

**International Institute of
Professional Studies, davv**

Line following robot System

Guided By:

Manas Patidar

Dr.Shaligram Prajapat Sir

IT-2k19-28

BONAFIDE CERTIFICATE

Certified that this project titled " line following robot System " is a bonafide work of MANAS PATIDAR(IT-2K19-28) who carried out the researched and completed the project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not

form part of any other project on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Internal examiner

External examiner

RECOMMENDATIONS

The Project is on line following robot System" submitted by MANAS PATIDAR is satisfactory account of the bona fide work under my supervision and is recommended towards the end of their III year of MTECH(I.T)- 2024.

Guided By:

ACKNOWLEDGEMENT

I would like to express my gratitude toward staff of IIPS -DAVV as well as the honourable director Dr. B. K. Tripathi for providing us a great opportunity to complete a project on "line following robot System". My sincere thanks go to Dr. Shaligram Prajapat without his support and guidance for the completion of this project.

6.1 Database Design:

An important aspect of system design is the design of data storage structure. To begin with a logical model of data structure is developed first. A database is a container object which contains tables, queries, reports and data validation policies enforcement rules or constraints etc. A logical data often represented as a records are kept in different tables after reducing anomalies and redundancies. The goodness of data base design lies in the table structure and its relationship.

*This software project maintains a database named **shopkeeper** which contains the following tables.*

Table Design:

The database of shopkeeper System contains 5 tables. The tables are normalized to minimize the redundancies of data and enforcing the validation rules of the organization. Most of the tables are designed to store master records. The tables and their structure are given below.

Table: shopkeeper

Column Name	Type	Size
Shooper _id	Integer	100
name	Varchar	100
city	Varchar	100
phone	Varchar	11
address	Varchar	40

Table: item

Column Name	Type	Size
-------------	------	------

Item_id	Integer	100
Item_name	varchar	100
description	Varchar	1000
price	float	(11,2)

Table: orderitem

<i>Column Name</i>	<i>Type</i>	<i>Size</i>
orderno	varchar	100
item_id	int	100
shopper_id	Varchar	100
quantity	int	100
Order date	Date	
price	float	(11,2)
discount	Float	(11,2)
amount	float	(11,2)

6.2 Menu Design:

JSS Infoware gateway comprises the following options, organized in a user friendly way. The menu system divided in Menu Bars, each having a pull down menus containing options for a specific task,

Sr.	Menu Bar	Pull Down Menu	Purpose	Forms Attached
1.	Robot Parts	Custmor entry	Insertion of Custmor records.	PubUI.java
		Item entry	Insertion of item records.	PubEditUI.java
		Order entry	Insertion of Order records.	PubDelUI.java
2.	view	Custmor list	view Custmor list	LibUI.java
		Item list	view Item list	LibEditUI.java
		Order list	view Order list	LibDelUI.java
5.	Quit	Application	Close the Application.	--

6.3 I/O Forms Design & Event Coding:

The software project for Public Library Management contains various forms along with programming codes. Forms (JFrames) and their event coding are given below.

Frame: MainUI.java

Coding for MainUI.java

```
public class MainMenuUI extends javax.swing.JFrame {
```

```
/** Creates new form MainMenuUI */  
  
public MainMenuUI() {  
    initComponents();  
}  
  
private void mnuCustAddActionPerformed(java.awt.event.ActionEvent evt) {  
    this.setVisible(false);  
    new robotline().setVisible(true);  
}  
  
private void mnulItemAddActionPerformed(java.awt.event.ActionEvent evt) {  
    this.setVisible(false);  
    new ItemINUI().setVisible(true);  
}  
  
private void mnuOrderActionPerformed(java.awt.event.ActionEvent evt) {  
    this.setVisible(false);  
    new OrderINUI().setVisible(true);  
}  
  
private void jMenu1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}  
  
private void jMenu2ActionPerformed(java.awt.event.ActionEvent evt) {  
}  
  
private void ListCustActionPerformed(java.awt.event.ActionEvent evt) {  
    this.setVisible(false);  
    new CustListUI().setVisible(true);  
}
```

```

}

private void ListItemActionPerformed(java.awt.event.ActionEvent evt) {

    this.setVisible(false);

    new ItemListUI().setVisible(true);
}

private void ListOrderActionPerformed(java.awt.event.ActionEvent evt) {

    this.setVisible(false);

    new OrdListUI().setVisible(true);
}

private void jMenu3ActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {

System.exit(0);    // TODO add your handling code here:

}

/**
 * @param args the command line arguments
 */

public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            new MainMenuUI().setVisible(true);

        }

    });

}

```


Coding of robotline

```
import java.sql.*;

import javax.swing.JOptionPane;

public class ShopINUI extends javax.swing.JFrame {

    /** Creates new form ShopINUI */
    public robotline() {
        initComponents();
    }

    private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {

        this.setVisible(false);

        new MainMenuUI().setVisible(true);
    }

    private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {

        try {

            Class.forName("com.mysql.jdbc.Driver");

            Connection con = (Connection)

                DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","kvuc");

            Statement stmt = null;

            ResultSet rs = null;    // ResultSet for SHOPKEEPER table.

            String SQL = "SELECT * FROM shopkeeper";
```

```

stmt = con.createStatement(); // Connection string for ResultSet - rs.

rs = stmt.executeQuery(SQL);


// Data transfer from JTextField control to variables

String shno = txtSHno.getText().trim();

String shname = txtSHName.getText();

String shadd = txtSHAddress.getText();

String shcity = txtSHCity.getText();

double shphone = Double.parseDouble(txtSHPhone.getText());

if (shno.length() <= 4) {

    String strSQL = "INSERT INTO shopkeeper(Shopper_id, Name, Address, City, Phone) VALUES
('"+(shno)+"','"+(shname)+"','"+(shadd)+"','"+(shcity)+"','"+(shphone)+"'");

    JOptionPane.showMessageDialog(this, "Record successfully inserted");

    int rowsEffectd = stmt.executeUpdate(strSQL);

    System.out.println(rowsEffectd + " rows effected");

}

else {

    JOptionPane.showMessageDialog(this, "Customer ID should not more than 4 character.");

}

con.close();

} catch (Exception e) {

    JOptionPane.showMessageDialog(this,e.getMessage());

    e.printStackTrace();

}

}

```

```

private void btnClearActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    txtSHno.setText("");

    txtSHName.setText("");

    txtSHAddress.setText("");

    txtSHCity.setText("");

    txtSHPhone.setText("");

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            new ShopINUI().setVisible(true);

        }

    });

}

```

Sports item input screen

Sports Item Input Form

Item Details...

Item Id.:

Item Name:

Description:

Price:

Coding for ItemINUI.java

```
import java.sql.*;
```

```
import javax.swing.JOptionPane;
```

```
public class ItemINUI extends javax.swing.JFrame {
```

```
    /** Creates new form ItemINUI */
```

```
    public ItemINUI() {
```

```
        initComponents();
```

```
    }
```

```
    private void btnClearActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        // TODO add your handling code here:
```

```
        txtItemno.setText("");
```

```
        txtItemName.setText("");
```

```
        txtItemDesc.setText("");
```

```

        txtItemPrice.setText("");
    }

    private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {

        try {

            Class.forName("com.mysql.jdbc.Driver");

            Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","kvuc");

            Statement stmt = null;

            ResultSet rs = null;    // ResultSet for SHOPKEEPER table.

            String SQL = "SELECT * FROM item";

            stmt = con.createStatement(); // Connection string for ResultSet - rs.

            rs = stmt.executeQuery(SQL);

            // Data transfer from JTextField control to variables

            String itemno = txtItemno.getText();

            String itemname = txtItemName.getText();

            String desc = txtItemDesc.getText();

            double itemprice = Double.parseDouble(txtItemPrice.getText());

            String strSQL = "INSERT INTO item(Item_Id, Item_Name, Description, Price) VALUES
('"+(itemno)+"','"+(itemname)+"','"+(desc)+"','"+(itemprice)+"')";

            JOptionPane.showMessageDialog(this, "Record successfully inserted");

            int rowsEffectd = stmt.executeUpdate(strSQL);

            System.out.println(rowsEffectd + " rows effected");

            con.close();

        } catch (Exception e) {

            JOptionPane.showMessageDialog(this,e.getMessage());

            e.printStackTrace();

```

```

    }
}

private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {

    this.setVisible(false);

    new MainMenuUI().setVisible(true);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            new ItemINUI().setVisible(true);

        }

    });
}

```

The screenshot shows a Java Swing window titled "Order Form". Inside the window, the title "Sports Item Order Form" is centered at the top. The form is organized into two main columns. The left column contains fields for "Order No.:", "Date:", "Shopkeeper Id and Name" (a list box with items "100 - Deepak yadav", "101 - ankit aswal", and "102 - Issarar ali"), "ID.:", "Name:", "Order Quantity:", and a "Save Order" button. The right column contains fields for "Item Id and Name" (a list box with items "100 - SHOE'S", "101 - TRACK SUIT", and "103 - BAT"), "Item ID.:", "Item price:", a "Discount" section with "Yes" and "No" radio buttons, "Discount (%)":, and a "Clear" button. At the bottom center, there is a "Back to Main" button. The window has standard Windows-style window controls (minimize, maximize, close) in the top right corner.

Coding for OrderINUI.Java

```
import javax.swing.DefaultListModel;

import java.sql.*;

import javax.swing.JOptionPane;

public class OrderINUI extends javax.swing.JFrame {
```

```

/** Creates new form OrderINUI */
public OrderINUI() {
    initComponents();
}

private void btnClearActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    txtOrdno.setText("");

    txtOrdDate.setText("");

    txtSID.setText("");

    txtItemID.setText("");

    txtSName.setText("");

    txtItemPrice.setText("");

    txtOrdQty.setText("");
}

private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {

    this.setVisible(false);

    new MainMenuUI().setVisible(true);
}

private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        Class.forName("com.mysql.jdbc.Driver");

        Connection con = (Connection)

            DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","kvuc");
    }
}

```



```

Statement stmt = null;

ResultSet rs = null;    // ResultSet for SHOPKEEPER table.

String SQL = "SELECT * FROM orderitem";

stmt = con.createStatement(); // Connection string for ResultSet - rs.

rs = stmt.executeQuery(SQL);


float discP = 0; // Discount

// Data transfer from JTextField control to variables

String Ordno = txtOrdno.getText();

String OrdDate = txtOrdDate.getText();

String ItemID = txtItemID.getText();

String ShopID = txtSID.getText();

String ItemPrice = txtItemPrice.getText();

double iPrice = Double.parseDouble(txtItemPrice.getText());

String ordQty = txtOrdQty.getText();


if (rdYes.isSelected()) {

    discP = Float.parseFloat(txtDisc.getText());

}

else {

    discP=0;

}


double amt = (Integer.parseInt(txtOrdQty.getText()) *
Double.parseDouble(txtItemPrice.getText())) - (Integer.parseInt(txtOrdQty.getText()) *
Double.parseDouble(txtItemPrice.getText()))*(discP*0.01);

```

```
String strSQL = "INSERT INTO orderitem(orderno, OrderDate, Item_Id, Shopper_Id, Quantity,  
price, discount, Amount ) VALUES  
('"+(Ordno)+"','"+(OrdDate)+"','"+(ItemID)+"','"+(ShopID)+"','"+(ordQty)+"','"+iPrice+"','"+  
discP+"','"+(amt)+"");
```

```
JOptionPane.showMessageDialog(this, "Order successfully placed");
```

```
int rowsEffectd = stmt.executeUpdate(strSQL);
```

```
System.out.println(rowsEffectd + " rows effected");
```

```
con.close();
```

```
} catch (Exception e) {
```

```
JOptionPane.showMessageDialog(this,e.getMessage());
```

```
e.printStackTrace();
```

```
}
```

```
}
```

```
private void formWindowGainedFocus(java.awt.event.WindowEvent evt) {
```

```
// Creating a ListModel object sModel to perform DefaultListModel
```

```
// method operations for Shopkeeper list
```

```
DefaultListModel sModel = (DefaultListModel) jList1.getModel();
```

```
// Creating a ListModel object iModel to perform DefaultComboBoxModel
```

```
// method operations for Item list
```

```
DefaultListModel iModel = (DefaultListModel) jList2.getModel();
```

```
sModel.clear();
```

```
iModel.clear();
```

```
try {
```

```
Class.forName("com.mysql.jdbc.Driver");

Connection con = (Connection)

    DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","kvuc");

Statement stmt = null;

ResultSet rs = null;

ResultSet rs1 = null;


String SQL = "SELECT * FROM shopkeeper";

String SQL1 = "SELECT * FROM item";

stmt = con.createStatement();


// Steps to extract shopkeepers id and name

rs = stmt.executeQuery(SQL);

while (rs.next()) {

    String sID = rs.getString("Shopper_id");

    String Sname = rs.getString("Name");

    sModel.addElement(sID + " - " + Sname);

}

jList1.setModel(sModel);


// Steps to extract item id and name

rs1 = stmt.executeQuery(SQL1);

while (rs1.next()) {

    String iID = rs1.getString("Item_Id");

    String Iname = rs1.getString("Item_Name");
```

```

        iModel.addElement(iID + " - " + lName);
    }

    jList2.setModel(iModel);

    con.close();

} catch (Exception e) {

    JOptionPane.showMessageDialog(this,e.getMessage());

    e.printStackTrace();

}

}

private void jList1MouseClicked(java.awt.event.MouseEvent evt) {

    // Extracting supplier id and name into a variable SidName

    String SidName = (String) jList1.getSelectedValue();

    String Sid = SidName.substring(0, 3);

    String Sname= SidName.substring(6);

    // Displays ID and name from ComboBox1

    txtSID.setText(Sid);

    txtSName.setText(Sname);

}

private void jList2MouseClicked(java.awt.event.MouseEvent evt) {

    try {

```

```

Class.forName("com.mysql.jdbc.Driver");

Connection con = (Connection)

    DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","kvuc");

Statement stmt = null;

ResultSet rs = null;


// Extracting item id and item name into a variable ItemIDName

String ItemIDName = (String) jList2.getSelectedValue();

String ItemID = ItemIDName.substring(0, 3);

String lame= ItemIDName.substring(7);

txtItemID.setText(ItemID);

String SQL = "SELECT * FROM item where Item_Id = '"+(ItemID)+"'";

stmt = con.createStatement();

rs = stmt.executeQuery(SQL);


while (rs.next()) {

    double iprice = rs.getDouble("Price");

    txtItemPrice.setText(Double.toString(iprice));

}

con.close();

} catch (Exception e) {

    JOptionPane.showMessageDialog(this,e.getMessage());

    e.printStackTrace();

}

```

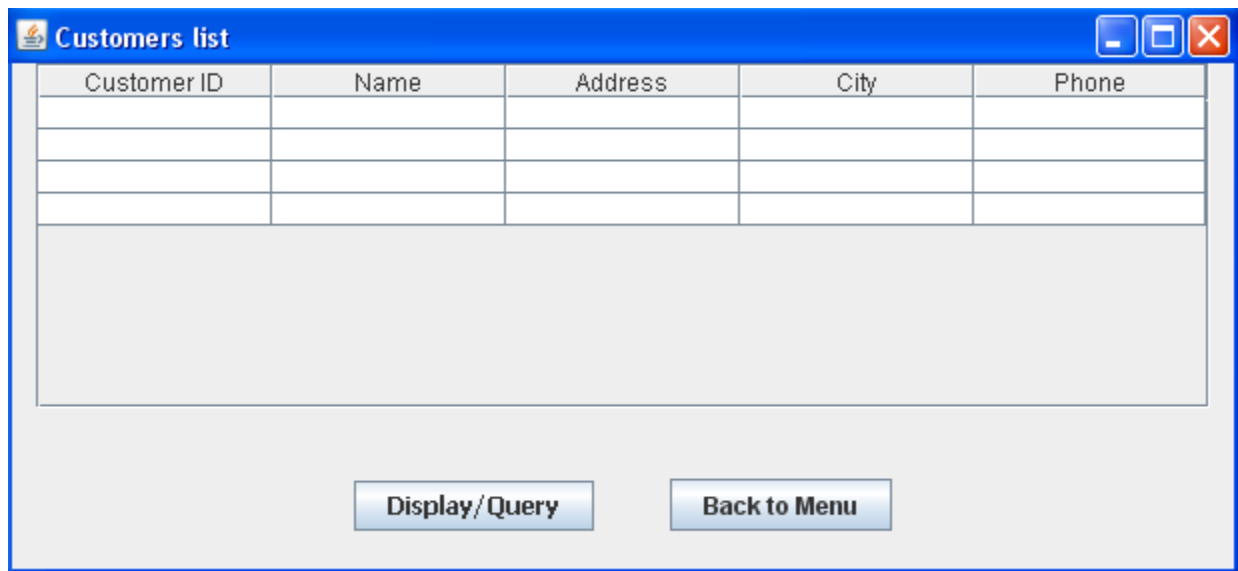
```

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new OrderINUI().setVisible(true);
        }
    });
}
}

```

Frame: CustListUI.java



Coding of CustListUI.java

```
import javax.swing.table.*;
```

```
import java.sql.*;
```

```
import javax.swing.JOptionPane;
```

```
/*
```

```
 * To change this template, choose Tools | Templates
```

```
 * and open the template in the editor.
```

```
 */
```

```
/*
```

```
 * CustListUI.java
```

```
 *
```

```
 * Created on Mar 18, 2010, 1:24:42 PM
```

```
 */
```

```
/**
```

```
 *
```

```
 * @author gagan
```

```
 */
```

```
public class CustListUI extends javax.swing.JFrame {
```

```
    /** Creates new form CustListUI */
```

```
    public CustListUI() {
```

```
        initComponents();
```

```
    }
```

```
    /** This method is called from within the constructor to
```

```
     * initialize the form.
```

*** WARNING: Do NOT modify this code. The content of this method is**

*** always regenerated by the Form Editor.**

***/**

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

setTitle("Customers list");

jButton1.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N

jButton1.setText("Display/Query");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Back to Menu");

jButton2.addActionListener(new java.awt.event.ActionListener() {


```

        .addContainerGap()

        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 587,
Short.MAX_VALUE))

        .addGroup(layout.createSequentialGroup())

        .addGap(171, 171, 171)

        .addComponent(jButton1)

        .addGap(38, 38, 38)

        .addComponent(jButton2)))

    .addContainerGap())

);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 172,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(35, 35, 35)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(jButton1)

                .addComponent(jButton2))

            .addContainerGap(19, Short.MAX_VALUE))

);

pack();

} // </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

```

```

// Before writting the followng line, you should import the line:

// import javax.swing.table.*; at the top of your application

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

// Clear the existing table

int rows = model.getRowCount();

if (rows > 0) {

    for (int i = 0; i < rows; i++) {

        model.removeRow(0);

    }

}

// SQL Query

String query = "SELECT * FROM shopkeeper";

try {

    // Connect to MySQL database

    Class.forName("com.mysql.jdbc.Driver").newInstance();

    Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","kvuc");

    // Create SQL statement and execute query.

    Statement stmt = con.createStatement();

    ResultSet rs = stmt.executeQuery(query);


    // Iterate through the result and display on screen

    while (rs.next()) {

        String Sid = rs.getString("Shopper_id");

        String SName = rs.getString("Name");

        String SAddress = rs.getString("Address");

```

```

        String SCity = rs.getString("City");

        String SPhone = rs.getString("Phone");

        System.out.println(Sid + "|" + SName + "|" + SAddress + "|" + SCity + "|" + SPhone);

        model.addRow(new Object[] {Sid, SName, SAddress, SCity, SPhone});

    }

} catch (Exception e) {

    JOptionPane.showMessageDialog(this, e.getMessage());

}

}

```

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    this.setVisible(false);

    new MainMenuUI().setVisible(true);

}

```

```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            new CustListUI().setVisible(true);

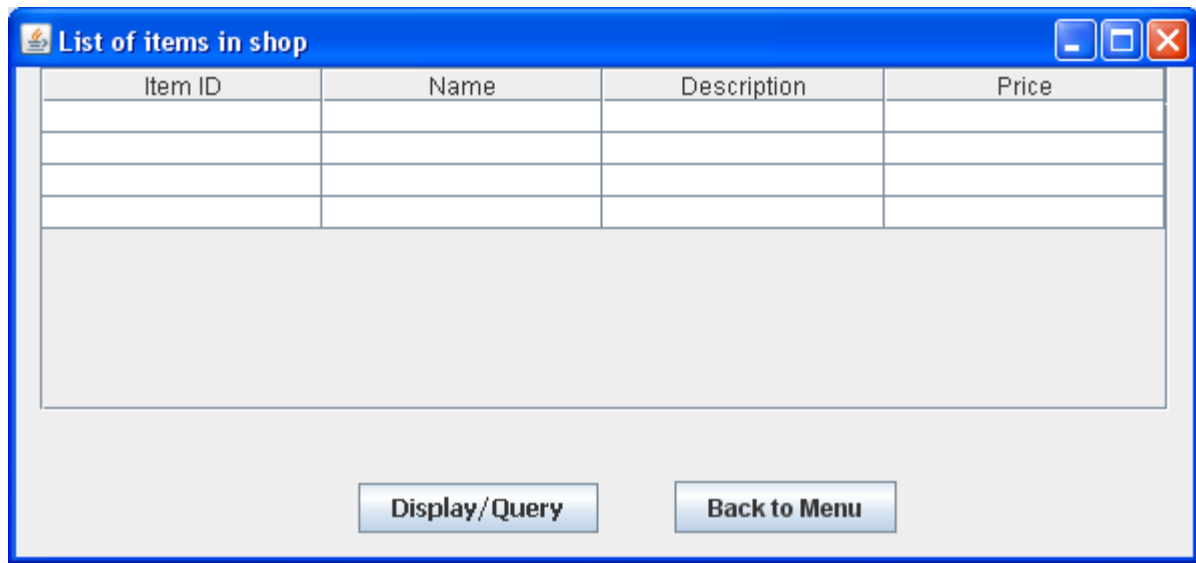
        }

    });

}

```

Frame: ItemListUI.java



Coding for ItemListUI.java

```
import javax.swing.table.*;
```

```
import java.sql.*;
```

```
import javax.swing.JOptionPane;
```

```
/*
```

```
 * To change this template, choose Tools | Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
/*
```

```
 * ItemListUI.java
```

```
 *
```

*** Created on Mar 18, 2010, 1:45:12 PM**

***/**

/**

*** @author gagan**

***/**

public class ItemListUI extends javax.swing.JFrame {

/ Creates new form ItemListUI */**

public ItemListUI() {

initComponents();

}

/ This method is called from within the constructor to**

*** initialize the form.**

*** WARNING: Do NOT modify this code. The content of this method is**

*** always regenerated by the Form Editor.**

***/**

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jButton2 = new javax.swing.JButton();

jButton1 = new javax.swing.JButton();

```
jScrollPane1 = new javax.swing.JScrollPane();
```

```
jTable1 = new javax.swing.JTable();
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
```

```
setTitle("List of items in shop");
```

```
jButton2.setText("Back to Menu");
```

```
jButton2.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        jButton2ActionPerformed(evt);
```

```
    }
```

```
});
```

```
jButton1.setFont(new java.awt.Font("Tahoma", 1, 12));
```

```
jButton1.setText("Display/Query");
```

```
jButton1.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        jButton1ActionPerformed(evt);
```

```
    }
```

```
});
```

```
jTable1.setModel(new javax.swing.table.DefaultTableModel(
```

```
    new Object [][] {
```

```
        {null, null, null, null},
```

```
        {null, null, null, null},
```

```

        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Item ID", "Name", "Description", "Price"
    }
));

jScrollPane1.setViewportViewView(jTable1);

```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());  
getContentPane().setLayout(layout);  
  
layout.setHorizontalGroup(  
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(layout.createSequentialGroup()  
            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)  
                .addGroup(layout.createSequentialGroup()  
                    .addGap(171, 171, 171)  
                    .addComponent(jButton1)  
                    .addGap(38, 38, 38)  
                    .addComponent(jButton2)))  
            .addContainerGap())  
);
```



```

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 172,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(35, 35, 35)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(jButton1)

                .addComponent(jButton2))

            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

        );

pack();
} // </editor-fold>

```

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    this.setVisible(false);

    new MainMenuUI().setVisible(true);

}

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    // Before writting the followng line, you should import the line:

    // import javax.swing.table.*; at the top of your application

    DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

    // Clear the existing table

    int rows = model.getRowCount();

```

```

if (rows > 0) {

    for (int i = 0; i < rows; i++) {

        model.removeRow(0);

    }

}

// SQL Query

String query = "SELECT * FROM Item";

try {

    // Connect to MySQL database

    Class.forName("com.mysql.jdbc.Driver").newInstance();

    Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","kvuc");

    // Create SQL statement and execute query.

    Statement stmt = con.createStatement();

    ResultSet rs = stmt.executeQuery(query);


    // Iterate through the result and display on screen

    while (rs.next()) {

        String ITid = rs.getString("Item_id");

        String IName = rs.getString("Item_Name");

        String IDesc = rs.getString("Description");

        String IPrice = rs.getString("Price");

        System.out.println(ITid + "|" + IName + "|" + IDesc + "|" + IPrice);

        model.addRow(new Object[] {ITid, IName, IDesc, IPrice});

    }

} catch (Exception e) {

```

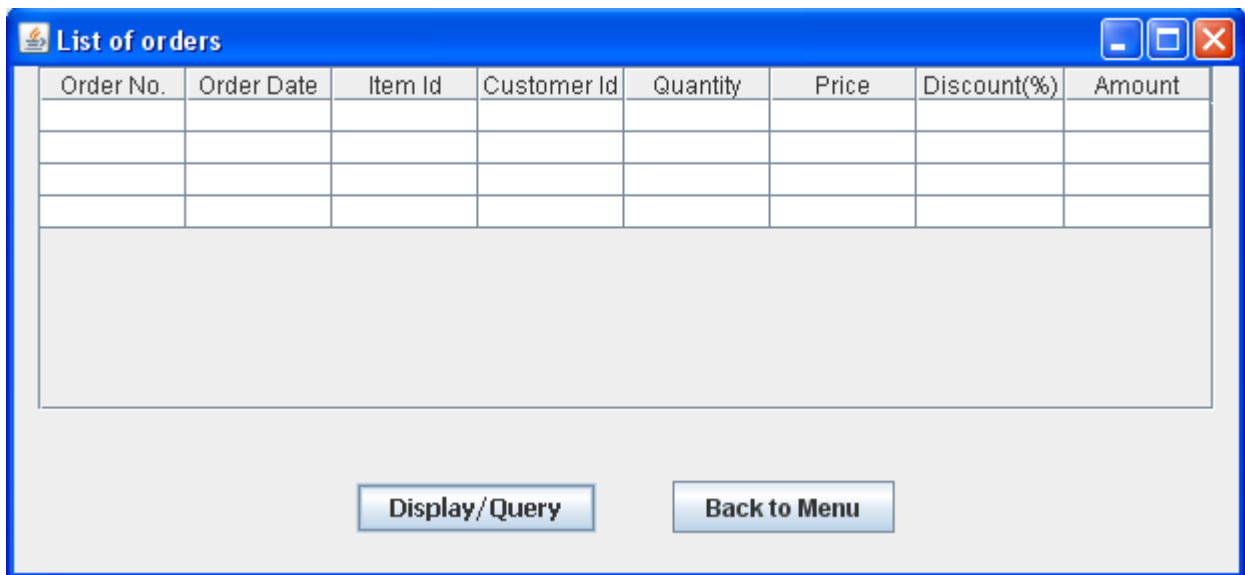
```

        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ItemListUI().setVisible(true);
        }
    });
}

```

Frame: OrdListUI.java



Order No.	Order Date	Item Id	Customer Id	Quantity	Price	Discount(%)	Amount

Display/Query
Back to Menu

Coding for OrdListUI.java

```
import javax.swing.table.*;
```

```
import java.sql.*;
```

```
import javax.swing.JOptionPane;
```

```
/*
```

```
 * To change this template, choose Tools | Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
/*
```

```
 * OrdListUI.java
```

```
 *
```

```
 * Created on Mar 18, 2010, 1:50:20 PM
```

```
*/
```

```
/**
```

```
 *
```

```
 * @author gagan
```

```
*/
```

```
public class OrdListUI extends javax.swing.JFrame {
```

```
    /** Creates new form OrdListUI */
```

```
    public OrdListUI() {
```

```

        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jButton2 = new javax.swing.JButton();
        jButton1 = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("List of orders");

        jButton2.setText("Back to Menu");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        }

```

```
});
```

```
jButton1.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
```

```
jButton1.setText("Display/Query");
```

```
jButton1.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        jButton1ActionPerformed(evt);
```

```
    }
```

```
});
```

```
jTable1.setModel(new javax.swing.table.DefaultTableModel(  
    new Object [][] {
```

```
        {null, null, null, null, null, null, null, null},
```

```
        {null, null, null, null, null, null, null, null},
```

```
        {null, null, null, null, null, null, null, null},
```

```
        {null, null, null, null, null, null, null, null}
```

```
    },
```

```
    new String [] {
```

```
        "Order No.", "Order Date", "Item Id", "Customer Id", "Quantity", "Price", "Discount(%)",  
        "Amount"
```

```
    }
```

```
));
```

```
jScrollPane1.setViewportView(jTable1);
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
```

```
getContentPane().setLayout(layout);
```

```

layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup())
                    .addContainerGap()
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 588,
Short.MAX_VALUE))
                .addGroup(layout.createSequentialGroup())
                    .addGap(171, 171, 171)
                    .addComponent(jButton1)
                    .addGap(38, 38, 38)
                    .addComponent(jButton2)))
            .addContainerGap())
);

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 172,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(35, 35, 35)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jButton1)
                .addComponent(jButton2))
            .addContainerGap(20, Short.MAX_VALUE))
);

```

```

        pack();
    } // </editor-fold>

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        new MainMenuUI().setVisible(true);
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // Before writting the followng line, you should import the line:
        // import javax.swing.table.*; at the top of your application
        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

        // Clear the existing table
        int rows = model.getRowCount();

        if (rows > 0) {
            for (int i = 0; i < rows; i++) {
                model.removeRow(0);
            }
        }

        // SQL Query
        String query = "SELECT * FROM orderitem";

        try {
            // Connect to MySQL database
            Class.forName("com.mysql.jdbc.Driver").newInstance();

```



```

        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","kvuc");

        // Create SQL statement and execute query.

        Statement stmt = con.createStatement();

        ResultSet rs = stmt.executeQuery(query);

        // Iterate through the result and display on screen

        while (rs.next()) {

            String Ordno = rs.getString("Orderno");

            String Ordd = rs.getString("OrderDate");

            String Oltemid = rs.getString("Item_Id");

            String OSid = rs.getString("Shopper_Id");

            String OQty = rs.getString("Quantity");

            String IPrice = rs.getString("price");

            String Disc = rs.getString("discount");

            String OAmount = rs.getString("Amount");

            System.out.println(Ordno + "|" + Ordd + "|" + Oltemid + "|" + OSid + "|" + OQty + "|" + IPrice
+ "|" + Disc + "|" + OAmount);

            model.addRow(new Object[] {Ordno, Ordd, Oltemid, OSid, OQty, IPrice, Disc, OAmount});

        }

    } catch (Exception e) {

        JOptionPane.showMessageDialog(this, e.getMessage());

    }

}

/**
 * @param args the command line arguments

```

```

*/

public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            new OrdListUI().setVisible(true);

        }

    });

}

```

QUIT



Coding of QUIT

```

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {

    System.exit(0);    // TODO add your handling code here:

}

```

