SUPER MARKET MANAGEMENYT SYSTEM

CODING:

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.*;
import java.util.HashMap;
import java.util.Map;

class SupermarketBillingSystem {

    private static DefaultTableModel cartTableModel;
    private static JLabel totalLabel;
    private static Map<String, Double> productCatalog = new
HashMap<>();

    public static void main(String[] args) {
        // Populate product catalog
        populateProductCatalog();

        // Create Main Frame
        JFrame frame = new JFrame("Supermarket Billing System");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(800, 600);

        // Layout
        frame.setLayout(new BorderLayout());

        // Product Panel
        JPanel productPanel = createProductSelectionPanel();
        frame.add(productPanel, BorderLayout.NORTH);

        // Cart Panel
        JPanel cartPanel = createCartPanel();
        frame.add(cartPanel, BorderLayout.CENTER);

        // Checkout Panel
        JPanel checkoutPanel = createCheckoutPanel();
        frame.add(checkoutPanel, BorderLayout.SOUTH);
        frame.setVisible(true);
    }

    private static void populateProductCatalog() {
        productCatalog.put("Apple", 1.2);  // Price per unit
        productCatalog.put("Banana", 0.8);
        productCatalog.put("Milk", 1.5);
        productCatalog.put("Bread", 2.0);
        productCatalog.put("Eggs (Dozen)", 3.0);
    }

    private static JPanel createProductSelectionPanel() {
        JPanel panel = new JPanel(new GridLayout(3, 2));

        // Form Inputs
        JLabel productLabel = new JLabel("Product:");
        JComboBox<String> productCombo = new
JComboBox<>(productCatalog.keySet().toArray(new String[0]));
        JLabel quantityLabel = new JLabel("Quantity:");
        JTextField quantityField = new JTextField();

        panel.add(productLabel);
        panel.add(productCombo);
        panel.add(quantityLabel);
        panel.add(quantityField);

        // Add to Cart Button
        JButton addButton = new JButton("Add to Cart");
        addButton.addActionListener(e -> {
            String selectedProduct = (String)
productCombo.getSelectedItem();
            int quantity;
            try {
                quantity = Integer.parseInt(quantityField.getText());
                if (quantity <= 0) throw new NumberFormatException();
            } catch (NumberFormatException ex) {
                JOptionPane.showMessageDialog(null, "Please enter a valid
quantity.");
                return;
            }

            double price = productCatalog.get(selectedProduct);
            double total = price * quantity;
            // Add product to cart
            cartTableModel.addRow(new Object[]{selectedProduct, quantity,
price, total});
            updateTotal();

            // Clear inputs
            quantityField.setText("");
        });

        panel.add(new JLabel()); // Spacer
        panel.add(addButton);

        return panel;
    }

    private static JPanel createCartPanel() {
        JPanel panel = new JPanel(new BorderLayout());

        // Table for cart items
        String[] columnNames = {"Product", "Quantity", "Price per Unit",
"Total"};
        cartTableModel = new DefaultTableModel(columnNames, 0);
        JTable cartTable = new JTable(cartTableModel);

        panel.add(new JScrollPane(cartTable), BorderLayout.CENTER);

        return panel;
    }

    private static JPanel createCheckoutPanel() {
        JPanel panel = new JPanel(new BorderLayout());

        // Total Label
        totalLabel = new JLabel("Total: $0.00");
        totalLabel.setHorizontalAlignment(SwingConstants.CENTER);
        totalLabel.setFont(new Font("Arial", Font.BOLD, 16));
        panel.add(totalLabel, BorderLayout.NORTH);

        // Checkout Button
        JButton checkoutButton = new JButton("Checkout");
        checkoutButton.addActionListener(e -> {
            if (cartTableModel.getRowCount() == 0) {
```

```
            JOptionPane.showMessageDialog(null, "Cart is empty. Add
items before checkout.");
            return;
         }

         // Generate receipt
         StringBuilder receipt = new StringBuilder("Supermarket
Receipt\n");
         receipt.append("===================\n");
         double grandTotal = 0.0;

         for (int i = 0; i < cartTableModel.getRowCount(); i++) {
            String product = (String) cartTableModel.getValueAt(i, 0);
            int quantity = (int) cartTableModel.getValueAt(i, 1);
            double price = (double) cartTableModel.getValueAt(i, 2);
            double total = (double) cartTableModel.getValueAt(i, 3);

            receipt.append(product).append(" - Qty:
").append(quantity).append(", Total: $").append(total).append("\n");
            grandTotal += total;
         }

         receipt.append("===================\n");
         receipt.append("Grand Total:
$").append(grandTotal).append("\n");
         receipt.append("Thank you for shopping!");

         JOptionPane.showMessageDialog(null, new JScrollPane(new
JTextArea(receipt.toString())), "Receipt",
JOptionPane.INFORMATION_MESSAGE);

         // Clear cart
         cartTableModel.setRowCount(0);
         updateTotal();
      });

      panel.add(checkoutButton, BorderLayout.CENTER);

      return panel;
   }

   private static void updateTotal() {
      double total = 0.0;
```

```
        for (int i = 0; i < cartTableModel.getRowCount(); i++) {
            total += (double) cartTableModel.getValueAt(i, 3);
        }
        totalLabel.setText("Total: $" + String.format("%.2f", total));
    }
}
```