Pharmacy Management System

A MINI-PROJECT BY:

`

SARVESH R	230701294
SIDDARTH SAKTHI M	230701314

in partial fulfillment of the award of the degree

OF

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2024

BONAFIDE CERTIFICATE

Certified that this project “Pharmacy Management System” is the bonafide work of **“SARVESH R , SIDDARTH SAKTHI M”** who carried out the project work under my supervision.

Submitted for the practical examination held on ____________________

SIGNATURE

Mr.G SARAVANA GOKUL
Professor and Academic Head,
Computer Science and Engineering,
Rajalakshmi Engineering College
(Autonomous),
Thandalam,Chennai-602105

SIGNATURE

Ms.V.JANANEE
Assistant Professor(SG),
Computer Science and Engineering,
Rajalakshmi Engineering College
(Autonomous),
Thandalam,Chennai-602105

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

The Pharmacy Management System is a robust Java application designed to streamline the management of pharmaceutical inventories through an intuitive graphical user interface (GUI). Leveraging the Swing framework, this system employs a CardLayout to facilitate seamless navigation between the login and medicine inventory panels.

Key Technical Features:

- **User Authentication**: The system implements a basic login mechanism that verifies user credentials against hardcoded values. Upon successful authentication, users gain access to the medicine inventory panel, enhancing security and user management.
- **Medicine Management**: A HashMap is utilized to store and manage medicine details, including attributes such as name, price, quantity, expiry date, and image path. This allows for efficient retrieval and display of medicine information.
- **Dynamic User Interface**: The application features a responsive UI that adapts to user interactions. The login panel is designed with a dark theme, while the medicine inventory panel utilizes a light gray background for improved readability. Buttons are styled with color changes on hover to enhance user experience.
- **Interactive Medicine Display**: Users can view detailed information about each medicine through dynamically generated buttons. When clicked, these buttons trigger dialog boxes displaying essential medicine details such as price, quantity available, and expiry date.
- **Inventory Tracking:** The system maintains an up-to-date inventory by allowing users to view current stock levels for each medicine, ensuring that pharmacists can effectively manage supplies and make informed decisions regarding restocking

TABLE OF CONTENTS

INTRODUCTION

1.1 INTRODUCTION

Pharmacies play a crucial role in the healthcare sector, ensuring that medicines are accessible and accurately dispensed. However, manual processes in pharmacy management can lead to inefficiencies, stockouts, and errors in tracking inventory. To address these challenges, the

Pharmacy Management System introduces a technology- driven solution for automating and simplifying these operations.

This system focuses on:

1. **Inventory Management:** Keeping track of stock levels, batch numbers, and expiry dates.
2. **Sales Monitoring:** Recording transactions and generating reports for analysis (future scope).
3. **Medicine Information:** Providing detailed information about each medicine, including price, quantity, and expiry.
4. **User Authentication:** Ensuring secure access to the system for authorized users only.
5. **Alerts and Notifications:** Highlighting medicines with low stock or nearing expiry.

With its emphasis on scalability and ease of use, this system is designed to meet the needs of small to medium-sized pharmacies.

1.2-----Implementation

- Developed using **Java Swing**, which provides a graphical user interface for managing pharmacy operations.
- **MySQL** is used as the database management system for storing and retrieving data persistently

1.3----Scope of the Project

Efficiently manage medicine stocks, track medicine details, and provide secure access for pharmacists using a login system.

2. System Specifications

2.1 Hardware Specifications:

- Processor: Intel i3 or above.
- RAM: Minimum 4 GB.
- Hard Disk: At least 10 GB free space.

2.2 Software Specifications:

- Operating System: Windows/Linux/MacOS.
- Language: Java 8 or above.
- Database: MySQL (optional).

IDE: IntelliJ IDEA or Eclipse

SAMPLE CODE

SAMPLE DESIGN

```
import java.awt.*; import
javax.swing.*; import
java.sql.
import java.util.HashMap;

public class Main {
    public static void main(String[] args) {

        SwingUtilities.invokeLater(() -> new

MainWindow().createAndShowGUI());
    }
}
```

3.1 MainWindow Class:

```
class MainWindow {
    private JFrame frame;
    private CardLayout cardLayout; private
    JPanel mainPanel;
    private Dashboard dashboard;

    public void createAndShowGUI() {
        frame = new JFrame("Pharmacy Management System");
        frame.setSize(600, 400);

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        frame.getContentPane().setBackground(new Color(240, 240, 240));


        cardLayout = new CardLayout();

        mainPanel = new JPanel(cardLayout); dashboard =


        new Dashboard(mainPanel);


        mainPanel.add(new LoginPanel(dashboard),
"LoginPanel");

        mainPanel.add(new MedicinePanel(dashboard),
"MedicinePanel");


        frame.add(mainPanel);

        frame.setVisible(true);

    }

}
```

3.1 Dashboard Class:

```
    class Dashboard {

        private Connection connection; public


        Dashboard(JPanel mainPanel) {
```

```
        try {
            initializeDatabaseConnection();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private void initializeDatabaseConnection() throws SQLException {
        String url = "jdbc:mysql://localhost:3306/PharmacyDB"; String username
        = "root";
        String password = "password";
        connection = DriverManager.getConnection(url, username,
password);
    }

        String query = "SELECT name, price, quantity, expiry, imagePath
FROM medicines";

        try (Statement stmt = connection.createStatement();
            ResultSet rs = stmt.executeQuery(query)) { while
            (rs.next()) {
                String name = rs.getString("name"); double
                price = rs.getDouble("price"); int quantity =
                rs.getInt("quantity"); String expiry =
                rs.getString("expiry");
                String imagePath = rs.getString("imagePath");


                medicines.put(name, new Medicine(name, price, quantity,
expiry, imagePath));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }


        return medicines;
    }
}
```

3.2 LoginPanel Class:

```
class LoginPanel extends JPanel
 { private Dashboard dashboard;
```

```
public LoginPanel(Dashboard dashboard) {
    this.dashboard = dashboard;
    setLayout(new GridBagLayout());
    setBackground(new Color(30, 30, 30));

    GridBagConstraints gbc = new GridBagConstraints(); gbc.insets =
    new Insets(10, 10, 10, 10);

    JLabel titleLabel = new JLabel("PHARMACY
    MANAGEMENT SYSTEM");
    titleLabel.setFont(new Font("Arial", Font.BOLD, 24));
    titleLabel.setForeground(new Color(255, 215, 0));

    JLabel usernameLabel = new JLabel("Username:");
    usernameLabel.setForeground(Color.WHITE);

    JTextField usernameField = new JTextField(15);

    JLabel passwordLabel = new JLabel("Password:");
    passwordLabel.setForeground(Color.WHITE);
```

```
JPasswordField passwordField = new JPasswordField(15);

JButton loginButton = createLoginButton(usernameField,
passwordField);

gbc.gridx = 0;
gbc.gridy = 0;
gbc.gridwidth = 2;
add(titleLabel, gbc);

gbc.gridwidth = 1;
gbc.gridy++; gbc.gridx
= 0;
add(usernameLabel, gbc);

gbc.gridx = 1; add(usernameField,
gbc);

gbc.gridx = 0;
gbc.gridy++;
add(passwordLabel, gbc);
```

```
        gbc.gridx = 1; add(passwordField,
        gbc);


        gbc.gridy++; gbc.gridwidth =
        2;
        add(loginButton, gbc);

    }


    private JButton createLoginButton(JTextField usernameField,
    JPasswordField passwordField) {
        JButton loginButton = new JButton("Login");
        loginButton.setBackground(new Color(0x007BFF));
        loginButton.setForeground(Color.WHITE);


        loginButton.addActionListener(e -> {
            String username = usernameField.getText(); String
            password = new
String(passwordField.getPassword());
            if (validateLogin(username, password)) { ((CardLayout)
getParent().getLayout()).show(getParent(), "MedicinePanel");
                showWelcomeMessage();
            } else {
}JOptionPane.showMessageDialog(this,  JOptionPane.ERROR_MESSAGE);
"Invalid login!", "Login Failed",
                return loginButton;

    }


    private boolean validateLogin(String username, String password) {

        return "admin".equals(username) &&
 "password".equals(password);

    }


    private void showWelcomeMessage() {

        JOptionPane.showMessageDialog(this,

                "Welcome to the Pharmacy Management System!", "Welcome",

                    JOptionPane.INFORMATION_MESSAGE);

    }

 }
```

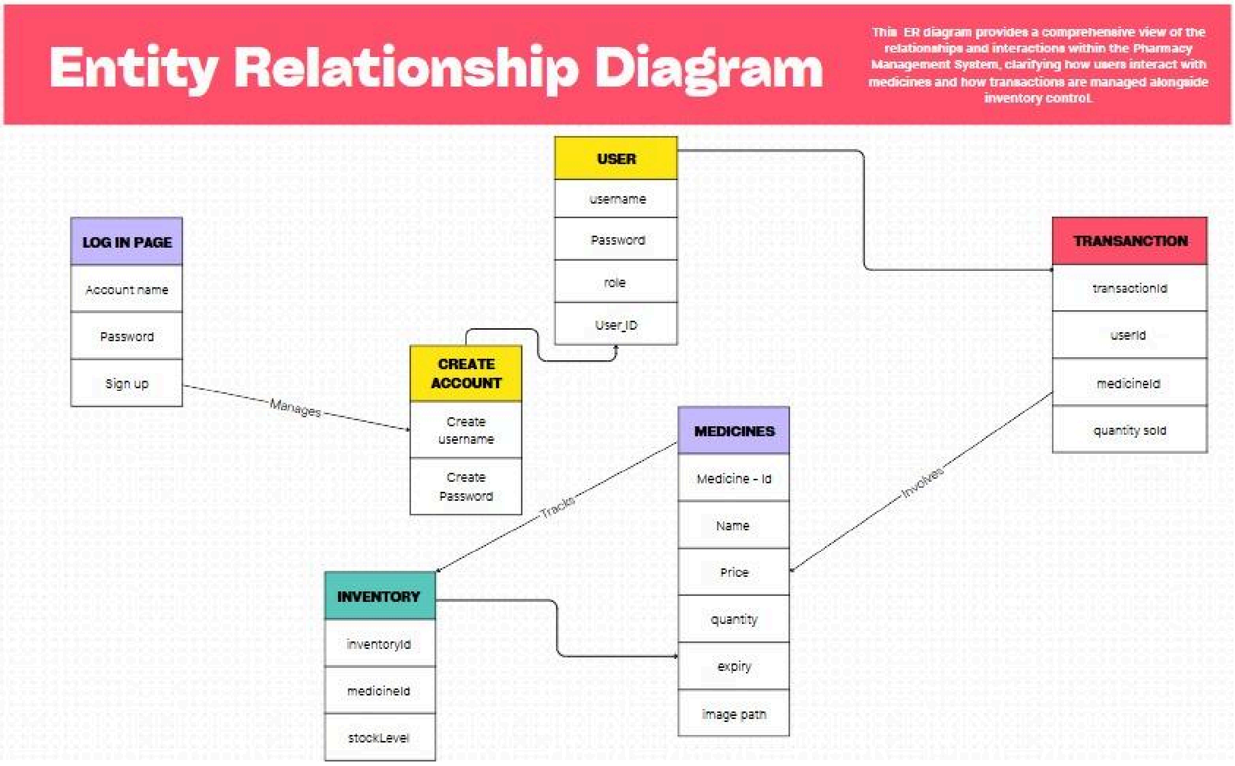
3.3 MedicinePanel Class:

```
class MedicinePanel extends JPanel { private
    Dashboard dashboard;


    public MedicinePanel(Dashboard dashboard) {
        this.dashboard = dashboard;
        setLayout(new BorderLayout());
        setBackground(new Color(240, 240, 240));


        JLabel headerLabel = new JLabel("MEDICINE
        INVENTORY", JLabel.CENTER);
        headerLabel.setFont(new Font("Arial", Font.BOLD, 24));
        headerLabel.setForeground(new Color(0x007BFF));


        add(headerLabel, BorderLayout.NORTH);


                 JPanel medicineListPanel = new JPanel(new
         GridLayout(0, 1));
        medicineListPanel.setBackground(Color.WHITE);


        loadMedicines(medicineListPanel);
        JScrollPane scrollPane = new JScrollPane(medicineListPanel);
        scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);

        add(scrollPane, BorderLayout.CENTER);

        JButton logoutButton = createLogoutButton();
        add(logoutButton, BorderLayout.SOUTH);
    }

    private void loadMedicines(JPanel medicineListPanel) {
        for (Medicine medicine : dashboard.getMedicines().values()) {
            JButton medicineButton = new JButton(medicine.getName());
            medicineButton.setBackground(new Color(220, 220, 220));
            medicineButton.addActionListener(e -> showMedicineDetails(medicine));
            medicineListPanel.add(medicineButton);
        }
    }
    private JButton createLogoutButton() {
        JButton logoutButton = new JButton("Logout");
        logoutButton.setBackground(new Color(0xDC3545));
        logoutButton.setForeground(Color.WHITE);

        logoutButton.addActionListener(e -> ((CardLayout)
getParent().getLayout()).show(getParent(), "LoginPanel"));

        return logoutButton;
    }

    private void showMedicineDetails(Medicine medicine) {
        JOptionPane.showMessageDialog(this,
                            String.format("<html><b>Medicine:</b>
                %s<br><b>Price:</b> $%.2f<br><b>Quantity:</b>
%d<br><b>Expiry:</b> %s</html>",
                        medicine.getName(), medicine.getPrice(),
medicine.getQuantity(), medicine.getExpiry()),
                "Medicine Details",
                JOptionPane.INFORMATION_MESSAGE);
    }
}
```

3.4 Medicine Class:

```
class Medicine {
    private String name;
    private double price;
    private int quantity;
    private String expiry;
    private String imagePath;

    public Medicine(String name, double price, int quantity, String expiry, String
imagePath) {
        this.name = name;
        this.price = price;
        this.quantity = quantity;
        this.expiry = expiry;
        this.imagePath = imagePath;
    }

    public String getName() { return
        name;
    }

    public double getPrice() {
        return price;
    }
  public int getQuantity() { return
        quantity;
    }
 public String getExpiry() {
        return expiry;
    }
   public String getImagePath() { return
        imagePath;
    }
}
```

DATABASE ER DIAGRAM



Entity Relationship Diagram

This ER diagram provides a comprehensive view of the relationships and interactions within the Pharmacy Management System, clarifying how users interact with medicines and how transactions are managed alongside inventory control.

DATABASE CONNECTION

```
 import java.sql.Connection;
 import java.sql.DriverManager; import
 java.sql.SQLException;


 public class DatabaseConnection {
private static final String URL =
 "jdbc:mysql://localhost:3306/pharmacy"; // Change to your DB URL
     private static final String USER = "root"; // Change to your DB user
     private static final String PASSWORD = "password"; // Change to
 your DB password


     public static Connection getConnection() throws SQLException {
         return DriverManager.getConnection(URL, USER, PASSWORD);
     }
 }


 import java.sql.*;
 import java.util.ArrayList;
 import java.util.List;



 class Medicine {
     private String name;
     private double price;
     private int quantity;
     private String expiry;
     private String imagePath;
```

```
    public Medicine(String name, double price, int quantity, String expiry) {
        this(name, price, quantity, expiry, ""); // Default empty path
    }

    public Medicine(String name, double price, int quantity, String expiry, String
imagePath) {
        this.name = name;
        this.price = price;
        this.quantity = quantity;
        this.expiry = expiry;
        this.imagePath = imagePath; // Set image path
    }

    // Getters and toString() method...
```

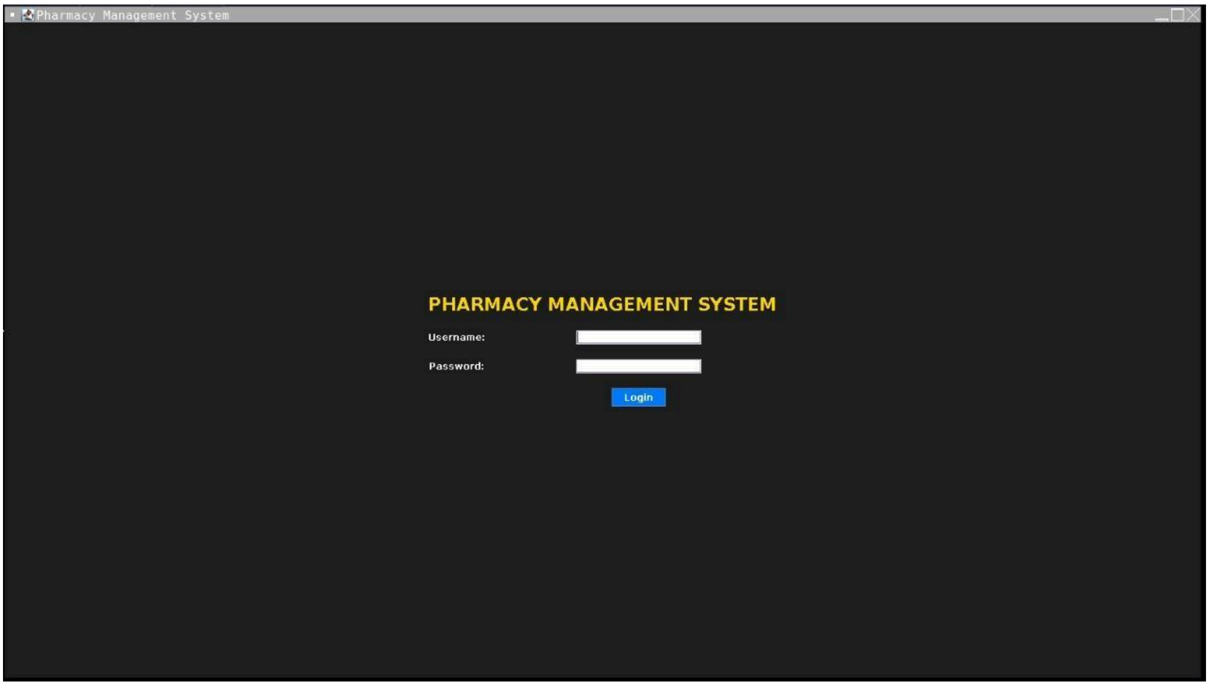
```
   public static List<Medicine> fetchMedicinesFromDatabase()
{
      List<Medicine> medicines = new ArrayList<>();
      String query = "SELECT name, price, quantity, expiry
FROM medicines"; // Adjust your query as needed


      try (Connection conn = DatabaseConnection.getConnection();
          Statement stmt = conn.createStatement(); ResultSet rs
          = stmt.executeQuery(query)) {


         while (rs.next()) {
            String name = rs.getString("name"); double
            price = rs.getDouble("price"); int quantity =
            rs.getInt("quantity"); String expiry =
            rs.getString("expiry");
            medicines.add(new Medicine(name, price, quantity,
expiry));
         }
      } catch (SQLException e) {
         e.printStackTrace(); // Handle exceptions properly in production code
      }
```

```
        return medicines;
    }
}

class Dashboard {
    private List<Medicine> medicines;

    public Dashboard(JPanel mainPanel) {
        medicines = Medicine.fetchMedicinesFromDatabase(); // Load from DB
    }

    public List<Medicine> getMedicines() { return
        medicines;
```

```
private void loadMedicines(JPanel medicineListPanel) { for (Medicine
   medicine : dashboard.getMedicines()) {
       JButton medicineButton = new JButton(medicine.getName());
       medicineButton.setBackground(new Color(220, 220,
220));
       medicineButton.addActionListener(e ->
showMedicineDetails(medicine));
       medicineListPanel.add(medicineButton);
   }
}
```

SNAPSHOTS:

Registration page

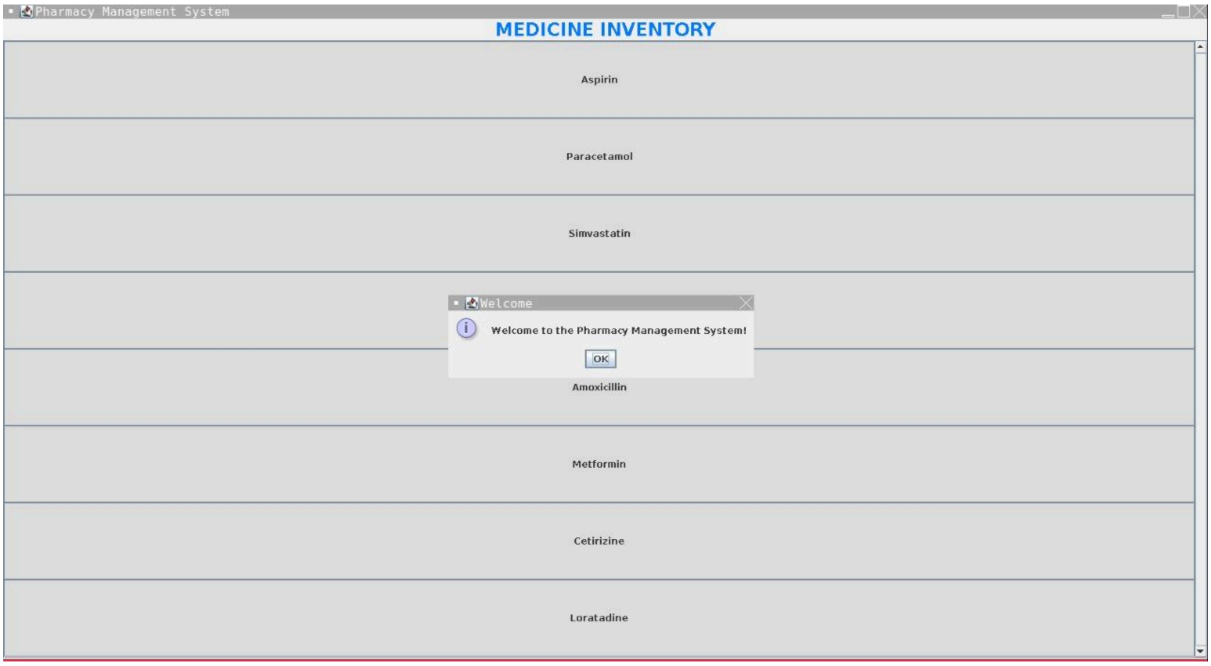


PHARMACY MANAGEMENT SYSTEM

Username:

Password:

Medicine inventory



MEDICINE INVENTORY

Aspirin

Paracetamol

Simvastatin

Welcome to the Pharmacy Management System!

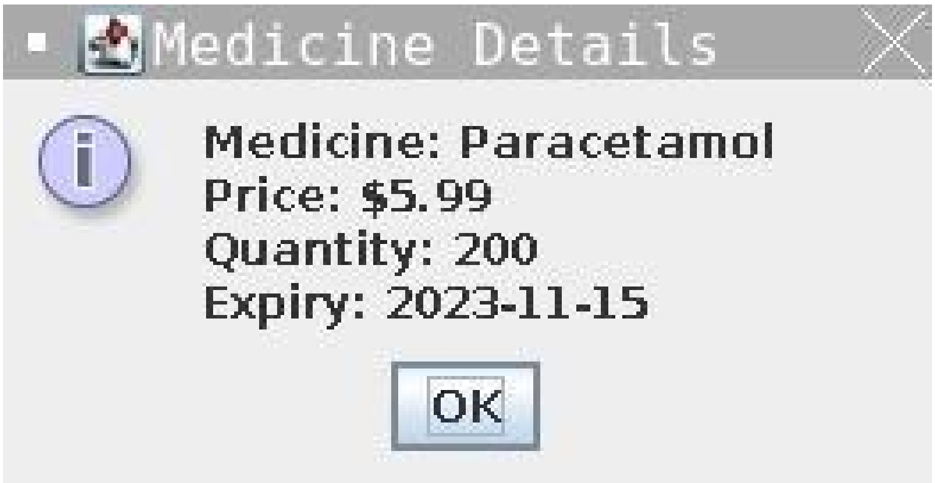
Amoxicillin

Metformin

Cetirizine

Loratadine

Medicine details



Medicine: Paracetamol
Price: $5.99
Quantity: 200
Expiry: 2023-11-15

Conclusion:

The Pharmacy Management System simplifies the management of medicine inventory, ensuring seamless tracking of stock, expiry dates, and pricing. With the help of this system, pharmacy staff can efficiently handle medicine data, restock as needed, and provide accurate information to customers. The user-friendly interface organizes critical data and makes it easily accessible through secured accounts, saving significant time and effort while enhancing overall productivity and customer satisfaction