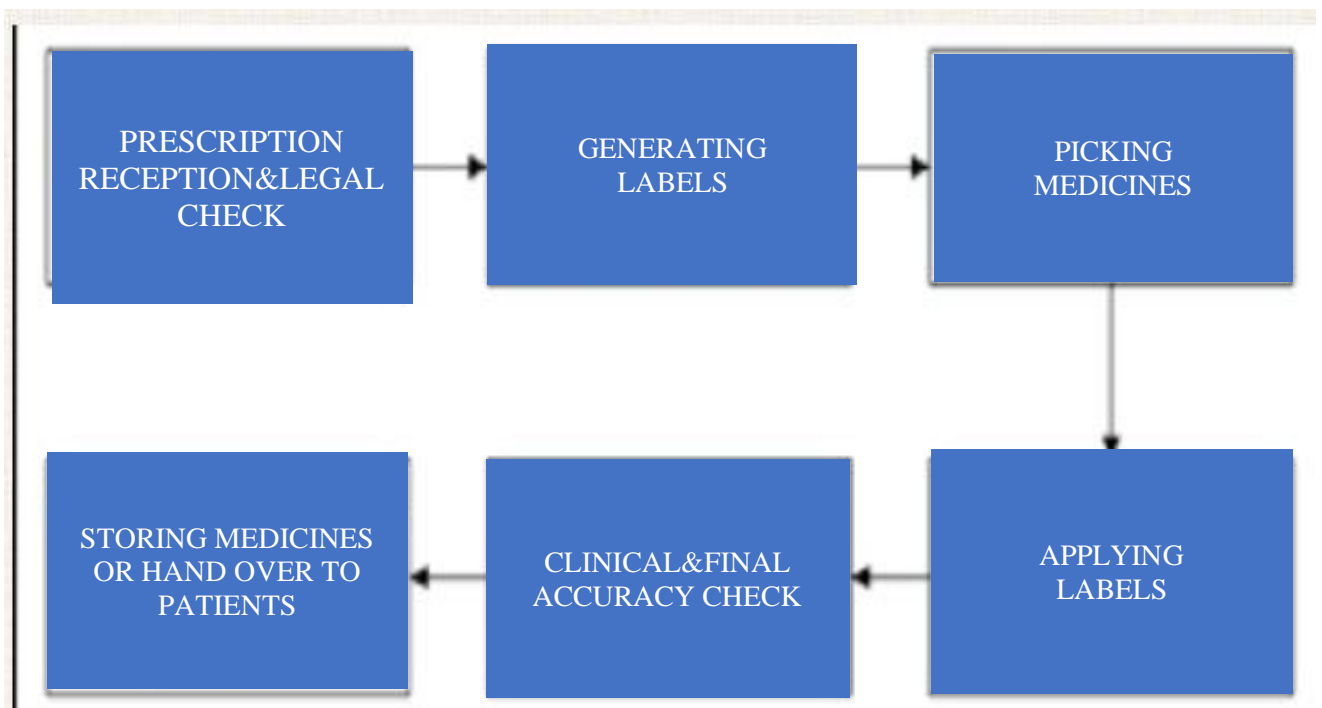
## CHAPTER 2

### PROJECT METHODOLOGY

#### 2.1 Proposed Work

The proposed Pharmacy Management System aims to design and implement a comprehensive solution to streamline pharmacy operations, ensuring efficiency, accuracy, and better patient care. The system will cater to prescription handling, inventory management, billing, and regulatory compliance while improving the overall workflow

#### 2.2 Block Diagram



## **CHAPTER 3**

### **MODULE DESCRIPTION**

#### **3.1 Prescription Module**

The Prescription Module is a core component of the Pharmacy Management System, designed to handle the entire lifecycle of prescription management, from receipt and verification to dispensing and record-keeping. This module ensures the accuracy of medication dispensation and enhances workflow efficiency for pharmacists

#### **3.2 Prescription Management Module**

The Prescription Management Module is a key component of the Pharmacy Management System, responsible for managing the entire lifecycle of prescriptions. From receiving prescriptions to dispensing medications and maintaining records, this module is crucial for ensuring the accuracy, efficiency, and safety of pharmacy operations. Here is a detailed breakdown of the Prescription Management Module.

#### **3.3 User Interaction Module (Main Menu)**

This is where the user provides information or instructions, like typing text or using voice input. It can include fields for text, checkboxes, or even complex forms depending on the application. The User Interaction Module (UIM) refers to the system or interface that facilitates communication between a user and a software or application. It serves as a bridge, allowing users to input commands, ask questions, or interact with the system in various ways

#### **3.4 Data Storage Module**

The Data Storage Module is a crucial component in any system or application that deals with the storage, retrieval, and management of data. It serves as the backbone for saving and accessing information persistently. This module is responsible for organizing data in a way that allows the system to retrieve and manipulate it efficiently.

## **CHAPTER-4**

### **CONCLUSION & FUTURE SCOPE**

#### **4.1 CONCLUSION**

The Pharmacy Management System is a well-structured Java application that effectively demonstrates the use of object-oriented programming principles to solve real-world problems. It streamlines the management of medical prescriptions through a modular design, enabling functionalities such as adding, verifying, dispensing, and viewing prescriptions. The system ensures operational integrity by enforcing rules like mandatory verification before dispensing, preventing errors and improving safety in prescription handling.

#### **4.2 FUTURE SCOPE**

##### **1. Integration with AI and Machine Learning**

- **AI-Driven Data Optimization:** Machine learning algorithms could be used to predict and automate data management tasks such as storage scaling, archiving, and load balancing based on usage patterns.
- **Data-Driven Insights:** With advanced AI analytics, data storage systems can offer deeper insights into data usage, trends, and performance bottlenecks, making it easier for users to optimize storage without manual intervention.

##### **2. Decentralized and Blockchain-Based Storage**

- **Blockchain Integration:** With the rise of blockchain technology, decentralized data storage could become more mainstream. Distributed storage on blockchain platforms can provide higher security, data integrity, and transparency, as every transaction (data access, change, or deletion) would be securely recorded.
- **Decentralized Cloud Storage:** Emerging platforms like Filecoin and Storj aim to provide decentralized cloud storage, where data is split into smaller pieces and stored across

## REFERENCES

### 1. **Object-Oriented Programming (OOP)**

- Use Classes and Objects to model real-world entities like Medicine, Customer, Order.
- Implement Encapsulation for data protection, Inheritance for reusable logic, and Polymorphism for dynamic behaviors.

### 2. **Data Structures and Collections**

- Use Lists (e.g., ArrayList) to store and manage collections of medicines or orders.
- Use Maps (e.g., HashMap) for key-value pair data like tracking medicine stock (medicineId -> quantity).

### 3. **Database Connectivity with JDBC**

- Manage interactions with the database for CRUD operations.
- Use PreparedStatement for secure and efficient queries.

### 4. **Java Swing/JavaFX**

- Build the user interface for interactions like searching for medicines, generating bills, and managing inventory.

### 5. **Exception Handling**

- Handle runtime errors such as invalid input, database connection failures, or out-of-stock scenarios using try-catch-finally

## **APPENDIX A**

### **(SOURCE CODE)**

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class PharmacyManagementSystem extends JFrame implements ActionListener {

    // Components for the GUI
    private JTextField medicineNameField, medicinePriceField, medicineQuantityField;
    private JButton addButton, updateButton, deleteButton, clearButton;
    private JTable medicineTable;
    private DefaultTableModel tableModel;

    public PharmacyManagementSystem() {
        // Setting up the Frame
        setTitle("Pharmacy Management System");
        setSize(700, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        // Panel for input fields and buttons
        JPanel inputPanel = new JPanel(new GridLayout(5, 2, 10, 10));

        // Creating input fields
        inputPanel.add(new JLabel("Medicine Name:"));
        medicineNameField = new JTextField();
        inputPanel.add(medicineNameField);

        inputPanel.add(new JLabel("Price:"));
```



```

medicinePriceField = new JTextField();
inputPanel.add(medicinePriceField);

inputPanel.add(new JLabel("Quantity:"));
medicineQuantityField = new JTextField();
inputPanel.add(medicineQuantityField);

// Adding buttons
addButton = new JButton("Add");
updateButton = new JButton("Update");
deleteButton = new JButton("Delete");
clearButton = new JButton("Clear");

addButton.addActionListener(this);
updateButton.addActionListener(this);
deleteButton.addActionListener(this);
clearButton.addActionListener(this);

inputPanel.add(addButton);
inputPanel.add(updateButton);
inputPanel.add(deleteButton);
inputPanel.add(clearButton);

// Table for displaying medicine data with an additional column for Total Price
tableModel = new DefaultTableModel(new String[]{"Medicine Name", "Price", "Quantity",
"Total Price"}, 0);
medicineTable = new JTable(tableModel);
JScrollPane tableScrollPane = new JScrollPane(medicineTable);

// Adding components to the frame
setLayout(new BorderLayout());
add(inputPanel, BorderLayout.NORTH);
add(tableScrollPane, BorderLayout.CENTER);
}

```

```

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == addButton) {
        addMedicine();
    } else if (e.getSource() == updateButton) {
        updateMedicine();
    } else if (e.getSource() == deleteButton) {
        deleteMedicine();
    } else if (e.getSource() == clearButton) {
        clearFields();
    }
}

// Method to add a new medicine to the table
private void addMedicine() {
    String name = medicineNameField.getText();
    String priceText = medicinePriceField.getText();
    String quantityText = medicineQuantityField.getText();

    if (!name.isEmpty() && !priceText.isEmpty() && !quantityText.isEmpty()) {
        try {
            double price = Double.parseDouble(priceText);
            int quantity = Integer.parseInt(quantityText);
            double totalPrice = price * quantity;

            tableModel.addRow(new Object[]{name, price, quantity, totalPrice});
            clearFields();
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(this, "Please enter valid numbers for price and
quantity", "Error", JOptionPane.ERROR_MESSAGE);
        }
    } else {
        JOptionPane.showMessageDialog(this, "Please fill in all fields", "Error",

```

```

JOptionPane.ERROR_MESSAGE);
    }
}

// Method to update the selected medicine in the table
private void updateMedicine() {
    int selectedRow = medicineTable.getSelectedRow();
    if (selectedRow >= 0) {
        String name = medicineNameField.getText();
        String priceText = medicinePriceField.getText();
        String quantityText = medicineQuantityField.getText();

        if (!name.isEmpty() && !priceText.isEmpty() && !quantityText.isEmpty()) {
            try {
                double price = Double.parseDouble(priceText);
                int quantity = Integer.parseInt(quantityText);
                double totalPrice = price * quantity;

                tableModel.setValueAt(name, selectedRow, 0);
                tableModel.setValueAt(price, selectedRow, 1);
                tableModel.setValueAt(quantity, selectedRow, 2);
                tableModel.setValueAt(totalPrice, selectedRow, 3);
                clearFields();
            } catch (NumberFormatException ex) {
                JOptionPane.showMessageDialog(this, "Please enter valid numbers for price and
quantity", "Error", JOptionPane.ERROR_MESSAGE);
            }
        } else {
            JOptionPane.showMessageDialog(this, "Please fill in all fields", "Error",
JOptionPane.ERROR_MESSAGE);
        }
    } else {
        JOptionPane.showMessageDialog(this, "Please select a row to update", "Error",
JOptionPane.ERROR_MESSAGE);
    }
}

```

```

    }
}

// Method to delete the selected medicine from the table
private void deleteMedicine() {
    int selectedRow = medicineTable.getSelectedRow();
    if (selectedRow >= 0) {
        tableModel.removeRow(selectedRow);
    } else {
        JOptionPane.showMessageDialog(this, "Please select a row to delete", "Error",
JOptionPane.ERROR_MESSAGE);
    }
}

// Method to clear input fields
private void clearFields() {
    medicineNameField.setText("");
    medicinePriceField.setText("");
    medicineQuantityField.setText("");
}

// Main method to run the application
public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> {
        PharmacyManagementSystem frame = new PharmacyManagementSystem();
        frame.setVisible(true);
    });
}
}

```

## APPENDIX B

### (SCREENSHOTS)

Pharmacy Management System

Medicine Name:

Price:

Quantity:

Medicine Name	Price	Quantity	Total Price
---------------	-------	----------	-------------

Pharmacy Management System

Medicine Name: DOLO

Price:

Quantity:

Medicine Name	Price	Quantity	Total Price
---------------	-------	----------	-------------

Pharmacy Management System

Medicine Name:

Price:

Quantity:

Medicine Name	Price	Quantity	Total Price
DOLO	3.0	4	12.0

Pharmacy Management System

Medicine Name: PARACETAMOL

Price: 4

Quantity: 5

Add Update

Delete Clear

Medicine Name	Price	Quantity	Total Price
DOLO	3.0	4	12.0

Pharmacy Management System

Medicine Name:

Price:

Quantity:

Add Update

Delete Clear

Medicine Name	Price	Quantity	Total Price
DOLO	3.0	4	12.0
PARACETAMOL	4.0	5	20.0

