# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

Assignment No: 2

1) **Java Implementation of Ecommerce Java Application with GUI + Database.**

CODE:

package mybankproject1;  
  
import javax.swing.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.\*;  
import java.util.ArrayList;  
import java.util.List;  
  
public class MainScreen {  
 public static void main(String[] args) {  
 SwingUtilities.*invokeLater*(() -> new LoginScreen());  
 }  
}  
class LoginScreen extends JFrame {  
 private JTextField usernameField;  
 private JPasswordField passwordField;  
 private JButton loginButton;  
  
 public LoginScreen() {  
 setTitle("Login");  
 setSize(300, 200);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setLayout(null);  
  
 JLabel usernameLabel = new JLabel("Username:");  
 usernameLabel.setBounds(10, 10, 150, 25);  
 add(usernameLabel);  
  
 usernameField = new JTextField();  
 usernameField.setBounds(150, 10, 120, 25);  
 add(usernameField);  
  
 JLabel passwordLabel = new JLabel("Password:");  
 passwordLabel.setBounds(10, 50, 150, 25);  
 add(passwordLabel);  
  
 passwordField = new JPasswordField();  
 passwordField.setBounds(150, 50, 120, 25);  
 add(passwordField);  
  
 loginButton = new JButton("Login");  
 loginButton.setBounds(10, 90, 260, 25);  
 add(loginButton);  
  
 loginButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 login();  
 }  
 });  
  
 setVisible(true);  
 }  
  
 private void login() {  
 String username = usernameField.getText();  
 String password = new String(passwordField.getPassword());  
  
 if (validateUser(username, password)) {  
 JOptionPane.*showMessageDialog*(this, "Login Successful!");  
 this.dispose();

new HomeScreen();} else {  
 JOptionPane.*showMessageDialog*(this, "Invalid username or password.");  
 }  
 }  
  
 private boolean validateUser(String username, String password) {  
 String sql = "SELECT \* FROM userEcomm WHERE username = ? AND password = ?";  
 try (Connection conn = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/dbms",  
 "root", "");  
 PreparedStatement pstmt = conn.prepareStatement(sql)) {  
 pstmt.setString(1, username);  
 pstmt.setString(2, password);  
 ResultSet rs = pstmt.executeQuery();  
 return rs.next(); *// If user exists, return true* } catch (SQLException e) {  
 e.printStackTrace();  
 return false;  
 }  
 }  
}  
class HomeScreen extends JFrame {  
 private static final String *URL* = "jdbc:mysql://localhost:3306/dbms";  
 private static final String *USER* = "root";  
 private static final String *PASSWORD* = "";  
  
 private JTextField customerNameField;  
 private JRadioButton[] productButtons;  
 private JTextField quantityField;  
 private JButton orderButton;  
 private JTextArea cartArea;  
 private List<PizzaOrder> cart;  
  
 public HomeScreen() {  
 setTitle("E-Commerce Application");  
 setSize(400, 500);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setLayout(null);  
  
 cart = new ArrayList<>();  
  
 JLabel customerNameLabel = new JLabel("Customer Name:");  
 customerNameLabel.setBounds(10, 10, 150, 25);  
 add(customerNameLabel);  
  
 customerNameField = new JTextField();  
 customerNameField.setBounds(150, 10, 200, 25);  
 add(customerNameField);  
  
 JLabel productLabel = new JLabel("Select Product:");  
 productLabel.setBounds(10, 50, 150, 25);  
 add(productLabel);  
  
 String[] products = {  
 "Smartphone",  
 "Laptop",  
 "Tablet",  
 "Smartwatch",  
 "Wireless Headphones",  
 "Bluetooth Speaker",  
 "Gaming Console",  
 "Camera",  
 "Smart TV",  
 "Portable Charger"  
 };  
  
 productButtons = new JRadioButton[products.length];  
 ButtonGroup productGroup = new ButtonGroup();  
  
 for (int i = 0; i < products.length; i++) {  
 productButtons[i] = new JRadioButton(products[i]);  
 productButtons[i].setBounds(150, 50 + (i \* 30), 200, 25);  
 productGroup.add(productButtons[i]);  
 add(productButtons[i]);  
 }  
  
 JLabel quantityLabel = new JLabel("Quantity:");  
 quantityLabel.setBounds(10, 320, 150, 25);  
 add(quantityLabel);  
  
 quantityField = new JTextField();  
 quantityField.setBounds(150, 320, 200, 25);  
 add(quantityField);  
  
 orderButton = new JButton("Add to Cart");  
 orderButton.setBounds(10, 360, 340, 25);  
 add(orderButton);  
  
 cartArea = new JTextArea();  
 cartArea.setBounds(10, 400, 360, 150);  
 cartArea.setEditable(false);  
 add(cartArea);  
  
 orderButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 addToCart();  
 }  
 });  
  
 setVisible(true);  
 }  
  
 private void addToCart() {  
 String customerName = customerNameField.getText();  
 double totalPrice = 0;  
  
 for (JRadioButton productButton : productButtons) {  
 if (productButton.isSelected()) {  
 String product = productButton.getText();  
 int quantity = Integer.*parseInt*(quantityField.getText().isEmpty() ? "1" : quantityField.getText()); *// Get quantity from input field, default to 1* double price = getPrice(product);  
 totalPrice += quantity \* price;  
  
 PizzaOrder order = new PizzaOrder(customerName, product, quantity, totalPrice);  
 cart.add(order);  
 insertOrderIntoDB(order);  
 }  
 }  
  
 updateCartDisplay();  
 }  
  
 private void insertOrderIntoDB(PizzaOrder order) {  
 String sql = "INSERT INTO orders (user\_id, product\_id, quantity, total\_price) VALUES (?, " +  
 "?, ?, ?)";  
 try (Connection conn = DriverManager.*getConnection*(*URL*, *USER*, *PASSWORD*);  
 PreparedStatement pstmt = conn.prepareStatement(sql)) {  
 pstmt.setString(1, order.getCustomerName());  
 pstmt.setString(2, order.getProduct());  
 pstmt.setInt(3, order.getQuantity());  
 pstmt.setDouble(4, order.getTotalPrice());  
  
 pstmt.executeUpdate();  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
  
  
 private double getPrice(String product) {  
 switch (product) {

case "Smartphone":  
 return 699.99;  
 case "Laptop":  
 return 999.99;  
 case "Tablet":  
 return 299.99;  
 case "Smartwatch":  
 return 199.99;  
 case "Wireless Headphones":  
 return 149.99;  
 case "Bluetooth Speaker":  
 return 89.99;  
 case "Gaming Console":  
 return 399.99;  
 case "Camera":  
 return 599.99;  
 case "Smart TV":  
 return 799.99;  
 case "Portable Charger":  
 return 29.99;  
 default:  
 return 0.0;

}  
 }  
  
 private void updateCartDisplay() {  
 StringBuilder cartContent = new StringBuilder("Shopping Cart:\n");  
 for (PizzaOrder order : cart) {  
 cartContent.append(order.getCustomerName())  
 .append(": ")  
 .append(order.getQuantity())  
 .append(" x ")  
 .append(order.getProduct())  
 .append(" = $")  
 .append(order.getTotalPrice())  
 .append("\n");  
 }  
 cartArea.setText(cartContent.toString());  
 }  
  
 class PizzaOrder {  
 private String customerName;  
 private String product;  
 private int quantity;  
 private double totalPrice;  
  
 public PizzaOrder(String customerName, String product, int quantity, double totalPrice) {  
 this.customerName = customerName;  
 this.product = product;  
 this.quantity = quantity;  
 this.totalPrice = totalPrice;  
 }  
  
 public String getCustomerName() {  
 return customerName;  
 }  
  
 public String getProduct() {  
 return product;  
 }  
  
 public int getQuantity() {  
 return quantity;  
 }  
  
 public double getTotalPrice() {  
 return totalPrice;  
 }  
 }  
}

OUTPUT:

 

 

