AIM:

TO PRESENT A FASTFOOD BILL PROJECTOR USING JAVA AND DBMS

SOURCE CODE:

```
 package db;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

class DatabaseConnection {

  // Database URL, username, and password
  static final String DB_URL = "jdbc:mysql://localhost:3306/mydatabase"; // Replace 'mydatabase' with
your DB name
  static final String USER = "root"; // Replace with your username
  static final String PASS = "Madhan@123"; // Replace with your password

  public static void main(String[] args) {
    Connection connection = null;
    Statement statement = null;

    try {
      // 1. Establish a connection
      connection = DriverManager.getConnection(DB_URL, USER, PASS);
      System.out.println("Connected to the database successfully.");

      // 2. Create a statement
      statement = connection.createStatement();
      String sql = "SELECT id, name, email FROM users"; // Replace 'users' with your table name

      // 3. Execute the query
      ResultSet resultSet = statement.executeQuery(sql);

      // 4. Process the result set
      while (resultSet.next()) {
        // Retrieve data by column name
        int id = resultSet.getInt("id");
        String name = resultSet.getString("name");
        String email = resultSet.getString("email");

        // Display values
        System.out.println("ID: " + id + ", Name: " + name + ", Email: " + email);
      }
```

```
            // Close the result set
            resultSet.close();

        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            // 5. Clean up the environment
            try {
                if (statement != null) statement.close();
                if (connection != null) connection.close();
            } catch (SQLException se) {
                se.printStackTrace();
            }
        }
    }
}
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;
import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.Locale;

// Model for Menu Item
class MenuItem {
    private String name;
    private double price;
    private String category;

    public MenuItem(String name, double price, String category) {
        this.name = name;
        this.price = price;
        this.category = category;
    }

    public String getName() { return name; }
    public double getPrice() { return price; }

    @Override
    public String toString() {
        return name + " - " + formatCurrency(price);
    }

    private String formatCurrency(double amount) {
        NumberFormat currencyFormat = NumberFormat.getCurrencyInstance(new Locale("en", "IN"));
```

```
        return currencyFormat.format(amount);
    }
}

// Model for Order
class Order {
    private ArrayList<MenuItem> items = new ArrayList<>();
    private static final double TAX_RATE = 0.18; // 18% GST in India

    public void addItem(MenuItem item) {
        items.add(item);
    }

    public void removeItem(MenuItem item) {
        items.remove(item);
    }

    public double calculateSubTotal() {
        return items.stream().mapToDouble(MenuItem::getPrice).sum();
    }

    public double calculateTax() {
        return calculateSubTotal() * TAX_RATE;
    }

    public double calculateTotal() {
        return calculateSubTotal() + calculateTax();
    }

    public ArrayList<MenuItem> getItems() {
        return items;
    }
}

// Main Application
class FastFoodPOS extends JFrame {
    private DefaultListModel<MenuItem> menuListModel;
    private DefaultListModel<MenuItem> orderListModel;
    private JList<MenuItem> menuList;
    private JList<MenuItem> orderList;
    private JLabel subTotalLabel;
    private JLabel taxLabel;
    private JLabel totalLabel;

    public FastFoodPOS() {
        // Set up the frame
        setTitle("Fast Food POS System - INR");
        setSize(800, 600);
```

```
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        // Set custom font and colors
        Font mainFont = new Font("Arial", Font.PLAIN, 14);
        Color bgColor = new Color(245, 245, 245);
        Color panelBgColor = new Color(225, 225, 225);
        Color buttonColor = new Color(60, 179, 113); // Sea green for buttons

        getContentPane().setBackground(bgColor); // Set frame background color

        // Initialize models
        menuListModel = new DefaultListModel<>();
        orderListModel = new DefaultListModel<>();
        Order order = new Order();

        // Populate menu with sample items
        menuListModel.addElement(new MenuItem("Burger", 150.0, "Main"));
        menuListModel.addElement(new MenuItem("Fries", 80.0, "Side"));
        menuListModel.addElement(new MenuItem("Coke", 50.0, "Drink"));
        // Populate menu with sample items
        menuListModel.addElement(new MenuItem("Burger", 150.0, "Main"));
        menuListModel.addElement(new MenuItem("Fries", 80.0, "Side"));
        menuListModel.addElement(new MenuItem("Coke", 50.0, "Drink"));

    // Additional 40 items added below
        menuListModel.addElement(new MenuItem("Grilled Sandwich", 120.0, "Main"));
        menuListModel.addElement(new MenuItem("Veggie Wrap", 110.0, "Main"));
        menuListModel.addElement(new MenuItem("Paneer Roll", 140.0, "Main"));
        menuListModel.addElement(new MenuItem("Chicken Roll", 160.0, "Main"));
        menuListModel.addElement(new MenuItem("Pasta Alfredo", 200.0, "Main"));
        menuListModel.addElement(new MenuItem("Pasta Arrabbiata", 210.0, "Main"));
        menuListModel.addElement(new MenuItem("BBQ Chicken Wings", 180.0, "Side"));
        menuListModel.addElement(new MenuItem("Garlic Naan", 40.0, "Side"));
        menuListModel.addElement(new MenuItem("Butter Naan", 50.0, "Side"));
        menuListModel.addElement(new MenuItem("Tandoori Roti", 30.0, "Side"));
        menuListModel.addElement(new MenuItem("Chicken Biryani", 250.0, "Main"));
        menuListModel.addElement(new MenuItem("Veg Biryani", 220.0, "Main"));
        menuListModel.addElement(new MenuItem("Paneer Biryani", 240.0, "Main"));
        menuListModel.addElement(new MenuItem("Butter Paneer", 260.0, "Main"));
        menuListModel.addElement(new MenuItem("Paneer Butter Masala", 280.0, "Main"));
        menuListModel.addElement(new MenuItem("Masala Dosa", 90.0, "Main"));
        menuListModel.addElement(new MenuItem("Plain Dosa", 70.0, "Main"));
        menuListModel.addElement(new MenuItem("Uttapam", 100.0, "Main"));
        menuListModel.addElement(new MenuItem("Idli Sambar", 50.0, "Main"));
        menuListModel.addElement(new MenuItem("Mango Lassi", 60.0, "Drink"));
        menuListModel.addElement(new MenuItem("Masala Chai", 20.0, "Drink"));
        menuListModel.addElement(new MenuItem("Espresso", 80.0, "Drink"));
```

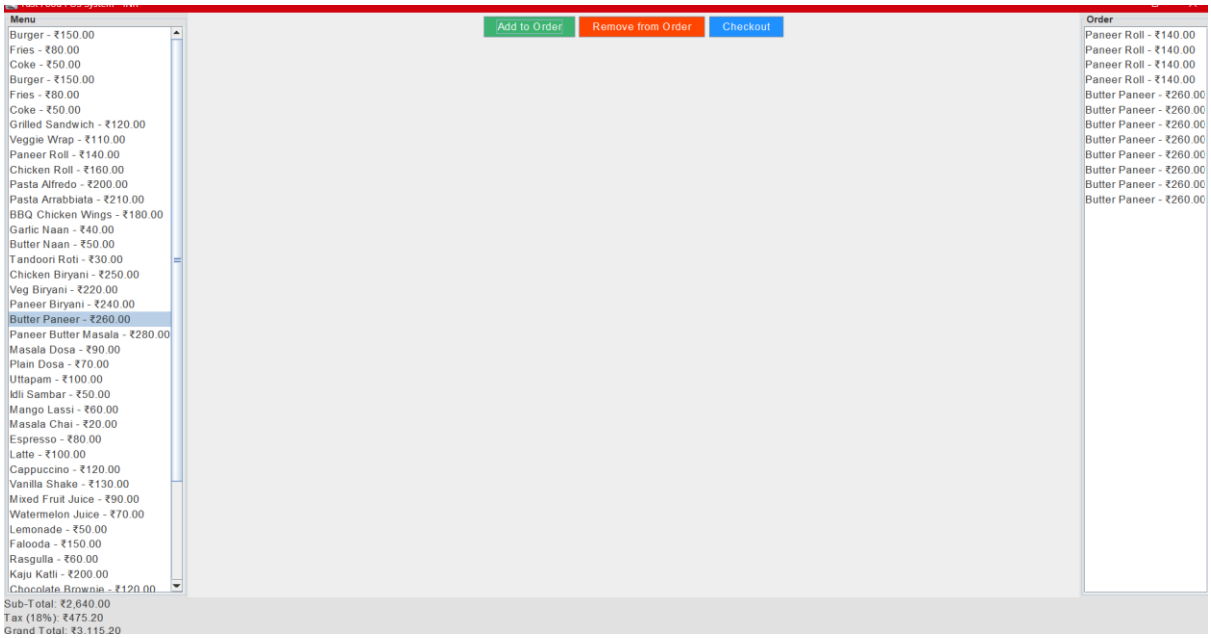
```
        menuListModel.addElement(new MenuItem("Latte", 100.0, "Drink"));
        menuListModel.addElement(new MenuItem("Cappuccino", 120.0, "Drink"));
        menuListModel.addElement(new MenuItem("Vanilla Shake", 130.0, "Drink"));
        menuListModel.addElement(new MenuItem("Mixed Fruit Juice", 90.0, "Drink"));
        menuListModel.addElement(new MenuItem("Watermelon Juice", 70.0, "Drink"));
        menuListModel.addElement(new MenuItem("Lemonade", 50.0, "Drink"));
        menuListModel.addElement(new MenuItem("Falooda", 150.0, "Dessert"));
        menuListModel.addElement(new MenuItem("Rasgulla", 60.0, "Dessert"));
        menuListModel.addElement(new MenuItem("Kaju Katli", 200.0, "Dessert"));
        menuListModel.addElement(new MenuItem("Chocolate Brownie", 120.0, "Dessert"));
        menuListModel.addElement(new MenuItem("Hot Chocolate Fudge", 160.0, "Dessert"));
        menuListModel.addElement(new MenuItem("Fruit Salad", 80.0, "Dessert"));
        menuListModel.addElement(new MenuItem("Apple Pie", 140.0, "Dessert"));
        menuListModel.addElement(new MenuItem("Banana Split", 150.0, "Dessert"));
        menuListModel.addElement(new MenuItem("Chocolate Cake", 180.0, "Dessert"));
        menuListModel.addElement(new MenuItem("Black Forest Cake", 200.0, "Dessert"));
        menuListModel.addElement(new MenuItem("Vanilla Cupcake", 90.0, "Dessert"));
        menuListModel.addElement(new MenuItem("Strawberry Cupcake", 90.0, "Dessert"));

// Remaining items in original code if any...

        // Add other items as needed...

        // Menu Panel
        JPanel menuPanel = new JPanel(new BorderLayout());
        menuPanel.setBorder(BorderFactory.createTitledBorder("Menu"));
        menuPanel.setBackground(panelBgColor);
        menuList = new JList<>(menuListModel);
        menuList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        menuList.setFont(mainFont);
        menuPanel.add(new JScrollPane(menuList), BorderLayout.CENTER);

        // Order Panel
        JPanel orderPanel = new JPanel(new BorderLayout());
        orderPanel.setBorder(BorderFactory.createTitledBorder("Order"));
        orderPanel.setBackground(panelBgColor);
        orderList = new JList<>(orderListModel);
        orderList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        orderList.setFont(mainFont);
        orderPanel.add(new JScrollPane(orderList), BorderLayout.CENTER);

        // Buttons Panel
        JPanel buttonPanel = new JPanel();
        JButton addButton = new JButton("Add to Order");
        JButton removeButton = new JButton("Remove from Order");
        JButton checkoutButton = new JButton("Checkout");

        // Style buttons
```

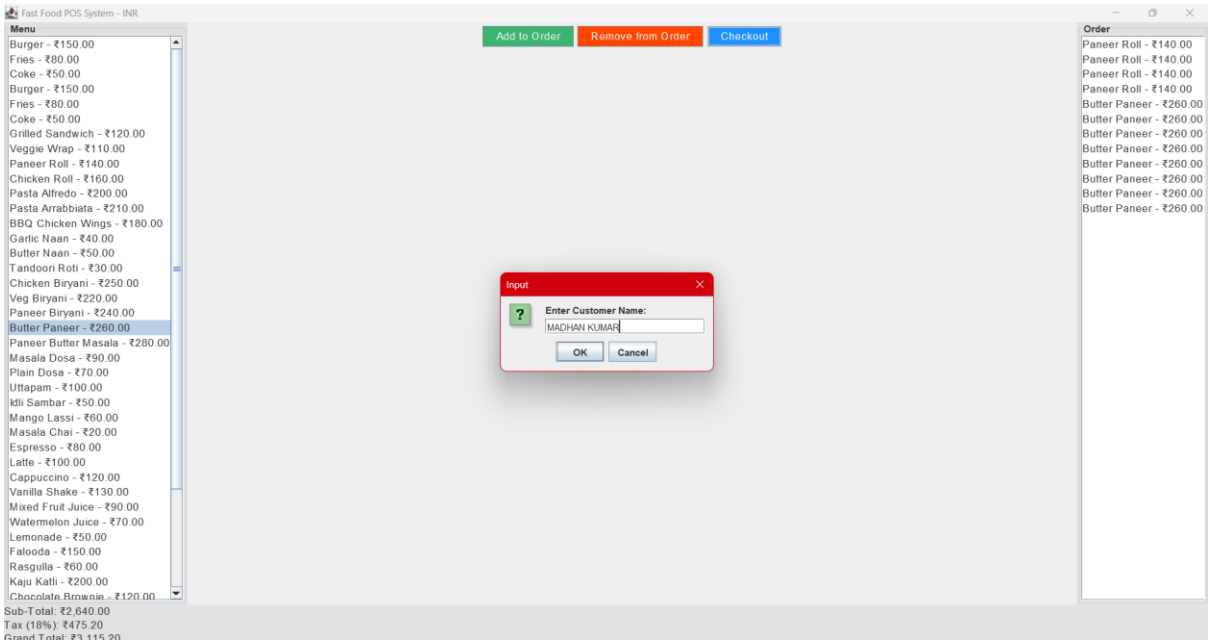
```
addButton.setBackground(buttonColor);
addButton.setForeground(Color.WHITE);
removeButton.setBackground(new Color(255, 69, 0)); // Red-orange for remove
removeButton.setForeground(Color.WHITE);
checkoutButton.setBackground(new Color(30, 144, 255)); // Dodger blue for checkout
checkoutButton.setForeground(Color.WHITE);

addButton.setFont(mainFont);
removeButton.setFont(mainFont);
checkoutButton.setFont(mainFont);

buttonPanel.add(addButton);
buttonPanel.add(removeButton);
buttonPanel.add(checkoutButton);

// Total Panel
JPanel totalPanel = new JPanel(new GridLayout(3, 1));
subTotalLabel = new JLabel("Sub-Total: ₹0.00");
taxLabel = new JLabel("Tax (18%): ₹0.00");
totalLabel = new JLabel("Grand Total: ₹0.00");

subTotalLabel.setFont(mainFont);
taxLabel.setFont(mainFont);
totalLabel.setFont(mainFont);
totalPanel.setBackground(panelBgColor);
totalPanel.add(subTotalLabel);
totalPanel.add(taxLabel);
totalPanel.add(totalLabel);

// Add listeners
addButton.addActionListener(e -> {
  MenuItem selectedItem = menuList.getSelectedValue();
  if (selectedItem != null) {
    order.addItem(selectedItem);
    orderListModel.addElement(selectedItem);
    updateTotals(order);
  }
});

removeButton.addActionListener(e -> {
  MenuItem selectedItem = orderList.getSelectedValue();
  if (selectedItem != null) {
    order.removeItem(selectedItem);
    orderListModel.removeElement(selectedItem);
    updateTotals(order);
  }
});
```

```
    checkoutButton.addActionListener(e -> {
        String customerName = JOptionPane.showInputDialog("Enter Customer Name:");
        if (customerName != null && !customerName.trim().isEmpty()) {
            double totalAmount = order.calculateTotal();
            saveToDatabase(customerName, totalAmount); // Save to database
            JOptionPane.showMessageDialog(FastFoodPOS.this,
                    "Customer: " + customerName + "\n" +
                            "Sub-Total: " + formatCurrency(order.calculateSubTotal()) + "\n" +
                            "Tax: " + formatCurrency(order.calculateTax()) + "\n" +
                            "Grand Total: " + formatCurrency(totalAmount),
                    "Checkout",
                    JOptionPane.INFORMATION_MESSAGE);
            order.getItems().clear();
            orderListModel.clear();
            updateTotals(order);
        }
    });

    // Assemble the UI
    add(menuPanel, BorderLayout.WEST);
    add(orderPanel, BorderLayout.EAST);
    add(buttonPanel, BorderLayout.CENTER);
    add(totalPanel, BorderLayout.SOUTH);
}

private void updateTotals(Order order) {
    subTotalLabel.setText("Sub-Total: " + formatCurrency(order.calculateSubTotal()));
    taxLabel.setText("Tax (18%): " + formatCurrency(order.calculateTax()));
    totalLabel.setText("Grand Total: " + formatCurrency(order.calculateTotal()));
}

private String formatCurrency(double amount) {
    NumberFormat currencyFormat = NumberFormat.getCurrencyInstance(new Locale("en", "IN"));
    return currencyFormat.format(amount);
}

private void saveToDatabase(String customerName, double totalAmount) {
    String url = "jdbc:mysql://localhost:3306/FastFoodPOS";
    String user = "root";
    String password = "Madhan@123"; // Replace with your MySQL password

    String insertSQL = "INSERT INTO CustomerRecords (customer_name, total_amount) VALUES (?, ?)";

    try (Connection connection = DriverManager.getConnection(url, user, password);
         PreparedStatement preparedStatement = connection.prepareStatement(insertSQL)) {

        preparedStatement.setString(1, customerName);
        preparedStatement.setDouble(2, totalAmount);
```

```
            preparedStatement.executeUpdate();

            System.out.println("Record saved: " + customerName + ", ₹" + totalAmount);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new FastFoodPOS().setVisible(true));
    }
}
```

OUTPUT:





RESULT:

THE GIVE PROJECT IS EXECUTED SUCCESFULLY.

TEAM MEMBERS:

MADHAN KUMAR B 231901028

KEERTHANA 231901022