```java
import java.awt.*;

import java.awt.event.*;

import javax.swing.*;

import javax.swing.border.*;

import java.util.*;

import java.util.List; // Correct import


// Product class

class Product {

   private int id;

   private String name;

   private double price;

   private int stock;


   public Product(int id, String name, double price, int stock) {

      this.id = id;

      this.name = name;

      this.price = price;

      this.stock = stock;

   }


   public int getId() {

      return id;

   }


   public String getName() {
      return name;
   }

   public double getPrice() {
      return price;
   }

   public int getStock() {
      return stock;
   }

   public void setStock(int stock) {
      this.stock = stock;
   }

   @Override
   public String toString() {
      return id + ". " + name + " - $" + price + " (Stock: " + stock + ")";
   }
}

// Order class
class Order {
   private int orderId;
   private List<Product> products;
   private double total;
   public Order(int orderId, List<Product> products, double total) {
      this.orderId = orderId;
      this.products = products;
      this.total = total;
   }

   @Override
   public String toString() {
      StringBuilder orderDetails = new StringBuilder("Order ID: " + orderId + "\n");
      for (Product product : products) {
          orderDetails.append("- ").append(product.getName()).append("
($").append(product.getPrice()).append(")\n");
      }
      orderDetails.append("Total: $").append(total);
      return orderDetails.toString();
   }
}

// Main Frame Class
public class Main extends JFrame {

   private static final long serialVersionUID = 1L;
   private JPanel contentPane;
   private ECommerceSystem system;
```

```
public static void main(String[] args) {
    EventQueue.invokeLater(() -> {
        try {
            Main frame = new Main();
            frame.setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    });
}

public Main() {
    system = new ECommerceSystem();
    system.addAdmin("admin", "admin123"); // Sample admin credentials
    system.addProduct(new Product(1, "Laptop", 999.99, 5));
    system.addProduct(new Product(2, "Smartphone", 499.99, 10));
    system.addProduct(new Product(3, "Headphones", 79.99, 15));

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 600, 400);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    contentPane.setLayout(null);
    setContentPane(contentPane);

    JLabel lblTitle = new JLabel("E-Commerce System");
```

```
        lblTitle.setBounds(200, 10, 200, 20);
        contentPane.add(lblTitle);

        JButton btnUserLogin = new JButton("User Login/Register");
        btnUserLogin.setBounds(200, 50, 200, 30);
        btnUserLogin.addActionListener(e -> system.userLogin(this));
        contentPane.add(btnUserLogin);

        JButton btnAdminLogin = new JButton("Admin Login");
        btnAdminLogin.setBounds(200, 100, 200, 30);
        btnAdminLogin.addActionListener(e -> system.adminLogin(this));
        contentPane.add(btnAdminLogin);

        JButton btnExit = new JButton("Exit");
        btnExit.setBounds(200, 150, 200, 30);
        btnExit.addActionListener(e -> System.exit(0));
        contentPane.add(btnExit);
    }
}

// ECommerceSystem class
class ECommerceSystem {
    private List<Product> productList = new ArrayList<>();
    private List<Order> orderList = new ArrayList<>();
    private List<User> userList = new ArrayList<>();
    private Map<String, String> adminCredentials = new HashMap<>();
    private int orderCounter = 1;


    public void addAdmin(String username, String password) {
        adminCredentials.put(username, password);
    }


    public void addProduct(Product product) {
        productList.add(product);
    }


    public void displayProducts(JFrame parent) {
        if (productList.isEmpty()) {
            JOptionPane.showMessageDialog(parent, "No products available.");
        } else {
            StringBuilder products = new StringBuilder("Available Products:\n");
            for (Product product : productList) {
                products.append(product).append("\n");
            }
            JOptionPane.showMessageDialog(parent, products.toString());
        }
    }


    public void placeOrder(JFrame parent) {
        String input = JOptionPane.showInputDialog(parent, "Enter product IDs to add to cart
(comma-separated): ");
        if (input == null || input.isEmpty()) {
```

```
      JOptionPane.showMessageDialog(parent, "No products selected.");
      return;
    }

    String[] ids = input.split(",");
    List<Product> cart = new ArrayList<>();
    double total = 0;

    for (String idStr : ids) {
      try {
        int productId = Integer.parseInt(idStr.trim());
        Optional<Product> productOptional = productList.stream().filter(p -> p.getId() == productId).findFirst();

        if (productOptional.isPresent()) {
          Product product = productOptional.get();
          if (product.getStock() > 0) {
            cart.add(product);
            total += product.getPrice();
            product.setStock(product.getStock() - 1);
          } else {
            JOptionPane.showMessageDialog(parent, "Product out of stock: " + product.getName());
          }
        } else {
          JOptionPane.showMessageDialog(parent, "Invalid product ID: " + productId);
        }
        } catch (NumberFormatException e) {
          JOptionPane.showMessageDialog(parent, "Invalid input: " + idStr);
        }
      }

      if (!cart.isEmpty()) {
        String[] paymentOptions = {"Credit Card", "PayPal", "Debit Card", "Cash on Delivery"};
        int paymentChoice = JOptionPane.showOptionDialog(parent, "Choose a payment
option:", "Payment",
            JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, null,
paymentOptions, paymentOptions[0]);

        if (paymentChoice != -1) {
          Order order = new Order(orderCounter++, cart, total);
          orderList.add(order);
          JOptionPane.showMessageDialog(parent, "Order placed successfully using " +
paymentOptions[paymentChoice] + "!\n" + order);
        } else {
          JOptionPane.showMessageDialog(parent, "Payment failed. Order not placed.");
        }
      } else {
        JOptionPane.showMessageDialog(parent, "No valid products selected. Order not
placed.");
      }
    }

    public void displayOrders(JFrame parent) {
        if (orderList.isEmpty()) {
            JOptionPane.showMessageDialog(parent, "No orders have been placed yet.");
        } else {
            StringBuilder orders = new StringBuilder("Order History:\n");
            for (Order order : orderList) {
                orders.append(order).append("\n-------------------\n");
            }
            JOptionPane.showMessageDialog(parent, orders.toString());
        }
    }

    public void registerUser(JFrame parent) {
        String username = JOptionPane.showInputDialog(parent, "Enter username:");
        String password = JOptionPane.showInputDialog(parent, "Enter password:");
        if (username != null && password != null && !username.isEmpty() &&
!password.isEmpty()) {
            userList.add(new User(username, password));
            JOptionPane.showMessageDialog(parent, "Registration successful. You can now log
in.");
        } else {
            JOptionPane.showMessageDialog(parent, "Invalid input. Registration failed.");
        }
    }

    public void userLogin(JFrame parent) {
        String[] options = {"Register", "Login"};
        int choice = JOptionPane.showOptionDialog(parent, "Choose an option:", "User
Authentication",
                JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, null,
options, options[0]);


        if (choice == 0) { // Register
            registerUser(parent);
        } else if (choice == 1) { // Login
            String username = JOptionPane.showInputDialog(parent, "Enter username:");
            String password = JOptionPane.showInputDialog(parent, "Enter password:");
            Optional<User> user = userList.stream().filter(u ->
u.getUsername().equals(username) && u.getPassword().equals(password)).findFirst();
            if (user.isPresent()) {
                JOptionPane.showMessageDialog(parent, "Login successful!");
                userMenu(parent);
            } else {
                JOptionPane.showMessageDialog(parent, "Invalid credentials. Login failed.");
            }
        }
    }


    public void userMenu(JFrame parent) {
        JFrame userMenuFrame = new JFrame("User Menu");
        userMenuFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        userMenuFrame.setBounds(100, 100, 600, 400);
        JPanel userMenuPanel = new JPanel();
        userMenuPanel.setLayout(new GridLayout(4, 1));
        userMenuFrame.setContentPane(userMenuPanel);

        JButton btnViewProducts = new JButton("View Products");
        btnViewProducts.addActionListener(e -> displayProducts(userMenuFrame));
        userMenuPanel.add(btnViewProducts);

        JButton btnPlaceOrder = new JButton("Place Order");
        btnPlaceOrder.addActionListener(e -> placeOrder(userMenuFrame));
        userMenuPanel.add(btnPlaceOrder);

        JButton btnViewOrders = new JButton("View Orders");
        btnViewOrders.addActionListener(e -> displayOrders(userMenuFrame));
        userMenuPanel.add(btnViewOrders);

        JButton btnLogout = new JButton("Logout");
        btnLogout.addActionListener(e -> userMenuFrame.dispose());
        userMenuPanel.add(btnLogout);

        userMenuFrame.setVisible(true);
    }

    public void adminLogin(JFrame parent) {
        String username = JOptionPane.showInputDialog(parent, "Enter admin username:");
        String password = JOptionPane.showInputDialog(parent, "Enter admin password:");
        if (adminCredentials.containsKey(username) &&
adminCredentials.get(username).equals(password)) {
          JOptionPane.showMessageDialog(parent, "Admin login successful.");
          adminMenu(parent);
        } else {
          JOptionPane.showMessageDialog(parent, "Invalid admin credentials.");
        }
    }

    public void adminMenu(JFrame parent) {
        JFrame adminMenuFrame = new JFrame("Admin Menu");
        adminMenuFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        adminMenuFrame.setBounds(100, 100, 600, 400);
        JPanel adminMenuPanel = new JPanel();
        adminMenuPanel.setLayout(new GridLayout(4, 1));
        adminMenuFrame.setContentPane(adminMenuPanel);

        JButton btnAddProduct = new JButton("Add Product");
        btnAddProduct.addActionListener(e -> {
          String name = JOptionPane.showInputDialog(adminMenuFrame, "Enter product
name:");
          String priceStr = JOptionPane.showInputDialog(adminMenuFrame, "Enter product
price:");
          String stockStr = JOptionPane.showInputDialog(adminMenuFrame, "Enter product
stock:");
          try {
            double price = Double.parseDouble(priceStr);
            int stock = Integer.parseInt(stockStr);
            addProduct(new Product(productList.size() + 1, name, price, stock));
```

```
                JOptionPane.showMessageDialog(adminMenuFrame, "Product added
successfully.");
            } catch (NumberFormatException ex) {
                JOptionPane.showMessageDialog(adminMenuFrame, "Invalid input. Product not
added.");
            }
        });
        adminMenuPanel.add(btnAddProduct);

        JButton btnViewProducts = new JButton("View Products");
        btnViewProducts.addActionListener(e -> displayProducts(adminMenuFrame));
        adminMenuPanel.add(btnViewProducts);

        JButton btnViewOrders = new JButton("View Orders");
        btnViewOrders.addActionListener(e -> displayOrders(adminMenuFrame));
        adminMenuPanel.add(btnViewOrders);

        JButton btnLogout = new JButton("Logout");
        btnLogout.addActionListener(e -> adminMenuFrame.dispose());
        adminMenuPanel.add(btnLogout);

        adminMenuFrame.setVisible(true);
    }
}

// User class
class User {
    private String username;
    private String password;

    public User(String username, String password) {
        this.username = username;
        this.password = password;
    }

    public String getUsername() {
        return username;
    }

    public String getPassword() {
        return password;
    }
}
```