GROCERY MANAGEMENT SYSTEM

A MINI-PROJECT BY:
R Jai Harish 231001068
A Jegaeswaran 231001075

in partial fulfillment of the award of the degree

OF

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2024

BONAFIDE CERTIFICATE

Certified that this project **"GROCERY MANAGEMENT SYSTEM"** is the bonafide work of **"R Jai Harish,A Jegaeswaran"** who carried out the project work under my supervision.

Submitted for the practical examination held on _____

SIGNATURE

SIGNATURE

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

This project develops a Grocery Management System using Java Swing to streamline product management for grocery store owners. The system provides a user-friendly interface for managing inventory details such as product name, price, discount, stock, and expiry date. It consists of three key components: a secure login for owner authentication, a dynamic product display panel, and an add product panel for database management.

The system uses Java Swing for the GUI and JDBC for database connectivity, storing product data in a MySQL database. Features include real-time inventory updates, input validation, and seamless navigation between different sections of the application. This project aims to reduce manual errors and improve operational efficiency for grocery store owners.

By applying object-oriented programming principles and database integration, the system addresses real-world inventory management challenges and provides an intuitive solution for retail operations.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 INTRODUCTION
1.2 IMPLEMENTATION
1.3 SCOPE OF THE PROJECT
1.4 WEBSITE FEATURES

2. SYSTEM SPECIFICATION

2.1 HARDWARE SPECIFICATION
2.2 SOFTWARE SPECIFICATION

3. SAMPLE CODE

4. SNAPSHOTS

4.1 LOGIN PAGE
4.2 UPDATING PRODUCTS
4.3 PRODUCTS LIST

5. CONCLUSION

6. REFERENCES

INTRODUCTION

1.1 INTRODUCTION

The Grocery Management System is a Java Swing-based application designed to simplify inventory management for grocery stores. It allows store owners to manage product details like stock, pricing, and expiry dates, with data stored in a MySQL database. The system provides a user-friendly interface to streamline daily operations and enhance efficiency.

1.2 IMPLEMENTATION

The **GROCERY MANAGEMENT SYSTEM** project discussed here is implemented using the concepts of **JAVA SWINGS** and **MYSQL**.

1.3 SCOPE OF THE PROJECT

The website is designed in a way where students will have to register on the website by creating an account for themselves in order for the students to get all their data in one place, where everything is organized for them.Thus saving them time and giving a sense of professionalism.

1.4 WEBSITE FEATURES

- Login Page
- Product Dashboard
- Add New Product
- Edit Product Details
- Database Integration

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS:

PROCESSOR	:	Intel i5
MEMORY SIZE	:	4GB(Minimum)
HARD DISK	:	500 GB of free space

2.2 SOFTWARE SPECIFICATIONS:

PROGRAMMING LANGUAGE	:	Java, MySQL
FRONT-END	:	Java
BACK-END	:	MySQL
OPERATING SYSTEM	:	Windows 10

SAMPLE CODE

PROGRAM CODE

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

public class GroceryManagementSystem extends JFrame {
    // JDBC Connection variables
    private Connection connection;
    private CardLayout cardLayout;
    private JPanel mainPanel;

    // Login Panel Components
    private JTextField usernameField;
    private JPasswordField passwordField;

    // Product Panel Components
    private JTextField nameField, mrpField, discountField, stockField, expiryField;
    private JTable productTable;

    public static void main(String[] args) {
        // Setup the frame
        GroceryManagementSystem frame = new GroceryManagementSystem();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(800, 600);
        frame.setVisible(true);
    }

    public GroceryManagementSystem() {
        try {
            // Initialize JDBC connection
            connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/grocerydb",
"root", "Sanjay1234%");

            // Setup CardLayout and main panel
            cardLayout = new CardLayout();
            mainPanel = new JPanel(cardLayout);
            add(mainPanel);

            // Add Login Panel
            mainPanel.add(createLoginPanel(), "Login");
```

```
            // Add Product Display Panel
            mainPanel.add(createProductDisplayPanel(), "ProductDisplay");

            // Add Add Product Panel
            mainPanel.add(createAddProductPanel(), "AddProduct");
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this, "Database connection failed", "Error",
JOptionPane.ERROR_MESSAGE);
        }
    }

    // Login Panel
    private JPanel createLoginPanel() {
        JPanel loginPanel = new JPanel();
        loginPanel.setLayout(new GridLayout(3, 2));

        // Username and Password fields
        loginPanel.add(new JLabel("Username:"));
        usernameField = new JTextField();
        loginPanel.add(usernameField);

        loginPanel.add(new JLabel("Password:"));
        passwordField = new JPasswordField();
        loginPanel.add(passwordField);

        // Login Button
        JButton loginButton = new JButton("Login");
        loginButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String username = usernameField.getText();
                String password = new String(passwordField.getPassword());

                // Simulate login check (you can replace this with actual authentication logic)
                if ("owner".equals(username) && "password".equals(password)) {
                    cardLayout.show(mainPanel, "ProductDisplay");
                } else {
                    JOptionPane.showMessageDialog(loginPanel, "Invalid credentials", "Error",
JOptionPane.ERROR_MESSAGE);
                }
            }
        });

        loginPanel.add(loginButton);
        return loginPanel;
    }
```

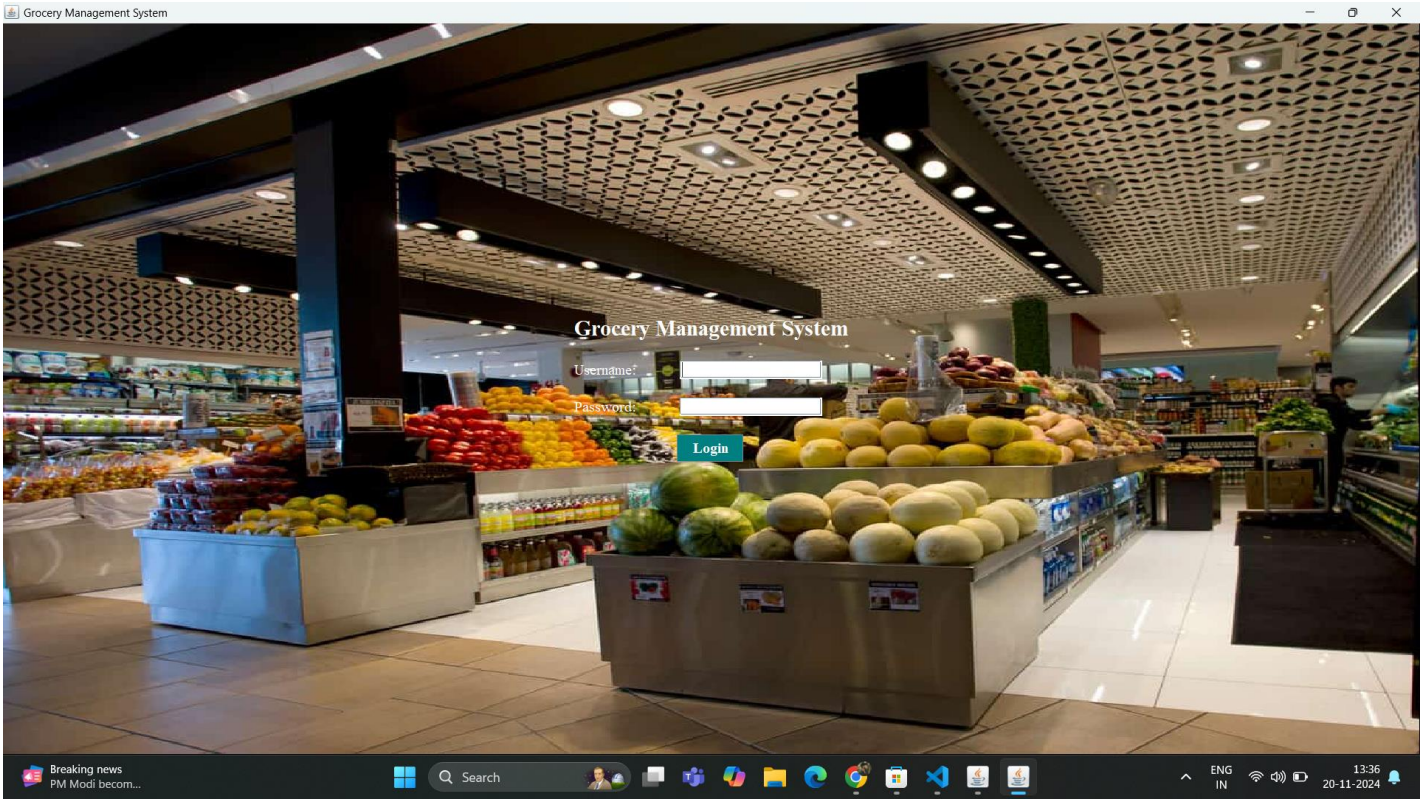
```
    // Product Display Panel
    private JPanel createProductDisplayPanel() {
        JPanel productDisplayPanel = new JPanel();
        productDisplayPanel.setLayout(new BorderLayout());

        // Product Table
        productTable = new JTable(new DefaultTableModel(new Object[]{"ID", "Product
Name", "MRP", "Discount", "Stock", "Expiry Date"}, 0));
        JScrollPane scrollPane = new JScrollPane(productTable);
        productDisplayPanel.add(scrollPane, BorderLayout.CENTER);

        // Add New Product Button
        JButton addProductButton = new JButton("Add New Product");
        addProductButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                cardLayout.show(mainPanel, "AddProduct");
            }
        });
        productDisplayPanel.add(addProductButton, BorderLayout.SOUTH);

        // Refresh table on panel load
        refreshProductTable(productTable);

        return productDisplayPanel;
    }

    // Add Product Panel
    private JPanel createAddProductPanel() {
        JPanel addProductPanel = new JPanel();
        addProductPanel.setLayout(new GridLayout(6, 2));

        // Input Fields
        addProductPanel.add(new JLabel("Product Name:"));
        nameField = new JTextField();
        addProductPanel.add(nameField);

        addProductPanel.add(new JLabel("MRP:"));
        mrpField = new JTextField();
        addProductPanel.add(mrpField);

        addProductPanel.add(new JLabel("Discount:"));
        discountField = new JTextField();
        addProductPanel.add(discountField);

        addProductPanel.add(new JLabel("Stock:"));
        stockField = new JTextField();
        addProductPanel.add(stockField);
```

```
        addProductPanel.add(new JLabel("Expiry Date (yyyy-mm-dd):"));
        expiryField = new JTextField();
        addProductPanel.add(expiryField);

        // Add Button
        JButton addButton = new JButton("Add Product");
        addButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    // Get input values
                    String name = nameField.getText();
                    double mrp = Double.parseDouble(mrpField.getText());
                    double discount = Double.parseDouble(discountField.getText());
                    int stock = Integer.parseInt(stockField.getText());
                    String expiryDate = expiryField.getText();

                    // Insert product into database
                    String insertQuery = "INSERT INTO products (name, mrp, discount, stock,
expiry_date) VALUES (?, ?, ?, ?, ?)";
                    PreparedStatement ps = connection.prepareStatement(insertQuery);
                    ps.setString(1, name);
                    ps.setDouble(2, mrp);
                    ps.setDouble(3, discount);
                    ps.setInt(4, stock);
                    ps.setString(5, expiryDate);
                    ps.executeUpdate();

                    JOptionPane.showMessageDialog(addProductPanel, "Product added
successfully!", "Success", JOptionPane.INFORMATION_MESSAGE);

                    // Refresh the product table
                    refreshProductTable(productTable);

                    // Switch back to the product display panel
                    cardLayout.show(mainPanel, "ProductDisplay");
                } catch (SQLException ex) {
                    ex.printStackTrace();
                    JOptionPane.showMessageDialog(addProductPanel, "Error adding product!",
"Error", JOptionPane.ERROR_MESSAGE);
                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(addProductPanel, "Please enter valid numeric
values!", "Error", JOptionPane.ERROR_MESSAGE);
                }
            }
        });
        addProductPanel.add(addButton);
```

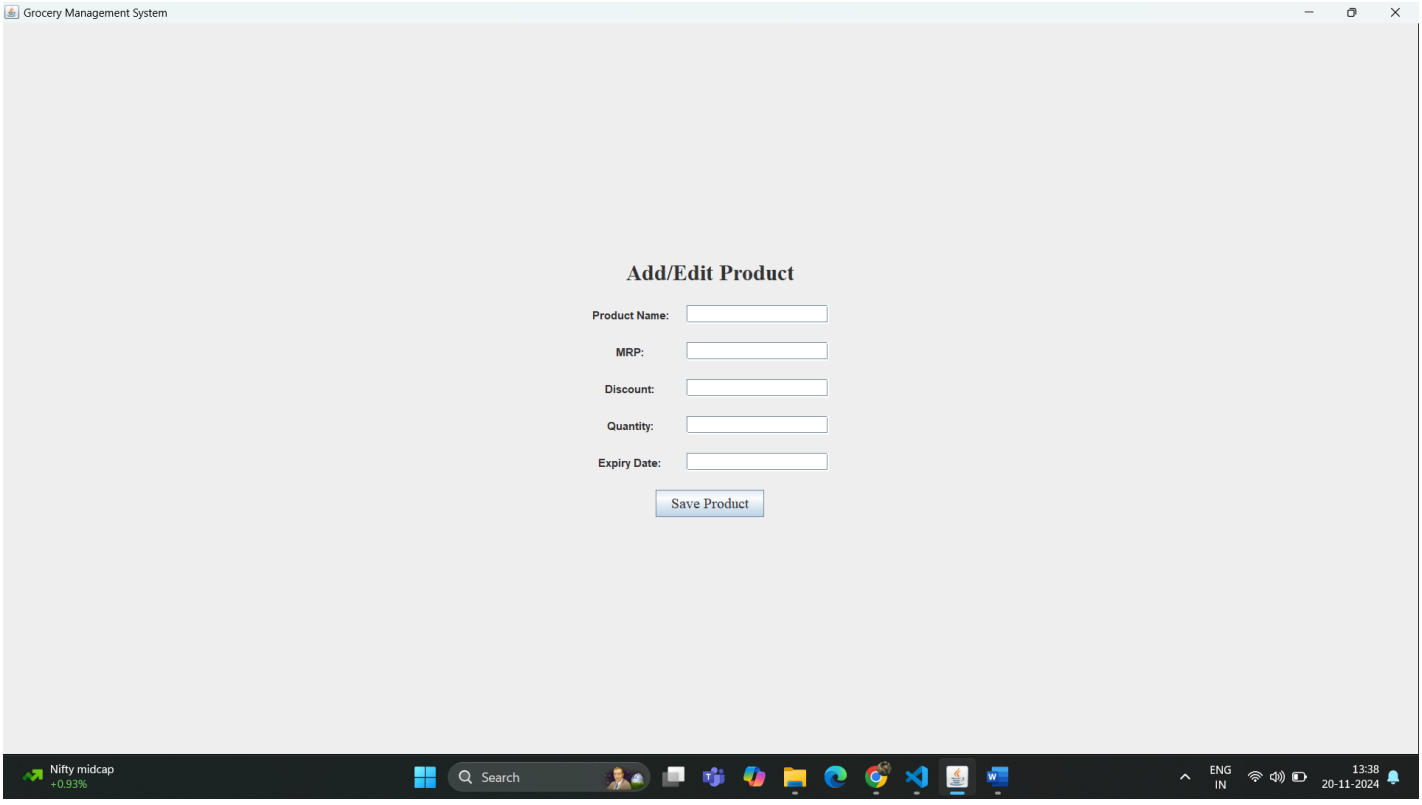
```
        return addProductPanel;
    }

    // Method to refresh product table
    private void refreshProductTable(JTable table) {
        try {
            String query = "SELECT * FROM products";
            PreparedStatement ps = connection.prepareStatement(query);
            ResultSet rs = ps.executeQuery();

            // Get table model and clear existing rows
            DefaultTableModel tableModel = (DefaultTableModel) table.getModel();
            tableModel.setRowCount(0);

            // Add rows to the table
            while (rs.next()) {
                Object[] row = {
                    rs.getInt("id"),
                    rs.getString("name"),
                    rs.getDouble("mrp"),
                    rs.getDouble("discount"),
                    rs.getInt("stock"),
                    rs.getDate("expiry_date")
                };
                tableModel.addRow(row);
            }
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error refreshing product table", "Error",
JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

SNAPSHOTS

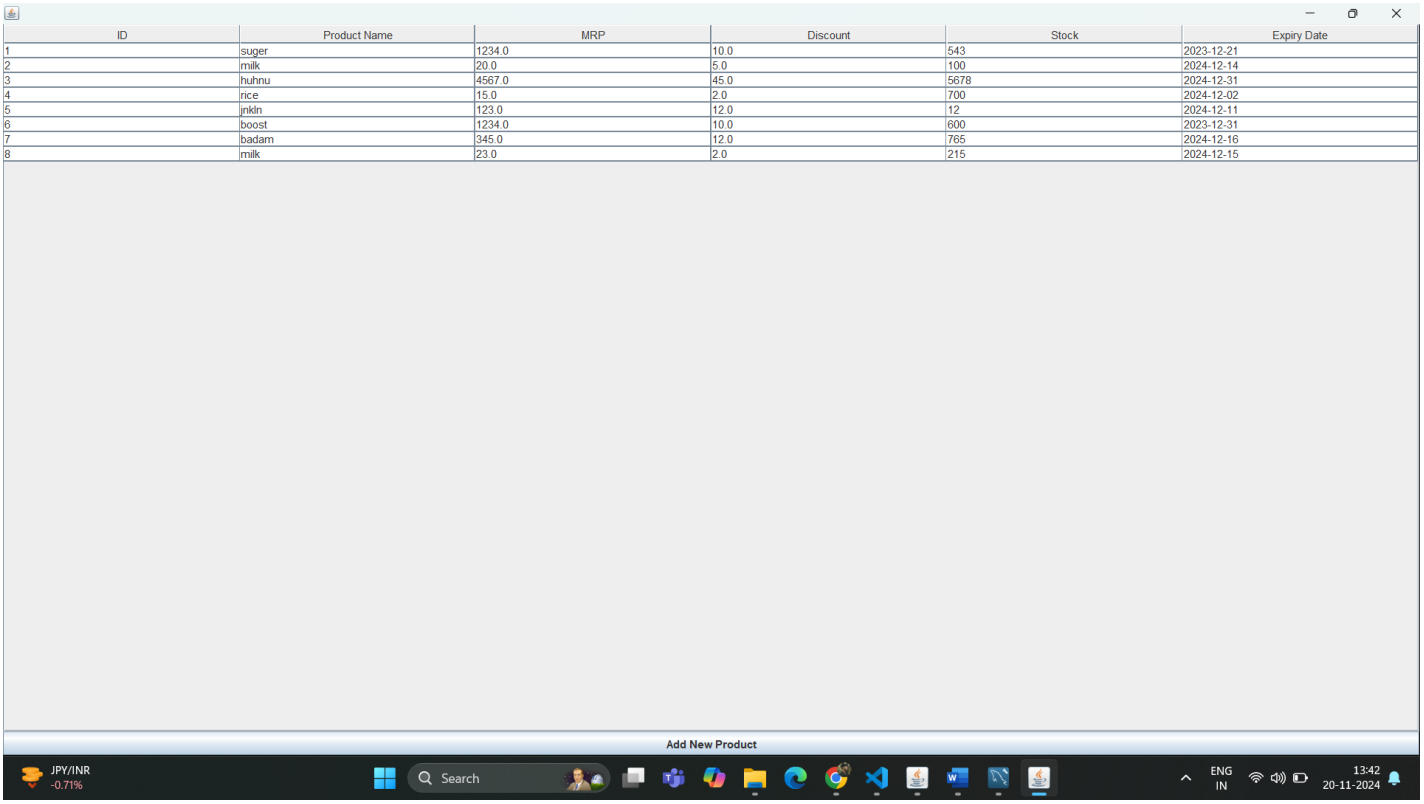
4.1 LOGIN PAGE



UPDATING PRODUCTS



4.2 PRODUCTS LIST



CONCLUSION

the design and backend implementation of the grocery management system using Java Swing and JDBC provide a solid foundation for building a user-friendly and functional application. The registration and login pages ensure secure user authentication, while the dashboard offers easy navigation for managing products and other key features. By integrating frontend components with a MySQL database, the system becomes scalable and efficient. Overall, this system serves as a starting point for a fully-fledged grocery management application that can be expanded with additional functionalities as needed.

REFERENCES

1. **https://www.javatpoint.com/java-tutorial**
2. https://www.wikipedia.org/
3. https://www.w3schools.com/sql/