# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

Assignment No: 1

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

CODE:

package mybankproject1;  
  
import javax.swing.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.SQLException;  
  
public class MainScreen 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 JComboBox<String> pizzaTypeCombo;  
 private JTextField quantityField;  
 private JButton orderButton;  
  
 public MainScreen() {  
 setTitle("Pizza Order Application");  
 setSize(300, 300);  
 setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 setLayout(null);  
  
 JLabel customerNameLabel = new JLabel("Customer Name:");  
 customerNameLabel.setBounds(10, 10, 150, 25);  
 add(customerNameLabel);  
  
 customerNameField = new JTextField();  
 customerNameField.setBounds(150, 10, 120, 25);  
 add(customerNameField);  
  
 JLabel pizzaTypeLabel = new JLabel("Pizza Type:");  
 pizzaTypeLabel.setBounds(10, 50, 150, 25);  
 add(pizzaTypeLabel);  
  
 pizzaTypeCombo = new JComboBox<>(new String[]{"Margherita", "Pepperoni", "Veggie"});  
 pizzaTypeCombo.setBounds(150, 50, 120, 25);  
 add(pizzaTypeCombo);  
  
 JLabel quantityLabel = new JLabel("Quantity:");  
 quantityLabel.setBounds(10, 90, 150, 25);  
 add(quantityLabel);  
  
 quantityField = new JTextField();  
 quantityField.setBounds(150, 90, 120, 25);  
 add(quantityField);  
  
 orderButton = new JButton("Place Order");  
 orderButton.setBounds(10, 130, 260, 25);  
 add(orderButton);  
  
 orderButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 placeOrder();  
 }  
 });  
 }  
  
 private void placeOrder() {  
 String customerName = customerNameField.getText();  
 String pizzaType = (String) pizzaTypeCombo.getSelectedItem();  
 int quantity;  
  
 try {  
 quantity = Integer.parseInt(quantityField.getText());  
 double totalPrice = quantity \* 10.0;   
  
 PizzaOrder order = new PizzaOrder(customerName, pizzaType, quantity, totalPrice);  
 saveOrderToDatabase(order);  
 } catch (NumberFormatException e) {  
 JOptionPane.showMessageDialog(this, "Please enter a valid quantity.");  
 }  
 }  
  
 private void saveOrderToDatabase(PizzaOrder order) {  
 String sql = "INSERT INTO orders (customer\_name, pizza\_type, 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.getPizzaType());  
 pstmt.setInt(3, order.getQuantity());  
 pstmt.setDouble(4, order.getTotalPrice());  
 pstmt.executeUpdate();  
 JOptionPane.showMessageDialog(this, "Order placed successfully!");  
 } catch (SQLException e) {  
 e.printStackTrace();  
 JOptionPane.showMessageDialog(this, "Failed to place order.");  
 }  
 }  
  
 public static void main(String[] args) {  
 SwingUtilities.invokeLater(() -> {  
 MainScreen gui = new MainScreen();  
 gui.setVisible(true);  
 });  
 }  
  
   
 class PizzaOrder {  
 private String customerName;  
 private String pizzaType;  
 private int quantity;  
 private double totalPrice;  
  
 public PizzaOrder(String customerName, String pizzaType, int quantity, double totalPrice) {  
 this.customerName = customerName;  
 this.pizzaType = pizzaType;  
 this.quantity = quantity;  
 this.totalPrice = totalPrice;  
 }  
  
  
 public String getCustomerName() {  
 return customerName;  
 }  
  
 public String getPizzaType() {  
 return pizzaType;  
 }  
  
 public int getQuantity() {  
 return quantity;  
 }  
  
 public double getTotalPrice() {  
 return totalPrice;  
 }  
 }  
}

OUTPUT:

 

