



# Java Institute for Advanced Technology

[Project Document]

[UM.Mawantha Deeganu]

[981441150V]

GAMPAHA



# **GOLDEN Jewellery Shop**

1. Customer Registration.....
2. Supplier Registration.....
3. Employee Registration.....
4. Branch Registration.....
5. Owner Registration.....
6. Add Jewelry.....
7. Stock.....
8. Reports.....
  - I. GRN
  - II. Invoice
  - III. Stock
  - IV. Employee
  - V. Supplier
9. Exchange Item.....
10. Company Registration.....
11. Branch address.....

## 1. Customer Registration

**Jewelry SHOP**

08:06:22 PM

**Customer Registration**

Search Customer

First Name:

Last Name:

Contact No:

City:

Create Account

Update

Add City

ID	First Name	Last Name	Contact No	City
9	Mawantha	Deeganu	0753267252	Kaluthar
11	Kasuni	Sankalpa	0762258952	Baticloew
12	amal	perera	0762265565	Baticloew
13	kasun	nirmal	0753254525	Trinco
14	nirmala	jayasanka	0762333658	Colombo
15	kasun	nirmal	0753254525	Trinco
16	nadi	nisansala	0728569874	Kandy

To load data into the table –

```
public void loadcustomer() {  
    try {  
        ResultSet rs = MySql.search("SELECT  
`user`.`id`,`user`.`first_name`,`user`.`last_name`,`user`.`contact_number`,`city`.`name` AS  
`city_name` FROM `user` INNER JOIN `city` ON `user`.`city_id`=`city`.`id`;");  
        //DefaultTableModel dtm = new DefaultTableModel();  
        DefaultTableModel dtm = (DefaultTableModel) jTable3.getModel();  
        dtm.setRowCount(0);
```

```
while (rs.next()) {

    Vector v = new Vector();

    v.add(rs.getString("id"));

    v.add(rs.getString("first_name"));

    v.add(rs.getString("last_name"));

    v.add(rs.getString("contact_number"));

    v.add(rs.getString("city_name"));

    dtm.addRow(v);

}

jTable3.setModel(dtm);

} catch (Exception e) {

    e.printStackTrace();

}

}
```

## Search –

```
public void loadcustomer(String text) {

    try {

        ResultSet rs = MySql.search("SELECT
`user`.`id`,`user`.`first_name`,`user`.`last_name`,`user`.`contact_number`,`city`.`name` AS
`city_name` FROM `user` INNER JOIN `city` ON `user`.`city_id`=`city`.`id` WHERE
`user`.`first_name` LIKE '"+text+"%' OR `user`.`contact_number` LIKE '"+text+"%' ORDER BY
`user`.`id` ASC;");

        //DefaultTableModel dtm = new DefaultTableModel();

        DefaultTableModel dtm = (DefaultTableModel) jTable3.getModel();

        dtm.setRowCount(0);

        while (rs.next()) {

            Vector v = new Vector();

            v.add(rs.getString("id"));

            v.add(rs.getString("first_name"));

            v.add(rs.getString("last_name"));

            v.add(rs.getString("contact_number"));

            v.add(rs.getString("city_name"));

            dtm.addRow(v);

        }

    }
```

```
jTable3.setModel(dtm);

} catch (Exception e) {
    e.printStackTrace();
}
}
```

### **Search (text Field) –**

```
String text = jTextField13.getText();
loadcustomer(text);
```

### **Table Mouse Clicked-**

```
if (evt.getClickCount() == 1) {
    int r = jTable3.getSelectedRow();

    if (r == -1) {
        JOptionPane.showMessageDialog(this, "Please select a Supplier", "Warning",
JOptionPane.WARNING_MESSAGE);
    } else {

        String fname = jTable3.getValueAt(r, 1).toString();
```

```

String lname = jTable3.getValueAt(r, 2).toString();

String cno = jTable3.getValueAt(r, 3).toString();

String city = jTable3.getValueAt(r, 4).toString();


jTextField9.setText(fname);

jTextField10.setText(lname);

jTextField12.setText(cno);

jComboBox6.setSelectedItem(city);

}

}

```

#### **FirstName (Textfield) –**

```

private void jTextField9KeyTyped(java.awt.event.KeyEvent evt) {

    // TODO add your handling code here:

    char m = evt.getKeyChar();


    if(Character.isLetter(m) || Character.isISOControl(m)){

        jTextField9.setEditable(true);

    }else{

        jTextField9.setEditable(false);

    }

}

```

### **Last Name (TextFiel)**

```
private void jTextField10KeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
  
    char m = evt.getKeyChar();  
  
    if(Character.isLetter(m) || Character.isISOControl(m)){  
  
        jTextField10.setEditable(true);  
    }else{  
  
        jTextField10.setEditable(false);  
    }  
}
```

### **Contact No (TextField) –**

```
private void jTextField12KeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
  
    System.out.println("1");  
    String mobile = jTextField12.getText();  
    String text = mobile + evt.getKeyChar();
```



```
if (text.length() == 1) {  
    if (!text.equals("0")) {  
        evt.consume();  
    }  
} else if (text.length() == 2) {  
    if (!text.equals("07")) {  
        evt.consume();  
    }  
} else if (text.length() == 3) {  
    if (!Pattern.compile("07[01245678]").matcher(text).matches()) {  
        evt.consume();  
    }  
} else if (text.length() <= 10) {  
    if (!Pattern.compile("07[01245678][0-9]+").matcher(text).matches()) {  
        evt.consume();  
    }  
} else {  
    evt.consume();  
}  
}
```

### **City (ComboBox) -**

```
public void loadCities() {  
    try {  
        ResultSet rs = MySql.search("SELECT * FROM `city`");  
        Vector v = new Vector();  
        v.add("Select");  
        while (rs.next()) {  
            v.add(rs.getString("name"));  
        }  
        DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);  
        jComboBox6.setModel(dcm);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

### **ResetField –**

```
public void resetFields() {  
    jTextField9.setText("");  
    jTextField10.setText("");  
    jTextField12.setText("");  
    jComboBox6.setSelectedIndex(0);  
}
```

```
jTextField9.grabFocus();  
}
```

#### **Create Account Button –**

```
private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    String f_name = jTextField9.getText();  
    String l_name = jTextField10.getText();  
    String mobile = jTextField12.getText();  
    String city = jComboBox6.getSelectedItem().toString();  
  
    if (f_name.isEmpty()) {  
        JOptionPane.showMessageDialog(this, "Please Enter First Name", "warning",  
JOptionPane.WARNING_MESSAGE);  
    } else if (l_name.isEmpty()) {  
        JOptionPane.showMessageDialog(this, "Please Enter Last Name", "warning",  
JOptionPane.WARNING_MESSAGE);  
    } else if (mobile.isEmpty()) {  
        JOptionPane.showMessageDialog(this, "Please Enter Contact No", "warning",  
JOptionPane.WARNING_MESSAGE);  
    } else if (city.equals("Select")) {
```

```

        JOptionPane.showMessageDialog(this, "Please Select city", "warning",
JOptionPane.WARNING_MESSAGE);

    } else {

        try {

            ResultSet rs1 = MySql.search("SELECT `id` FROM `city` WHERE `name` = '" + city + "'");

            rs1.next();

            String city_id = rs1.getString("id");

            MySql.iud("INSERT INTO `user`(`first_name`,`last_name`,`contact_number`,`city_id`)
VALUES ('" + f_name + "','" + l_name + "','" + mobile + "','" + Integer.parseInt(city_id) + "');");

            resetFields();

            loadcustomer();

            JOptionPane.showMessageDialog(this, "New User Acoount Regidterd", "Success",
JOptionPane.INFORMATION_MESSAGE);

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}

```

## Update Button –

```
int r = jTable3.getSelectedRow();

String id = jTable3.getValueAt(r, 0).toString();

String f_Name = jTextField9.getText();
String l_Name = jTextField10.getText();
String cno = jTextField12.getText();
String city = jComboBox6.getSelectedItem().toString();

if (f_Name.isEmpty()) {
    JOptionPane.showMessageDialog(this, "Please Select a Row you want to update from the table", "warning", JOptionPane.WARNING_MESSAGE);
} else if (l_Name.isEmpty()) {
    JOptionPane.showMessageDialog(this, "Please Enter last Number", "warning", JOptionPane.WARNING_MESSAGE);
} else if (cno.isEmpty()) {
    JOptionPane.showMessageDialog(this, "Please Enter a Mobile Number", "warning", JOptionPane.WARNING_MESSAGE);
} else if (city.isEmpty()) {
    JOptionPane.showMessageDialog(this, "Please Select a City", "warning", JOptionPane.WARNING_MESSAGE);
} else {

    try {
```

```
ResultSet rs1 = MySql.search("SELECT `id` FROM `city` WHERE `name` = '" + city +
""");

rs1.next();

String city_id = rs1.getString("id");

MySql.iud("Update user set first_name='" + f_Name + "',last_name='" + l_Name +
",contact_number='" + cno + "',city_id='" + Integer.parseInt(city_id) + "' where id='" + id + "';");

resetFields();

loadcustomer();

JOptionPane.showMessageDialog(this, "User status updated", "Success",
JOptionPane.INFORMATION_MESSAGE);

jButton9.setEnabled(true);

} catch (Exception e) {

    e.printStackTrace();

}

}
```

## 2. Supplier Registration

The screenshot shows a web application for 'Jewelry SHOP'. The left sidebar contains navigation links: Customer Registration, Supplier Registration (active), Employee Registration, Branch Registration, Owner Registration, Add Jewellery, Stock, Reports, Exchanged Item, and Logout. The main content area is titled 'Supplier Registration' and features a search bar. Below the search bar is a table with 5 columns: ID, Name, Contact No, Email, and Company Branch. The table contains 11 rows of data. Below the table are input fields for Name, Contact No, Email, and Co-Branch (a dropdown menu), along with 'Add' and 'Update' buttons.

ID	Name	Contact No	Email	Company Branch
1	Rocchana Rathnayaka	0769815988	Rochana@gmail.com	colombo branch
2	Thewindu Munnaa	0776358954	Thewindu@gmail.com	colombo branch
3	Rocchana Rathnayaka	0769815988	Rochana@gmail.com	colombo branch
4	Rocchana Rathnayaka	0769815989	Rochana@gmail.com	colombo branch
5	Rocchana Rathnayaka	0769815988	Rochana@gmail.com	colombo branch
6	Tashila kumaraya	0713573654	tashilakumaraya@gmail.com	jewell kandy
7	dasun sanku	0783658923	dk12@gmail.com	jewell kandy
8	Mendis soisa	0786589562	soisa@gmail.com	colombo branch
9	amal perera	0782365589	amal@gmail.com	colombo branch
10	mawan	0753269854	mawan@gmail.com	trinco branch
11	poorna	0758969665	poorna@gmail.com	colombo branch

To load data into the table –

```
public void loadsupplier() {  
  
    try {  
  
        ResultSet rs = MySql.search("SELECT  
        `supplier`.`id`,`supplier`.`name`,`supplier`.`contact_number`,`supplier`.`email`,`company_branc  
        h`.`name` AS `branch_name` FROM `supplier` INNER JOIN `company_branch` ON  
        `Supplier`.`company_branch_id`=`company_branch`.`id`");  
  
        //DefaultTableModel dtm = new DefaultTableModel();  
  
        DefaultTableModel dtm = (DefaultTableModel) jTable4.getModel();
```

```
dtm.setRowCount(0);

while (rs.next()) {

    Vector v = new Vector();
    v.add(rs.getString("id"));
    v.add(rs.getString("name"));
    v.add(rs.getString("contact_number"));
    v.add(rs.getString("email"));
    v.add(rs.getString("branch_name"));

    dtm.addRow(v);
}

jTable4.setModel(dtm);

} catch (Exception e) {

    e.printStackTrace();
}

}
```



## Search –

```
public void loadsupplier(String text) {

    try {

        ResultSet rs = MySql.search("SELECT
`supplier`.`id`,`supplier`.`name`,`supplier`.`contact_number`,`supplier`.`email`,`company_branc
h`.`name` AS `branch_name` FROM `supplier` INNER JOIN `company_branch` ON
`Supplier`.`company_branch_id`=`company_branch`.`id` WHERE `supplier`.`name` LIKE
'" + text + "%' OR `supplier`.`contact_number` LIKE '" + text + "%' ORDER BY `supplier`.`id` ASC;");

        //DefaultTableModel dtm = new DefaultTableModel();

        DefaultTableModel dtm = (DefaultTableModel) jTable4.getModel();

        dtm.setRowCount(0);

        while (rs.next()) {

            Vector v = new Vector();

            v.add(rs.getString("id"));

            v.add(rs.getString("name"));

            v.add(rs.getString("contact_number"));

            v.add(rs.getString("email"));

            v.add(rs.getString("branch_name"));

            dtm.addRow(v);

        }

    }
```

```
jTable4.setModel(dtm);

} catch (Exception e) {
    e.printStackTrace();
}
}
```

### **Search (text Field) –**

```
private void jTextField17KeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:

    String text = jTextField17.getText();
    loadsupplier(text);
}
```

### **Table Mouse Clicked-**

```
private void jTable4MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:

    if(evt.getClickCount() == 1){
        int r = jTable4.getSelectedRow();
```

```
        if (r== -1) {

            JOptionPane.showMessageDialog(this, "Please select a
Supplier","Warning",JOptionPane.WARNING_MESSAGE);

        } else {

            String name = jTable4.getValueAt(r, 1).toString();

            String c_no = jTable4.getValueAt(r, 2).toString();

            String email = jTable4.getValueAt(r, 3).toString();

            String co_branch = jTable4.getValueAt(r, 4).toString();

            jTextField14.setText(name);

            jTextField15.setText(c_no);

            jTextField16.setText(email);

            jComboBox7.setSelectedItem(co_branch);

            mmd.setEnabled(false);

        }

    }

}
```

### **Name(textfield) –**

```
private void jTextField14KeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
  
    char m = evt.getKeyChar();  
  
    if(Character.isLetter(m) || Character.isWhitespace(m) || Character.isISOControl(m)){  
  
        jTextField14.setEditable(true);  
    }else{  
  
        jTextField14.setEditable(false);  
    }  
}
```

### **Contact No(textfield) –**

```
private void jTextField15KeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
  
    System.out.println("1");  
  
    String mobile = jTextField15.getText();  
  
    String text = mobile + evt.getKeyChar();
```

```
if (text.length() == 1) {  
    if (!text.equals("0")) {  
        evt.consume();  
    }  
} else if (text.length() == 2) {  
    if (!text.equals("07")) {  
        evt.consume();  
    }  
} else if (text.length() == 3) {  
    if (!Pattern.compile("07[01245678]").matcher(text).matches()) {  
        evt.consume();  
    }  
} else if (text.length() <= 10) {  
    if (!Pattern.compile("07[01245678][0-9]+").matcher(text).matches()) {  
        evt.consume();  
    }  
} else {  
    evt.consume();  
}  
}
```

### Company Branch(ComboBox) –

```
public void loadbranchess() {  
  
    try {  
  
        ResultSet rs = MySql.search("SELECT * FROM `company_branch`");  
  
        Vector v = new Vector();  
        v.add("Select");  
  
        while (rs.next()) {  
            v.add(rs.getString("name"));  
  
        }  
  
        DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);  
        jComboBox7.setModel(dcm);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

### **Reset Field –**

```
public void resetsupplier() {  
    jTextField14.setText("");  
    jTextField15.setText("");  
    jTextField16.setText("");  
    jComboBox7.setSelectedIndex(0);  
    jTextField14.grabFocus();  
}
```

### 3. Employee Registration

**08:06:33 PM**

**Employee Registration**

Search:

ID	Name	User Name	Password	NIC	Email	Contact_...	Address	Category	Emp Cate...
1	maduka	maduka4@	4336	9845631582V	Mduka@gm...	0728954363	no.32/kada...	Ring	recipion
2	kasun sank...	kasun4@	55869	20015896357	kasun@gm...	0778658823	No.04,Colo...	Necious	Workers
3	dilshan	dilshan3@	589745	20016589638	dilshan@g...	0778965585	no.03,kandy...	Ring	Workers
4	maduka	maduka4@	4336	9845631582V	Mduka@gm...	0728954363	no.32/kada...	Ring	recipion
5	kasu	kasu@	45569	56972585	kalakjsj@g...	0758996585	kandy road	Necious	Workers

Registration Form Fields:

- Name:
- User Name:
- Password:
- NIC:
- Email:
- Mobile\_No:
- Address:
- JCategory:
- ECategory:
- 
- 

To load data into the table –

```
public void loademployee() {
    try {
        ResultSet rs = MySql.search("SELECT
`employee`.`id`,`employee`.`name`,`employee`.`user_name`,`employee`.`password`,`employee`
.`NIC`,`employee`.`email`,`employee`.`Contact_No`,`employee`.`Address`,`category`.`name` AS
`category`,`emp_category`.`name` AS `emp_category` FROM `employee` INNER JOIN `category`
ON `employee`.`category_id`=`category`.`id` INNER JOIN `emp_category` ON
`employee`.`emp_category_id`=`emp_category`.`id`");

        //DefaultTableModel dtm = new DefaultTableModel();
```



```
DefaultTableModel dtm = (DefaultTableModel) jTable5.getModel();

dtm.setRowCount(0);

while (rs.next()) {

    Vector v = new Vector();

    v.add(rs.getString("id"));

    v.add(rs.getString("name"));

    v.add(rs.getString("user_name"));

    v.add(rs.getString("password"));

    v.add(rs.getString("NIC"));

    v.add(rs.getString("email"));

    v.add(rs.getString("Contact_No"));

    v.add(rs.getString("Address"));

    v.add(rs.getString("category"));

    v.add(rs.getString("emp_category"));

    dtm.addRow(v);

}

jTable5.setModel(dtm);

} catch (Exception e) {

    e.printStackTrace();

}

}
```

## Search –

```
public void loademployee(String text) {

    try {

        ResultSet rs = MySQL.search("SELECT
`employee`.`id`,`employee`.`name`,`employee`.`user_name`,`employee`.`password`,`employee`
`.`NIC`,`employee`.`email`,`employee`.`Contact_No`,`employee`.`Address`,`category`.`name` AS
`category`,`emp_category`.`name` AS `emp_category` FROM `employee` INNER JOIN `category`
ON `employee`.`category_id`=`category`.`id` INNER JOIN `emp_category` ON
`employee`.`emp_category_id`=`emp_category`.`id` WHERE `employee`.`name` LIKE
'" + text + "%' OR `employee`.`Contact_No` LIKE '" + text + "%' ORDER BY `employee`.`id` ASC;;");

        //DefaultTableModel dtm = new DefaultTableModel();

        DefaultTableModel dtm = (DefaultTableModel) jTable5.getModel();

        dtm.setRowCount(0);

        while (rs.next()) {

            Vector v = new Vector();

            v.add(rs.getString("id"));
            v.add(rs.getString("name"));
            v.add(rs.getString("user_name"));
            v.add(rs.getString("password"));
            v.add(rs.getString("NIC"));
            v.add(rs.getString("email"));
            v.add(rs.getString("Contact_No"));
            v.add(rs.getString("Address"));
            v.add(rs.getString("category"));
            v.add(rs.getString("emp_category"));
```

```
        dtm.addRow(v);

    }

    jTable5.setModel(dtm);

} catch (Exception e) {
    e.printStackTrace();
}
}
```

#### **Search (text Field) –**

```
private void jTextField24KeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    String text = jTextField24.getText();
    loademployee(text);
}
```

#### **Table Mouse Clicked-**

```
private void jTable5MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:

    if(evt.getClickCount() == 1){
        int r = jTable5.getSelectedRow();
```

```
if (r== -1) {  
    JOptionPane.showMessageDialog(this, "Please select a  
Jewellery", "Warning", JOptionPane.WARNING_MESSAGE);  
} else {  
  
    String name = jTable5.getValueAt(r, 1).toString();  
    String u_name = jTable5.getValueAt(r, 2).toString();  
    String password = jTable5.getValueAt(r, 3).toString();  
    String nic = jTable5.getValueAt(r, 4).toString();  
    String email = jTable5.getValueAt(r, 5).toString();  
    String mobile_no = jTable5.getValueAt(r, 6).toString();  
    String address = jTable5.getValueAt(r, 7).toString();  
    String j_category = jTable5.getValueAt(r, 8).toString();  
    String e_category = jTable5.getValueAt(r, 9).toString();  
  
    jTextField18.setText(name);  
    jTextField19.setText(u_name);  
    jPasswordField2.setText(password);  
    jTextField20.setText(nic);  
    jTextField21.setText(email);  
    jTextField22.setText(mobile_no);  
    jTextField23.setText(address);  
    jComboBox9.setSelectedItem(j_category);  
    jComboBox8.setSelectedItem(e_category);  
    addbutton.setEnabled(false);
```

```
}
```

```
}
```

```
}
```

### **Name (Textfield) –**

```
private void jTextField18KeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
  
    char m = evt.getKeyChar();  
  
    if(Character.isLetter(m) || Character.isWhitespace(m) || Character.isISOControl(m)){  
  
        jTextField1.setEditable(true);  
    }else{  
  
        jTextField1.setEditable(false);  
    }  
}
```

### Contact No (TextField) –

```
private void jTextField22KeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
  
    System.out.println("1");  
    String mobile = jTextField22.getText();  
    String text = mobile + evt.getKeyChar();  
  
    if (text.length() == 1) {  
        if (!text.equals("0")) {  
            evt.consume();  
        }  
    } else if (text.length() == 2) {  
        if (!text.equals("07")) {  
            evt.consume();  
        }  
    } else if (text.length() == 3) {  
        if (!Pattern.compile("07[01245678]").matcher(text).matches()) {  
            evt.consume();  
        }  
    } else if (text.length() <= 10) {  
        if (!Pattern.compile("07[01245678][0-9]+").matcher(text).matches()) {  
            evt.consume();  
        }  
    }  
}
```

```
    }  
  } else {  
    evt.consume();  
  }  
}
```

### **JCategory (ComboBox) –**

```
public void loadjcategory() {  
  
    try {  
        ResultSet rs = MySql.search("SELECT * FROM `category`");  
  
        Vector v = new Vector();  
        v.add("Select");  
  
        while (rs.next()) {  
            v.add(rs.getString("name"));  
        }  
  
        DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);  
        jComboBox9.setModel(dcm);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

```
}  
}
```

### **ECategory (ComboBox) –**

```
public void loadempcategory() {  
  
    try {  
        ResultSet rs = MySql.search("SELECT * FROM `emp_category`");  
  
        Vector v = new Vector();  
        v.add("Select");  
  
        while (rs.next()) {  
            v.add(rs.getString("name"));  
        }  
  
        DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);  
        jComboBox8.setModel(dcm);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
}
```



### **ResetField –**

```
public void resetemployee() {  
    jTextField18.setText("");  
    jTextField19.setText("");  
    jPasswordField2.setText("");  
    jTextField20.setText("");  
    jTextField21.setText("");  
    jTextField22.setText("");  
    jTextField23.setText("");  
    jComboBox9.setSelectedIndex(0);  
    jComboBox8.setSelectedIndex(0);  
    jTextField18.grabFocus();  
}
```

### **Add button –**

```
private void addbuttonActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    String ename = jTextField18.getText();  
    String u_name = jTextField19.getText();
```

```
String password = jPasswordField2.getText();

String nic = jTextField20.getText();

String Email = jTextField21.getText();

String cont_no = jTextField22.getText();

String address = jTextField23.getText();

String jcate = jComboBox9.getSelectedItem().toString();

String ecat = jComboBox8.getSelectedItem().toString();


if (ename.isEmpty()) {

    JOptionPane.showMessageDialog(this, "Please Enter Name", "warning",
JOptionPane.WARNING_MESSAGE);

} else if (u_name.isEmpty()) {

    JOptionPane.showMessageDialog(this, "Please Enter User Name", "warning",
JOptionPane.WARNING_MESSAGE);

} else if (password.isEmpty()) {

    JOptionPane.showMessageDialog(this, "Please Enter Password", "warning",
JOptionPane.WARNING_MESSAGE);

} else if (nic.isEmpty()) {

    JOptionPane.showMessageDialog(this, "Please Enter NIC", "warning",
JOptionPane.WARNING_MESSAGE);

} else if (Email.isEmpty()) {

    JOptionPane.showMessageDialog(this, "Please Enter Email Address", "warning",
JOptionPane.WARNING_MESSAGE);

} else if (cont_no.isEmpty()) {

    JOptionPane.showMessageDialog(this, "Please Enter Contact Number", "warning",
JOptionPane.WARNING_MESSAGE);
```

```

    } else if (address.isEmpty()) {

        JOptionPane.showMessageDialog(this, "Please Select jeweller Category", "warning",
JOptionPane.WARNING_MESSAGE);

    } else if (jcate.equals("Select")) {

        JOptionPane.showMessageDialog(this, "Please Select type", "warning",
JOptionPane.WARNING_MESSAGE);

    } else if (ecat.equals("Select")) {

        JOptionPane.showMessageDialog(this, "Please Select Employee Category", "warning",
JOptionPane.WARNING_MESSAGE);

    } else {

        try {

            ResultSet rs1 = MySql.search("SELECT `id` FROM `category` WHERE `name` = '" + jcate
+ "'");

            rs1.next();

            ResultSet rs2 = MySql.search("SELECT `id` FROM `emp_category` WHERE `name` = '" +
ecat + "'");

            rs2.next();

            String j_id = rs1.getString("id");

            String e_id = rs2.getString("id");

            MySql.iud("INSERT INTO
`employee`(`name`,`user_name`,`password`,`NIC`,`email`,`Contact_No`,`Address`,`category_id`
`,`emp_category_id`) VALUES ('" + ename + "','" + u_name + "','" + password + "','" + nic + "','" +

```

```
Emaill + "," + cont_no + "," + address + "," + Integer.parseInt(j_id) + "," +  
Integer.parseInt(e_id) + ");");
```

```
resetemployee();
```

```
loademployee();
```

```
JOptionPane.showMessageDialog(this, "New Employee Regidterd", "Success",  
JOptionPane.INFORMATION_MESSAGE);
```

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

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

```
    }
```

```
}
```

```
}
```

### **Update button –**

```
private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
```

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

```
    int r = jTable5.getSelectedRow();
```

```
    String id = jTable5.getValueAt(r, 0).toString();
```

```
    String e_Name = jTextField18.getText();
```

```
    String u_name = jTextField19.getText();
```

```
    String password = jPasswordField2.getText();
```

```
String nic = jTextField20.getText();

String email = jTextField21.getText();

String m_no = jTextField22.getText();

String addr = jTextField23.getText();

String j_cate = jComboBox9.getSelectedItem().toString();

String e_cate = jComboBox8.getSelectedItem().toString();


if(e_Name.isEmpty()){

    JOptionPane.showMessageDialog(this, "Please Select a Row you want to update from
the table", "warning", JOptionPane.WARNING_MESSAGE);

}else if(u_name.isEmpty()){

    JOptionPane.showMessageDialog(this, "Please Enter User Name", "warning",
JOptionPane.WARNING_MESSAGE);

}else if(password.isEmpty()){

    JOptionPane.showMessageDialog(this, "Please Enter Password", "warning",
JOptionPane.WARNING_MESSAGE);

}else if(nic.isEmpty()){

    JOptionPane.showMessageDialog(this, "Please Enter NIC", "warning",
JOptionPane.WARNING_MESSAGE);

}else if(email.isEmpty()){

    JOptionPane.showMessageDialog(this, "Please Enter Email", "warning",
JOptionPane.WARNING_MESSAGE);

}else if(m_no.isEmpty()){

    JOptionPane.showMessageDialog(this, "Please Enter Mobile Number", "warning",
JOptionPane.WARNING_MESSAGE);

}else if(addr.isEmpty()){
```

```

        JOptionPane.showMessageDialog(this, "Please Enter Address", "warning",
JOptionPane.WARNING_MESSAGE);

        }else if(j_cate.isEmpty()){

            JOptionPane.showMessageDialog(this, "Please Select a Jewellery Category", "warning",
JOptionPane.WARNING_MESSAGE);

        }else if(e_cate.isEmpty()){

            JOptionPane.showMessageDialog(this, "Please Select a employee Category", "warning",
JOptionPane.WARNING_MESSAGE);

        }else{

            try{

                ResultSet rs1 = MySql.search("SELECT `id` FROM `category` WHERE `name` = '" + j_cate
+ "'");

                rs1.next();

                ResultSet rs2 = MySql.search("SELECT `id` FROM `emp_category` WHERE `name` = '" +
e_cate + "'");

                rs2.next();

                String cate_id = rs1.getString("id");

                String emp_id = rs2.getString("id");

                MySql.iud("Update employee set
name='"+e_Name+"',user_name='"+u_name+"',password='"+password+"',NIC='"+nic+"',email=
 '"+emaill+"',Contact_No='"+m_no+"',Address='"+addr+"',category_id='"+
Integer.parseInt(cate_id) + "',emp_category_id='"+ Integer.parseInt(emp_id) + "' where
id='"+id+"';");

```

```
resetemployee();
```

```
loademployee();
```

```
        JOptionPane.showMessageDialog(this, "User status updated",  
"Success",JOptionPane.INFORMATION_MESSAGE);
```

```
        addbutton.setEnabled(true);
```

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

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

```
    }
```

```
}
```

```
}
```

#### 4. Branch Registration -

The screenshot shows a web application for 'Jewelry Shop'. The left sidebar contains navigation links: Customer Registration, Supplier Registration, Employee Registration, Branch Registration (highlighted), Owner Registration, Add Jewellery, Stock, Reports, Exchanged Item, and Logout. The main content area is titled 'Branch Registration' and features a form with fields for Contact No., Company Name (dropdown), Branch Name, and Address (dropdown). There are buttons for 'Add Company', 'Add New Branch', and 'Add Branch Address'. Below the form is a table listing existing branches.

Branch ID	Branch Contact No	Branch Name	Company	Address
5	0753568954	ampara	Kingki (Pvt) Ltd	25,Ward Terrace,
6	0785555222	kongo	Kingki (Pvt) Ltd	25,Ward Terrace,
3	0769990235	trinco branch	Progress Factors Ltd	Kassapa Road,Colombo
2	0756984523	colombo branch	Gemsmart International (Pvt.)...	10,wijeyamangalarama Road,
1	0786589262	jewell kandy	Jewel of A Million	25,Ward Terrace,
4	0783358758	kurunegala branch	Shaminda	10,wijeyamangalarama Road,

To load data into the table –

```
public void loadbranches() {
```

```
    try {
```

```
        ResultSet rs = MySql.search("SELECT  
`company_branch`.`id`,`company_branch`.`branch_contact_number`,`company_branch`.`name`  
`,`company`.`name` AS `company`,`company_branch_address`.`Address` AS `Address` FROM  
`company_branch` INNER JOIN `company` ON `company_branch`.`company_id`=`company`.`id`  
INNER JOIN `company_branch_address` ON  
`company_branch`.`company_branch_address_id`=`company_branch_address`.`id`");
```



```
//DefaultTableModel dtm = new DefaultTableModel();

DefaultTableModel dtm = (DefaultTableModel) jTable7.getModel();

dtm.setRowCount(0);


while (rs.next()) {

    Vector v = new Vector();

    v.add(rs.getString("id"));

    v.add(rs.getString("branch_contact_number"));

    v.add(rs.getString("name"));

    v.add(rs.getString("company"));

    v.add(rs.getString("Address"));

    dtm.addRow(v);

}

jTable7.setModel(dtm);


} catch (Exception e) {

    e.printStackTrace();

}

}
```

### Contact No (TextField) –

```
private void jTextField27KeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
  
    System.out.println("1");  
    String mobile = jTextField27.getText();  
    String text = mobile + evt.getKeyChar();  
  
    if (text.length() == 1) {  
        if (!text.equals("0")) {  
            evt.consume();  
        }  
    } else if (text.length() == 2) {  
        if (!text.equals("07")) {  
            evt.consume();  
        }  
    } else if (text.length() == 3) {  
        if (!Pattern.compile("07[01245678]").matcher(text).matches()) {  
            evt.consume();  
        }  
    } else if (text.length() <= 10) {  
        if (!Pattern.compile("07[01245678][0-9]+").matcher(text).matches()) {  
            evt.consume();  
        }  
    }  
}
```

```
    }  
  } else {  
    evt.consume();  
  }  
}
```

### **Branch Name (Textfield) –**

```
private void jTextField28KeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
  
    char m = evt.getKeyChar();  
  
    if(Character.isLetter(m) || Character.isWhitespace(m) || Character.isISOControl(m)){  
  
        jTextField28.setEditable(true);  
    }else{  
  
        jTextField28.setEditable(false);  
    }  
}
```

### Company Name (ComboBox) –

```
public void loadcompany() {  
  
    try {  
  
        ResultSet rs = MySql.search("SELECT * FROM `company`");  
  
        Vector v = new Vector();  
        v.add("Select");  
  
        while (rs.next()) {  
            v.add(rs.getString("name"));  
  
        }  
  
        DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);  
        jComboBox12.setModel(dcm);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
}
```

### Address (ComboBox) –

```
public void loadaddress() {  
  
    try {  
  
        ResultSet rs = MySql.search("SELECT * FROM `company_branch_address`");  
  
        Vector v = new Vector();  
        v.add("Select");  
  
        while (rs.next()) {  
            v.add(rs.getString("Address"));  
  
        }  
  
        DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);  
        jComboBox13.setModel(dcm);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
}
```

### Add New Branch(Butoon) –

```
private void jButton19ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    // TODO add your handling code here:  
  
    String cno = jTextField27.getText();  
    String b_name = jTextField28.getText();  
    String com_name = jComboBox12.getSelectedItem().toString();  
    String addre = jComboBox13.getSelectedItem().toString();  
  
    if (cno.isEmpty()) {  
        JOptionPane.showMessageDialog(this, "Please Enter Contact Number", "warning",  
JOptionPane.WARNING_MESSAGE);  
    } else if (b_name.isEmpty()) {  
        JOptionPane.showMessageDialog(this, "Please Enter Branch Name", "warning",  
JOptionPane.WARNING_MESSAGE);  
    } else if (com_name.isEmpty()) {  
        JOptionPane.showMessageDialog(this, "Please Select Company Name", "warning",  
JOptionPane.WARNING_MESSAGE);  
    } else if (addre.isEmpty()) {  
        JOptionPane.showMessageDialog(this, "Please Select Address", "warning",  
JOptionPane.WARNING_MESSAGE);  
    } else {  
  
        try {
```

```
ResultSet rs1 = MySql.search("SELECT `id` FROM `company` WHERE `name` = '" +  
com_name + "'");
```

```
rs1.next();
```

```
ResultSet rs2 = MySql.search("SELECT `id` FROM `company_branch_address` WHERE  
`Address` = '" + addre + "'");
```

```
rs2.next();
```

```
String com_id = rs1.getString("id");
```

```
String addre_id = rs2.getString("id");
```

```
MySql.iud("INSERT INTO  
`company_branch`(`branch_contact_number`,`name`,`company_id`,`company_branch_address  
_id`) VALUES ('" + cno + "','" + b_name + "','" + Integer.parseInt(com_id) + "','" +  
Integer.parseInt(addre_id) + "')");
```

```
jTextField27.setText("");
```

```
jTextField28.setText("");
```

```
jComboBox12.setSelectedIndex(0);
```

```
jComboBox13.setSelectedIndex(0);
```

```
jTextField27.grabFocus();
```

```
loadbranches();
```

```
JOptionPane.showMessageDialog(this, "New User Acoount Regidterd", "Success",  
JOptionPane.INFORMATION_MESSAGE);
```

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

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

```
    }
```

```
}
```

```
}
```



## 5. Owner Registration

**Jewelry SHOP**

08:06:42 PM

**Owner Registration**

ID	Name	User Name	Email	Passcode	Emp_Category
1	Mawantha	Mawantha5@	mawanthadeeg...	5336	admin
2	kasun	kasun4@	kasunsankalpa6...	4522	reciption
3	nadun	nadun3@	nadun@gmail.c...	4556	reciption
4	sarath	sarath8@	sarath@gmail.c...	889624	reciption

Name:  User Name:  Emp Category:

Email:  Passcode:

To load data into the table –

```
public void loadowner() {  
  
    try {  
  
        ResultSet rs = MySql.search("SELECT  
`owner`.`id`,`owner`.`name`,`owner`.`user_name`,`owner`.`email`,`owner`.`passcode`,`emp_ca  
tegory`.`name` AS `category` FROM `owner` INNER JOIN `emp_category` ON  
`owner`.`emp_category_id`=`emp_category`.`id`");  
  
        //DefaultTableModel dtm = new DefaultTableModel();
```

```
DefaultTableModel dtm = (DefaultTableModel) jTable8.getModel();

dtm.setRowCount(0);

while (rs.next()) {

    Vector v = new Vector();

    v.add(rs.getString("id"));

    v.add(rs.getString("name"));

    v.add(rs.getString("user_name"));

    v.add(rs.getString("email"));

    v.add(rs.getString("passcode"));

    v.add(rs.getString("category"));

    dtm.addRow(v);

}

jTable8.setModel(dtm);

} catch (Exception e) {

    e.printStackTrace();

}

}
```

### Employee Category (ComboBox) –

```
public void loadempcategory() {  
  
    try {  
  
        ResultSet rs = MySql.search("SELECT * FROM `emp_category`");  
  
        Vector v = new Vector();  
        v.add("Select");  
  
        while (rs.next()) {  
            v.add(rs.getString("name"));  
  
        }  
  
        DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);  
        jComboBox14.setModel(dcm);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
}
```

### **ResetField –**

```
public void resetowner() {  
    jTextField29.setText("");  
    jTextField30.setText("");  
    jTextField31.setText("");  
    jPasswordField1.setText("");  
    jComboBox14.setSelectedIndex(0);  
    jTextField29.grabFocus();  
}
```

### **Mouse Clicked –**

```
private void jTable8MouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
  
    if(evt.getClickCount() == 1){  
        int r = jTable8.getSelectedRow();  
  
        if (r== -1) {  
            JOptionPane.showMessageDialog(this, "Please select a  
Jewellery","Warning",JOptionPane.WARNING_MESSAGE);  
        } else {
```

```
String name = jTable8.getValueAt(r, 1).toString();  
String u_name = jTable8.getValueAt(r, 2).toString();  
String email = jTable8.getValueAt(r, 3).toString();  
String passcode = jTable8.getValueAt(r, 4).toString();  
String e_category = jTable8.getValueAt(r, 5).toString();
```

```
jTextField29.setText(name);  
jTextField30.setText(u_name);  
jTextField31.setText(email);  
jPasswordField1.setText(passcode);  
jComboBox14.setSelectedItem(e_category);  
jButton27.setEnabled(false);
```

```
}
```

```
}
```

```
}
```

### Add button –

```
private void jButton27ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    String name = jTextField29.getText();  
    String u_name = jTextField30.getText();  
    String Email = jTextField31.getText();  
    String passcode = jPasswordField1.getText();  
    String ecat = jComboBox14.getSelectedItem().toString();  
  
    if (name.isEmpty()) {  
        JOptionPane.showMessageDialog(this, "Please Enter Name", "warning",  
JOptionPane.WARNING_MESSAGE);  
    } else if (u_name.isEmpty()) {  
        JOptionPane.showMessageDialog(this, "Please Enter User Name", "warning",  
JOptionPane.WARNING_MESSAGE);  
    } else if (passcode.isEmpty()) {  
        JOptionPane.showMessageDialog(this, "Please Enter Password", "warning",  
JOptionPane.WARNING_MESSAGE);  
    } else if (ecat.equals("Select")) {  
        JOptionPane.showMessageDialog(this, "Please Select Employee Category", "warning",  
JOptionPane.WARNING_MESSAGE);  
    } else {  
  
        try {
```

```
ResultSet rs1 = MySql.search("SELECT `id` FROM `emp_category` WHERE `name` = '" +  
ecat + "'");
```

```
rs1.next();
```

```
String e_id = rs1.getString("id");
```

```
MySql.iud("INSERT INTO  
`owner`(`name`,`user_name`,`email`,`passcode`,`emp_category_id`) VALUES ('" + name + "','" +  
u_name + "','" + Email + "','" + passcode + "','" + Integer.parseInt(e_id) + "')");
```

```
resetowner();
```

```
loadowner();
```

```
JOptionPane.showMessageDialog(this, "New Owner Registered", "Success",  
JOptionPane.INFORMATION_MESSAGE);
```

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

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

```
}
```

```
}
```

```
}
```

## Update button -

```
private void jButton28ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    int r = jTable8.getSelectedRow();  
  
    String id = jTable8.getValueAt(r, 0).toString();  
    String Name = jTextField29.getText();  
    String u_name = jTextField30.getText();  
    String Email = jTextField31.getText();  
    String password = jPasswordField1.getText();  
    String e_cate = jComboBox14.getSelectedItem().toString();  
  
    if(Name.isEmpty()){  
        JOptionPane.showMessageDialog(this, "Please Select a Row you want to update from  
the table", "warning", JOptionPane.WARNING_MESSAGE);  
    }else if(u_name.isEmpty()){  
        JOptionPane.showMessageDialog(this, "Please Enter User Name", "warning",  
JOptionPane.WARNING_MESSAGE);  
    }else if(Email.isEmpty()){  
        JOptionPane.showMessageDialog(this, "Please Enter Email", "warning",  
JOptionPane.WARNING_MESSAGE);  
    }else if(password.isEmpty()){
```



```

        JOptionPane.showMessageDialog(this, "Please Enter Password", "warning",
JOptionPane.WARNING_MESSAGE);

        }else if(e_cate.isEmpty()){

            JOptionPane.showMessageDialog(this, "Please Select a employee Category", "warning",
JOptionPane.WARNING_MESSAGE);

        }else{

            try{

                ResultSet rs1 = MySql.search("SELECT `id` FROM `emp_category` WHERE `name` = '" +
e_cate + "'");

                rs1.next();

                String emp_id = rs1.getString("id");

                MySql.iud("Update owner set
name='"+Name+"',user_name='"+u_name+"',email='"+Email+"',passcode='"+password+"',emp
_category_id='"+ Integer.parseInt(emp_id) +"' where id='"+id+"';");

                resetowner();

                loadowner();

                JOptionPane.showMessageDialog(this, "Owner status updated",
"Success",JOptionPane.INFORMATION_MESSAGE);

                jButton27.setEnabled(true);

            }catch(Exception e){

```

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

```
}
```

```
}
```

```
}
```

## 6. Add Jewelry

The screenshot shows a web application titled 'Jewelry SHOP'. The sidebar on the left contains the following menu items: Customer Registration, Supplier Registration, Employee Registration, Branch Registration, Owner Registration, Add Jewelry (highlighted), Stock, Reports, Exchanged Item, and Logout. The top header displays the time '08:06:47 PM'. The main content area is titled 'Add Jewelry' and features a search bar. Below the search bar is a table with the following data:

ID	Name	Caret No	Supplier	Category	Material
1	golden nec	06	Rocchana Rathn...	Ring	Silver
2	Jewelary Empire	03	Thewindu Munn...	Ring	Silver
5	golden egal	25	Rocchana Rathn...	Neclous	Silver
3	The Gold Lodge	10	Tashila kumaraya	Neclous	Gold
4	mmd	24	Rocchana Rathn...	Neclous	Gold
7	GOLD RGAL	24	dasun sank	Neclous	Gold
8	silver nec	24	Rocchana Rathn...	Neclous	Gold
9	golden necs	24	dasun sank	Ring	Gold
6	golden siver	24	Rocchana Rathn...	Neclous	Platinum

Below the table, there are three buttons: 'Add Jiwellery', 'Add Category', and 'Add Material'. The 'Add Jiwellery' button is highlighted.

To load data into the table –

```
public void loadjewellery() {
```

```
    try {
```

```
        ResultSet rs = MySql.search("SELECT  
`jewellery`.`id`,`jewellery`.`name`,`jewellery`.`caret_no`,`supplier`.`name` AS  
`supplier_name`,`category`.`name` AS `category`,`material`.`name` AS `material` FROM  
`jewellery` INNER JOIN `supplier` ON `jewellery`.`supplier_id`=`supplier`.`id` INNER JOIN `category`  
ON `jewellery`.`category_id`=`category`.`id` INNER JOIN `material` ON  
`jewellery`.`material_id`=`material`.`id`");
```

```
//DefaultTableModel dtm = new DefaultTableModel();

DefaultTableModel dtm = (DefaultTableModel) jTable6.getModel();

dtm.setRowCount(0);


while (rs.next()) {


    Vector v = new Vector();

    v.add(rs.getString("id"));

    v.add(rs.getString("name"));

    v.add(rs.getString("caret_no"));

    v.add(rs.getString("supplier_name"));

    v.add(rs.getString("category"));

    v.add(rs.getString("material"));

    dtm.addRow(v);

}


jTable6.setModel(dtm);


} catch (Exception e) {

    e.printStackTrace();

}

}
```

## Search –

```
public void loadjewellery(String text) {

    try {

        ResultSet rs = MySQL.search("SELECT
`jewellery`.`id`,`jewellery`.`name`,`jewellery`.`caret_no`,`supplier`.`name` AS
`supplier_name`,`category`.`name` AS `category`,`material`.`name` AS `material` FROM
`jewellery` INNER JOIN `supplier` ON `jewellery`.`supplier_id`=`supplier`.`id` INNER JOIN `category`
ON `jewellery`.`category_id`=`category`.`id` INNER JOIN `material` ON
`jewellery`.`material_id`=`material`.`id` WHERE `jewellery`.`name` LIKE '" + text + "%' OR
`jewellery`.`caret_no` LIKE '" + text + "%' ORDER BY `jewellery`.`id` ASC;");

        //DefaultTableModel dtm = new DefaultTableModel();

        DefaultTableModel dtm = (DefaultTableModel) jTable6.getModel();

        dtm.setRowCount(0);

        while (rs.next()) {

            Vector v = new Vector();

            v.add(rs.getString("id"));

            v.add(rs.getString("name"));

            v.add(rs.getString("caret_no"));

            v.add(rs.getString("supplier_name"));

            v.add(rs.getString("category"));

            v.add(rs.getString("material"));

            dtm.addRow(v);

        }

    }

}
```

```
        jTable6.setModel(dtm);

    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

### **Search (text Field) –**

```
private void jTextField26KeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:

    String text = jTextField26.getText();
    loadjewellery(text);
}
```

### **Table Mouse Clicked-**

```
private void jTable6MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:

    //    try {
        if (evt.getClickCount() == 1) {
            int r = jTable6.getSelectedRow();

            if (r == -1) {
```

```
JOptionPane.showMessageDialog(this, "Please select a Jewellery", "Warning",  
JOptionPane.WARNING_MESSAGE);
```

```
    } else {
```

```
        String name = jTable6.getValueAt(r, 1).toString();
```

```
        String car_no = jTable6.getValueAt(r, 2).toString();
```

```
        String supplier = jTable6.getValueAt(r, 3).toString();
```

```
        String j_category = jTable6.getValueAt(r, 4).toString();
```

```
        String material = jTable6.getValueAt(r, 5).toString();
```

```
        jTextField11.setText(name);
```

```
        jTextField25.setText(car_no);
```

```
        jComboBox10.setSelectedItem(supplier);
```

```
        jComboBox11.setSelectedItem(j_category);
```

```
        jComboBox5.setSelectedItem(material);
```

```
        jButton1.setEnabled(false);
```

```
    }
```

```
}
```

```
}
```

### **Jwellery Name (Textfield) –**

```
private void jTextField11KeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
  
    char m = evt.getKeyChar();  
  
    if(Character.isLetter(m) || Character.isWhitespace(m) || Character.isISOControl(m)){  
  
        jTextField11.setEditable(true);  
    }else{  
  
        jTextField11.setEditable(false);  
    }  
  
}
```

### **Category (ComboBox) –**

```
public void loadcategory() {  
  
    try {
```



```
ResultSet rs = MySql.search("SELECT * FROM `category`");
```

```
Vector v = new Vector();
```

```
v.add("Select");
```

```
while (rs.next()) {
```

```
    v.add(rs.getString("name"));
```

```
}
```

```
DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);
```

```
jComboBox11.setModel(dcm);
```

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

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

```
}
```

```
}
```

### **Materials (ComboBox) -**

```
public void loadmaterial() {  
  
    try {  
  
        ResultSet rs = MySql.search("SELECT * FROM `material`");  
  
        Vector v = new Vector();  
        v.add("Select");  
  
        while (rs.next()) {  
            v.add(rs.getString("name"));  
        }  
  
        DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);  
        jComboBox5.setModel(dcm);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
}
```

### **ResetField –**

```
public void resetaddjewellery() {  
    jTextField11.setText("");  
    jTextField25.setText("");  
    jComboBox10.setSelectedIndex(0);  
    jComboBox11.setSelectedIndex(0);  
    jComboBox5.setSelectedIndex(0);  
    jTextField11.grabFocus();  
}
```

### **Add Jewellery button –**

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    String name = jTextField11.getText();  
    String c_no = jTextField25.getText();  
    String sup = jComboBox10.getSelectedItem().toString();  
    String cat = jComboBox11.getSelectedItem().toString();  
    String mat = jComboBox5.getSelectedItem().toString();  
  
    if (name.isEmpty()) {  
        JOptionPane.showMessageDialog(this, "Please Enter Name", "warning",  
JOptionPane.WARNING_MESSAGE);  
    } else if (c_no.isEmpty()) {
```

```

JOptionPane.showMessageDialog(this, "Please Enter Caret No", "warning",
JOptionPane.WARNING_MESSAGE);

    } else if (sup.isEmpty()) {

        JOptionPane.showMessageDialog(this, "Please Select Supplier", "warning",
JOptionPane.WARNING_MESSAGE);

    } else if (cat.isEmpty()) {

        JOptionPane.showMessageDialog(this, "Please Select Category", "warning",
JOptionPane.WARNING_MESSAGE);

    } else if (mat.isEmpty()) {

        JOptionPane.showMessageDialog(this, "Please Select Material ", "warning",
JOptionPane.WARNING_MESSAGE);

    } else {

        try {

            ResultSet rs1 = MySql.search("SELECT `id` FROM `supplier` WHERE `name` = '" + sup +
        """);

            rs1.next();

            ResultSet rs2 = MySql.search("SELECT `id` FROM `category` WHERE `name` = '" + cat +
        """);

            rs2.next();

            ResultSet rs3 = MySql.search("SELECT `id` FROM `material` WHERE `name` = '" + mat +
        """);

            rs3.next();

            String sup_id = rs1.getString("id");

            String cat_id = rs2.getString("id");

```

```
String mat_id = rs3.getString("id");
```

```
MySql.iud("INSERT INTO  
`jewellery`(`name`,`caret_no`,`supplier_id`,`category_id`,`material_id`) VALUES ('" + name + "','"  
+ c_no + "','" + Integer.parseInt(sup_id) + "','" + Integer.parseInt(cat_id) + "','" +  
Integer.parseInt(mat_id) + "');");
```

```
resetaddjewellery();
```

```
loadjewellery();
```

```
JOptionPane.showMessageDialog(this, "New Jewellery added", "Success",  
JOptionPane.INFORMATION_MESSAGE);
```

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

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

```
    }
```

```
}
```

```
}
```

### **Update button –**

```
private void jButton16ActionPerformed(java.awt.event.ActionEvent evt) {
```

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

```
    int r = jTable6.getSelectedRow();
```

```
    String id = jTable6.getValueAt(r, 0).toString();
```

```

String J_Name = jTextField11.getText();
String Car_no = jTextField25.getText();
String supplier = jComboBox10.getSelectedItem().toString();
String j_category = jComboBox11.getSelectedItem().toString();
String material = jComboBox5.getSelectedItem().toString();

if (J_Name.isEmpty()) {
    JOptionPane.showMessageDialog(this, "Please Select a Row you want to update from
the table", "warning", JOptionPane.WARNING_MESSAGE);
} else if (Car_no.isEmpty()) {
    JOptionPane.showMessageDialog(this, "Please Enter Carret Number", "warning",
JOptionPane.WARNING_MESSAGE);
} else if (supplier.isEmpty()) {
    JOptionPane.showMessageDialog(this, "Please Select a Supplier", "warning",
JOptionPane.WARNING_MESSAGE);
} else if (j_category.isEmpty()) {
    JOptionPane.showMessageDialog(this, "Please Select a Jewellery Category", "warning",
JOptionPane.WARNING_MESSAGE);
} else if (material.isEmpty()) {
    JOptionPane.showMessageDialog(this, "Please Select a Jewellery Material", "warning",
JOptionPane.WARNING_MESSAGE);
} else {

    try {

        ResultSet rs1 = MySql.search("SELECT `id` FROM `supplier` WHERE `name` = '" +
supplier + "'");
        rs1.next();

```

```
ResultSet rs2 = MySql.search("SELECT `id` FROM `category` WHERE `name` = '" +  
j_category + "'");
```

```
rs2.next();
```

```
ResultSet rs3 = MySql.search("SELECT `id` FROM `material` WHERE `name` = '" +  
material + "'");
```

```
rs3.next();
```

```
String sup_id = rs1.getString("id");
```

```
String cat_id = rs2.getString("id");
```

```
String mat_id = rs3.getString("id");
```

```
MySql.iud("Update jewellery set name='" + J_Name + "',carat_no='" + Car_no +  
"',supplier_id='" + Integer.parseInt(sup_id) + "',category_id='" + Integer.parseInt(cat_id) +  
"',material_id='" + Integer.parseInt(mat_id) + "' where id='" + id + "'");
```

```
resetaddjewellery();
```

```
loadjewellery();
```

```
JOptionPane.showMessageDialog(this, "User status updated", "Success",  
JOptionPane.INFORMATION_MESSAGE);
```

```
jButton1.setEnabled(true);
```

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

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

```
}
```

```
}
```

```
}
```

## 7. Stock

**Jewelry SHOP**

08:06:52 PM

### Stock

Stock id	Jewellery id	Category	Name	Caret No	Quantity	Buying Price	Selling Price	Pitches Date	Status
67	7	Neclous	GOLD RGAL	24	2	6000.0	7000.0	2022-08-09	Availble
38	7	Neclous	GOLD RGAL	24	2	800.0	9000.0	2022-08-06	Availble
49	7	Neclous	GOLD RGAL	24	2	6000.0	7000.0	2022-08-07	Availble
55	5	Neclous	golden egal	25	4	40000.0	50000.0	2022-08-09	Unavailble
17	5	Neclous	golden egal	25	8	60000.0	80000.0	2022-08-01	Unavailble
26	1	Ring	golden nec	06	3	555.0	5555.0	2022-08-01	Unavailble
27	1	Ring	golden nec	06	2	2222.0	50000.0	2022-08-01	Unavailble
30	1	Ring	golden nec	06	2	60000.0	70000.0	2022-08-01	Unavailble
35	1	Ring	golden nec	06	2	6000.0	7000.0	2022-08-01	Unavailble
50	1	Ring	golden nec	06	2	2000.0	3000.0	2022-08-07	Unavailble

Buying Price : 0.00    Caret No :     PD : 8/10/22   

New Price :     Name :    

Category :

To load data into the table –

```
public void loadStock() {
```

```
try {
```

```
    ResultSet rs = MySql.search("SELECT DISTINCT  
`stock`.`Status`,`stock`.`id`,`jewellery`.`id`,`category`.`name`,`jewellery`.`caret_no`,`jewellery`.`name`,`stock`.`quantity`,`grn_item`.`buying_price`,`stock`.`selling_price`,`stock`.`mfd` FROM `stock`  
INNER JOIN `grn_item` ON `grn_item`.`stock_id`=`stock`.`id` INNER JOIN `jewellery` ON  
`stock`.`jewellery_id`=`jewellery`.`id` INNER JOIN `category` ON  
`jewellery`.`category_id`=`category`.`id` ORDER BY `jewellery`.`name` ASC");
```



```
DefaultTableModel dtm = (DefaultTableModel) jTable9.getModel();  
dtm.setRowCount(0);  
  
while (rs.next()) {  
    Vector v = new Vector();  
    v.add(rs.getString("stock.id"));  
    v.add(rs.getString("jewellery.id"));  
    v.add(rs.getString("category.name"));  
    v.add(rs.getString("jewellery.name"));  
    v.add(rs.getString("jewellery.caret_no"));  
    v.add(rs.getString("stock.quantity"));  
    v.add(rs.getString("grn_item.buying_price"));  
    v.add(rs.getString("stock.selling_price"));  
    v.add(rs.getString("stock.mfd"));  
    v.add(rs.getString("Status"));  
  
    dtm.addRow(v);  
}  
jTable9.setModel(dtm);  
  
} catch (Exception e) {  
    e.printStackTrace();  
}  
  
}
```

### **Caret No (ComboBox) –**

```
public void loadcaretno() {  
  
    try {  
  
        ResultSet rs = MySql.search("SELECT * FROM `jewellery`");  
  
        Vector v = new Vector();  
        v.add("Select");  
  
        while (rs.next()) {  
            v.add(rs.getString("caret_no"));  
  
        }  
  
        DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);  
        jComboBox15.setModel(dcm);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
}
```

### **Jwellery Name (ComboBox) –**

```
public void loadjewellername() {  
  
    try {  
  
        ResultSet rs = MySql.search("SELECT * FROM `jewellery`");  
  
        Vector v = new Vector();  
        v.add("Select");  
  
        while (rs.next()) {  
            v.add(rs.getString("name"));  
  
        }  
  
        DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);  
        jComboBox16.setModel(dcm);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
}
```

### Category (ComboBox) –

```
public void loadcategoryname() {  
  
    try {  
  
        ResultSet rs = MySql.search("SELECT * FROM `category`");  
  
        Vector v = new Vector();  
        v.add("Select");  
  
        while (rs.next()) {  
            v.add(rs.getString("name"));  
  
        }  
  
        DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);  
        jComboBox17.setModel(dcm);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
}
```

## Update button –

```
private void jButton29ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    String newPrice = jTextField32.getText();  
    String buyingPrice = jLabel131.getText();  
    int selectedRow = jTable9.getSelectedRow();  
  
    if(selectedRow== -1) {  
        JOptionPane.showMessageDialog(this, "Please select a  
stock", "Warning", JOptionPane.WARNING_MESSAGE);  
    }  
    else if(!Pattern.compile("(0)|([1-9][0-9]*)|([1-9][0-9]*).([0]*[1-9][0-9]*)|([0].)([0]*[1-9][0-9]*)").matcher(newPrice).matches()) {  
        JOptionPane.showMessageDialog(this, "Invalid Selling Price", "Warning",  
JOptionPane.WARNING_MESSAGE);  
    }else {  
        String stock_id = jTable9.getValueAt(jTable9.getSelectedRow(),0).toString();  
  
        if(Double.parseDouble(newPrice)<Double.parseDouble(buyingPrice)){  
            int x = JOptionPane.showConfirmDialog(this, "New price < Buying Price, Do you want  
to Continue", "Warning", JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);  
  
            if(x==JOptionPane.YES_NO_OPTION){  
                try {  
                    MySql.iud("UPDATE `stock` SET `selling_price`='"+newPrice+"' WHERE  
`id`='"+stock_id+"'");  
                } catch (Exception ex) {
```

```

        Logger.getLogger(Stock.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}else {
    try {
        MySql.iud("UPDATE `stock` SET `selling_price`='"+newPrice+"' WHERE
`id`='"+stock_id+"'");
    } catch (Exception ex) {
        Logger.getLogger(Stock.class.getName()).log(Level.SEVERE, null, ex);
    }
}
//reset();
}
}

```

### **Resetbutton -**

```

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    jComboBox15.setSelectedIndex(0);
    jComboBox16.setSelectedIndex(0);
    jComboBox17.setSelectedIndex(0);
    jTextField32.setText("");
    dateChooserCombo1.setDateFormat(null);

    loadStock();
}

```

## 8. Reports



## I. GRN

Id

None

Caret No

None

Category

None

Quantity

Name

None

Buying Price

Add To GRN

Select Jewellery

PD

Selling Price

Category	Jewellery Id	caret No	Name	Quantity	Buying Price	Selling Price	PD	Total

Select Supplier

ID

None

Name


None

Contact Number

None

Branch

None



Total Payment

None

Payment

Payment Meth...

Select

Balance

None

Print GRN

### Add GRN button –

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String sid = jLabel5.getText();  
    String pid = jLabel10.getText();  
    String qty = jTextField1.getText();  
    String buyingPrice = jTextField2.getText();
```



```

//Update
String selling_price = jTextField4.getText();
Date mfd = jDateChooser1.getDate();

//Update
if (sid.equals("None")) {
    JOptionPane.showMessageDialog(this, "Please select supplier", "Warning",
JOptionPane.WARNING_MESSAGE);
} else if (pid.equals("None")) {
    JOptionPane.showMessageDialog(this, "Please select product", "Warning",
JOptionPane.WARNING_MESSAGE);
} else if (!Pattern.compile("[1-9][0-9]*").matcher(qty).matches()) {
    JOptionPane.showMessageDialog(this, "Invalid quantity", "Warning",
JOptionPane.WARNING_MESSAGE);
} else if (!Pattern.compile("([1-9][0-9]*)|(((1-9)[0-9]*).([0]*[1-9][0-
9]*))|([0].([0]*[1-9][0-9]*))").matcher(buyingPrice).matches()) {
    JOptionPane.showMessageDialog(this, "Invalid buying price", "Warning",
JOptionPane.WARNING_MESSAGE);
} //Update
else if (!Pattern.compile("([1-9][0-9]*)|(((1-9)[0-9]*).([0]*[1-9][0-
9]*))|([0].([0]*[1-9][0-9]*))").matcher(selling_price).matches()) {
    JOptionPane.showMessageDialog(this, "Invalid selling price", "Warning",
JOptionPane.WARNING_MESSAGE);
} else if (Double.parseDouble(buyingPrice) > Double.parseDouble(selling_price)) {
    JOptionPane.showMessageDialog(this, "Invalid buying price & selling price",
"Warning", JOptionPane.WARNING_MESSAGE);
} else if (mfd == null) {
    JOptionPane.showMessageDialog(this, "Invalid MFD", "Warning",
JOptionPane.WARNING_MESSAGE);
} else if (mfd.after(new Date())) {
    JOptionPane.showMessageDialog(this, "Invalid MFD", "Warning",
JOptionPane.WARNING_MESSAGE);

} //Update
else {

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

    boolean isFound = false;
    int x = -1;

```

```

for (int i = 0; i < dtm.getRowCount(); i++) {
    String id = jTable1.getValueAt(i, 1).toString();

    if (id.equals(pid)) {
        isFound = true;
        x = i;
        break;
    }
}

if (isFound) {

    int option = JOptionPane.showConfirmDialog(this, "This product is already
added. Do you want to update?", "Confirmation", JOptionPane.YES_NO_OPTION,
JOptionPane.INFORMATION_MESSAGE);

    if (option == JOptionPane.YES_NO_OPTION) {

        int oldQty = Integer.parseInt(jTable1.getValueAt(x, 4).toString());

        int finalQty = oldQty + Integer.parseInt(qty);

        jTable1.setValueAt(String.valueOf(finalQty), x, 4);
        jTable1.setValueAt(buyingPrice, x, 5);

        double UpdatedItemTotal = finalQty * Double.parseDouble(buyingPrice);
        jTable1.setValueAt(String.valueOf(UpdatedItemTotal), x, 9);

        updateTotal();

        //double total = 0;
        //for (int i = 0; i < dtm.getRowCount(); i++) {
            //String t = jTable1.getValueAt(i, 6).toString();
            //total = total + Double.parseDouble(t);
            //}
        //jLabel20.setText(df.format(total));
    }
    resetFields();

} else {

```

```

        Vector v = new Vector();
        v.add(jLabel16.getText());
        v.add(pid);
        v.add(jLabel14.getText());
        v.add(jLabel12.getText());
        v.add(qty);
        v.add(buyingPrice);
        v.add(selling_price);
        //Update

        v.add(sdf.format(mfd));

        //Update

        double itemtotal = Integer.parseInt(qty) * Double.parseDouble(buyingPrice);
        v.add(String.valueOf(itemtotal));

        dtm.addRow(v);

        //double total = 0;
        //for (int i = 0; i < dtm.getRowCount(); i++) {
            //String t = jTable1.getValueAt(i, 6).toString();
            //total = total + Double.parseDouble(t);
            //}
        //jLabel20.setText(df.format(total));
        updateTotal();

        resetFields();

        JOptionPane.showMessageDialog(this, "Product added to the GRN",
        "Success", JOptionPane.INFORMATION_MESSAGE);

        if (jTable1.getRowCount() == 1) {
            jButton1.setText("Update Supplier");
            jButton1.setEnabled(false);
        }

    }

}
}

```

### Table MouseClicked -

```
private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:

    if (evt.getClickCount() == 2) {
        int r = jTable1.getSelectedRow();

        if (r == -1) {
            JOptionPane.showMessageDialog(this, "Please select a grn item", "Warning",
JOptionPane.WARNING_MESSAGE);
        } else {

            int option = JOptionPane.showConfirmDialog(this, "Do you want to remove
this item?", "Confirmation", JOptionPane.YES_NO_OPTION,
JOptionPane.INFORMATION_MESSAGE);

            if (option == JOptionPane.YES_OPTION) {

                DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();
                dtm.removeRow(r);

                updateTotal();
                //payments
                fulltotal.setText("0.00");
                jTextField3.setText("");
                jTextField3.setEditable(false);
                jLabel24.setText("0.00");
                jComboBox1.setSelectedIndex(0);
                //payments

                JOptionPane.showMessageDialog(this, "GRN item removed", "Success",
JOptionPane.INFORMATION_MESSAGE);

            }
        }
    }
}
```

### **Payment Method (Combobox) –**

```
public void loadPaymentType() {  
  
    try {  
  
        ResultSet rs = MySql.search("SELECT * FROM `payment_type`");  
  
        Vector v = new Vector();  
        v.add("Select");  
  
        while (rs.next()) {  
            v.add(rs.getString("name"));  
        }  
  
        DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);  
        jComboBox1.setModel(dcm);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
}
```

### **Total -**

```
public void updateTotal() {  
    double total = 0;  
  
    for (int i = 0; i < jTable1.getRowCount(); i++) {  
        String t = jTable1.getValueAt(i, 8).toString();  
        total = total + Double.parseDouble(t);  
    }  
  
    fulltotal.setText(df.format(total));  
}
```

### **ResetField –**

```
public void resetFields() {  
  
    jLabel10.setText("None");  
    jLabel12.setText("None");  
    jLabel14.setText("None");  
    jLabel16.setText("None");  
    jTextField1.setText("");  
    jTextField2.setText("");  
    //update  
    jTextField4.setText("");  
    jDateChooser1.setDate(null);  
  
    //update  
    jTextField1.grabFocus();  
  
}
```

### **Print GRN –**

```
public void printGRN() {  
    //Print Report  
    try {
```

```
Date d = new Date();

SimpleDateFormat s = new SimpleDateFormat("dd-MM-yyyy");

String date = s.format(d);


String grnid = jLabel12.getText();

String sname = jLabel6.getText();

String cno = jLabel7.getText();

String branch = jLabel8.getText();

String pmethod = jComboBox1.getSelectedItem().toString();

String ptotal = jLabel20.getText();

String paid = jTextField3.getText();

String balance = jLabel24.getText();


//      String filePath = "src//reports//grn.jrxml";

InputStream stream = GRN1.class.getResourceAsStream("GRN11.jrxml");

JasperReport jr = JasperCompileManager.compileReport(stream);


//      JasperReport jr = JasperCompileManager.compileReport(filePath);

HashMap parameters = new HashMap();

parameters.put("date", date);

// parameters.put("grnid", uniqueld);

parameters.put("sname", sname);

parameters.put("cno", cno);

parameters.put("branch", branch);

parameters.put("pmethod", pmethod);

parameters.put("paid", paid);
```

```

        parameters.put("balance", balance);
        parameters.put("total", ptotal);
        TableModel tm = jTable1.getModel();
        JTableModelDataSource dataSource = new JTableModelDataSource(tm);
//        JREmptyDataSource dataSource = new JREmptyDataSource();

        JasperPrint jp = JasperFillManager.fillReport(jr, parameters, dataSource);
        JasperViewer.viewReport(jp,false);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

### **Payment (textfield) –**

```

private void jTextField3KeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:

    String price = jTextField3.getText();
    String text = price + evt.getKeyChar();

    if (!Pattern.compile("(0|0[.]|0[.][0-9]*)|([1-9]|([1-9][0-9]*|([1-9][0-9]*[.])|([1-9][0-9]*[.][0-9]*)").matcher(text).matches()) {
        evt.consume();
    }
}

```



### Print GRN Button -

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    String paymentType = jComboBox1.getSelectedItem().toString();
    String payment = jTextField3.getText();

    if (jTable1.getRowCount() == 0) {
        JOptionPane.showMessageDialog(this, "Please add a Jewellery ", "Warning",
        JOptionPane.WARNING_MESSAGE);
    } else if (paymentType.equals("Select")) {
        JOptionPane.showMessageDialog(this, "Please select payment method",
        "Warning", JOptionPane.WARNING_MESSAGE);
    } else if (!Pattern.compile("(0)|([1-9][0-9]*)|(((1-9)[0-9]*)([0-9]*[1-9][0-9]*)|([0-9]*[1-9][0-9]*)|([0-9]*[1-9][0-9]*))").matcher(payment).matches()) {
        JOptionPane.showMessageDialog(this, "Invalid payment", "Warning",
        JOptionPane.WARNING_MESSAGE);
    } else {

        //GRN Insert
        long mTime = System.currentTimeMillis();

        String uniqueId = mTime + "-" + SignUp.userId;

        String cid = jLabel5.getText();

        SimpleDateFormat sdf2 = new SimpleDateFormat("yyyy-MM-dd");
        String dNow = sdf2.format(new Date());

        try {

            MySql.iud("INSERT INTO
            `grn`(`supplier_id`,`date_time`,`employee_id`,`unique_id`) VALUES('" +
            Integer.parseInt(cid) + "','" + dNow + "','" + SignUp.userId + "','" + uniqueId + "')");
            System.out.println("ok");
            //GRN Payment Insert
```

```

        ResultSet rs = MySql.search("SELECT * FROM `grn` WHERE `unique_id`='" +
uniqueId + "'");
        rs.next();
        String id = rs.getString("id");
        System.out.println("ok");

        ResultSet rs2 = MySql.search("SELECT * FROM `payment_type` WHERE
`name`='" + paymentType + "'");
        rs2.next();
        String paymentTupId = rs2.getString("id");

        System.out.println("ok");

        String balance = jLabel24.getText();

        MySql.iud("INSERT INTO
`grn_payment`(`grn_id`,`payment_type_id`,`payment`,`balance`) VALUES('" +
Integer.parseInt(id) + "','" + Integer.parseInt(paymentTupId) + "','" + payment + "','" +
balance + "')");
        //GRN payment Insert

        //GRN item Insert & stock Insert or update
        for (int i = 0; i < jTable1.getRowCount(); i++) {

            String pid = jTable1.getValueAt(i, 1).toString();
            String qty = jTable1.getValueAt(i, 4).toString();

            String buyingPrice = jTable1.getValueAt(i, 5).toString();
            String sellingPrice = jTable1.getValueAt(i, 6).toString();
            String mfd = jTable1.getValueAt(i, 7).toString();
            //String exd = jTable1.getValueAt(i, 8).toString();
            //          grn_id = id;
            //          stock_id = ?

            ResultSet rs3 = MySql.search("SELECT * FROM `stock` WHERE
`jewellery_id`='" + pid + "'AND `selling_price` = '" + sellingPrice + "' AND `mfd` = '" + mfd
+ "' ");
            {
                String stock_id;
                if (rs3.next()) {
                    //update

```

```

        stock_id = rs3.getString("id");
        String stock_qty = rs3.getString("quantity");

        int updatedQty = Integer.parseInt(stock_qty) + Integer.parseInt(qty);

        MySql.iud("UPDATE `stock` SET `quantity`='" + updatedQty + "' WHERE
`id`='" + stock_id + "'");

    } else {
        //insert

        MySql.iud("INSERT INTO
`stock`(`jewellery_id`,`quantity`,`selling_price`,`mfd`) VALUES('" + pid + "','" + qty + "','"
+ sellingPrice + "','" + mfd + "')");

        ResultSet rs4 = MySql.search("SELECT * FROM `stock` WHERE
`jewellery_id`='" + pid + "' AND `selling_price`='" + sellingPrice + "' AND `mfd`='" + mfd
+ "'");

        rs4.next();
        stock_id = rs4.getString("id");

    }

    MySql.iud("INSERT INTO
`grn_item`(`quantity`,`buying_price`,`grn_id`,`stock_id`) VALUES('" + qty + "','" +
buyingPrice + "','" + Integer.parseInt(id)+ "','" + Integer.parseInt(stock_id) + "')");
    }
}

resetFields();

//supplier
jButton1.setEnabled(true);
jButton1.setText("Select Customer");
jLabel5.setText("none");
jLabel6.setText("none");
jLabel7.setText("none");
jLabel8.setText("none");
//supplier

//payments
fulltotal.setText("0.00");

```

```

        jTextField3.setText("");
        jTextField3.setEditable(false);
        jLabel24.setText("0.00");
        jComboBox1.setSelectedIndex(0);
        //payments

//        DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();
//        dtm.setRowCount(0);

        JOptionPane.showMessageDialog(this, "New GRN Created", "Success",
JOptionPane.INFORMATION_MESSAGE);

        String filePath =
"C:\\Users\\DELL\\JaspersoftWorkspace\\MyReports\\GRN11.jrxml";
        JasperReport jr = JasperCompileManager.compileReport(filePath);

        HashMap parameters = new HashMap();

//        JREmptyDataSource dataSource = new JREmptyDataSource();
        TableModel tm = jTable1.getModel();
        JRTableModelDataSource dataSource = new JRTableModelDataSource(tm);

        JasperPrint jp = JasperFillManager.fillReport(jr, parameters, dataSource);

        JasperViewer.viewReport(jp,false);

        DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();
        dtm.setRowCount(0);

        //GRN item Insert & stock Insert or update
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

## Invoice Code –

Id : None

Caret No : None

Quanti...

Name : None

Categor... None

Selling Price :

Select Jewellery

PD :

Add To Invoice

Customer Name	Jwellery Id	Caret NO	Name	Quantity	Selling Price	PD	Total
---------------	-------------	----------	------	----------	---------------	----	-------

Select Customer

ID None

Fiest Name None

Last Name None

Contact No None

Total Payment : None

Payment

Print Invoice

Payment Meth...

Balance : None

package gui;

```
import java.awt.Color;
import java.io.InputStream;
import java.sql.ResultSet;
import java.text.DecimalFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
```

```
import java.util.Vector;
import java.util.regex.Pattern;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
import model.MySql;
import net.sf.jasperreports.engine.JasperCompileManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.data.JRTableModelDataSource;
import net.sf.jasperreports.view.JasperViewer;
```

```
/**
```

```
*
```

```
* @author DELL
```

```
*/
```

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

```
    DecimalFormat df = new DecimalFormat("0.00");
```

```
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
```

```
/**
```

```
* Creates new form Invoice
```

```
*/
```

```
public Invoice() {
```

```
initComponents();  
loadPaymentType();  
}
```

```
public void updateTotal() {  
    double total = 0;  
  
    for (int i = 0; i < jTable1.getRowCount(); i++) {  
        String t = jTable1.getValueAt(i, 7).toString();  
        total = total + Double.parseDouble(t);  
    }
```

```
    fulltotal.setText(df.format(total));  
}
```

```
public void loadPaymentType() {  
  
    try {  
  
        ResultSet rs = MySql.search("SELECT * FROM `payment_type`");  
  
        Vector v = new Vector();  
        v.add("Select");  
  
        while (rs.next()) {  
            v.add(rs.getString("name"));  
        }
```

```
        DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);
        jComboBox1.setModel(dcm);

    } catch (Exception e) {
        e.printStackTrace();
    }

}

public void resetFields() {

    jLabel10.setText("None");
    jLabel12.setText("None");
    jLabel14.setText("None");
    jLabel16.setText("None");
    jTextField1.setText("");
    jTextField2.setText("");
    //update

    jDateChooser1.setDate(null);

    //update
    jTextField1.grabFocus();

}
```



```
/**
 * 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() {

    jPanel1 = new javax.swing.JPanel();
    jPanel3 = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jPanel4 = new javax.swing.JPanel();
    jLabel9 = new javax.swing.JLabel();
    jLabel10 = new javax.swing.JLabel();
    jLabel11 = new javax.swing.JLabel();
    jLabel12 = new javax.swing.JLabel();
    jLabel13 = new javax.swing.JLabel();
    jLabel14 = new javax.swing.JLabel();
    jLabel15 = new javax.swing.JLabel();
    jLabel16 = new javax.swing.JLabel();
    jLabel17 = new javax.swing.JLabel();
    jTextField1 = new javax.swing.JTextField();
    jTextField2 = new javax.swing.JTextField();
    jButton2 = new javax.swing.JButton();
    jButton3 = new javax.swing.JButton();
}
```

```
jLabel25 = new javax.swing.JLabel();
jLabel26 = new javax.swing.JLabel();
jDateChooser1 = new com.toedter.calendar.JDateChooser();
jPanel5 = new javax.swing.JPanel();
jLabel19 = new javax.swing.JLabel();
fulltotal = new javax.swing.JLabel();
jLabel21 = new javax.swing.JLabel();
jComboBox1 = new javax.swing.JComboBox<>();
jLabel22 = new javax.swing.JLabel();
jTextField3 = new javax.swing.JTextField();
jLabel23 = new javax.swing.JLabel();
jLabel24 = new javax.swing.JLabel();
jButton5 = new javax.swing.JButton();
kGradientPanel1 = new com.k33ptoo.components.KGradientPanel();
jPanel2 = new javax.swing.JPanel();
jButton1 = new javax.swing.JButton();
jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
id = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
jLabel7 = new javax.swing.JLabel();
jLabel8 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();

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

```

setResizable(false);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

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

        },
    new String [] {
        "Customer Name", "Jwellery Id", "Caret NO", "Name", "Quantity", "Selling Price", "PD",
        "Total"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});

jTable1.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jTable1MouseClicked(evt);
    }
});

jScrollPane1.setViewportViewView(jTable1);

```

```
javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addGap()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 1147,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
jPanel3Layout.setVerticalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addGap()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 262,
Short.MAX_VALUE)
            .addGap())
        );

jLabel9.setText("Id :");

jLabel10.setText("None");

jLabel11.setText("Name :");

jLabel12.setText("None");
```

```
jLabel13.setText("Caret No :");
```

```
jLabel14.setText("None");
```

```
jLabel15.setText("Category :");
```

```
jLabel16.setText("None");
```

```
jLabel17.setText("Quantity :");
```

```
jTextField1.addKeyListener(new java.awt.event.KeyAdapter() {  
    public void keyTyped(java.awt.event.KeyEvent evt) {  
        jTextField1KeyTyped(evt);  
    }  
});
```

```
jTextField2.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jTextField2ActionPerformed(evt);  
    }  
});
```

```
jTextField2.addKeyListener(new java.awt.event.KeyAdapter() {  
    public void keyTyped(java.awt.event.KeyEvent evt) {  
        jTextField2KeyTyped(evt);  
    }  
});
```



```
.addContainerGap()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup())

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE, 48,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 67,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED_SIZE, 174,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup())
    .addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED_SIZE, 66,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jLabel14, javax.swing.GroupLayout.PREFERRED_SIZE, 97,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(jLabel17, javax.swing.GroupLayout.PREFERRED_SIZE, 56,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(35, 35, 35)
```

```

        .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 127,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGroup(jPanel4Layout.createSequentialGroup())

        .addComponent(jLabel15, javax.swing.GroupLayout.PREFERRED_SIZE, 66,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jLabel16, javax.swing.GroupLayout.PREFERRED_SIZE, 199,
javax.swing.GroupLayout.PREFERRED_SIZE))))

        .addGroup(jPanel4Layout.createSequentialGroup())

        .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 338,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(78, 78, 78)

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(jLabel26, javax.swing.GroupLayout.PREFERRED_SIZE, 39,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(jLabel25))

            .addGap(38, 38, 38)

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)

            .addComponent(jTextField2)

            .addComponent(jDateChooser1, javax.swing.GroupLayout.DEFAULT_SIZE, 123,
Short.MAX_VALUE))

            .addGap(42, 42, 42)

            .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 182,
javax.swing.GroupLayout.PREFERRED_SIZE)))

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

    );

    jPanel4Layout.setVerticalGroup(

```



```
jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup())
        .addContainerGap()

    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel9)
        .addComponent(jLabel10)
        .addComponent(jLabel17)
        .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel13)
        .addComponent(jLabel14))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel11)
        .addComponent(jLabel12)
        .addComponent(jLabel15)
        .addComponent(jLabel16))

    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel4Layout.createSequentialGroup())

    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel25)
        .addComponent(jButton3))
        .addGap(18, 18, 18)
```

```
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(jLabel26)
    .addComponent(jDateChooser1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addGap(18, 18, 18)
        .addComponent(jButton2)))
    .addContainerGap(44, Short.MAX_VALUE))
);
```

```
jLabel19.setText("Total Payment :");
```

```
fulltotal.setFont(new java.awt.Font("Dialog", 1, 11)); // NOI18N
```

```
fulltotal.setForeground(new java.awt.Color(255, 0, 0));
```

```
fulltotal.setText("None");
```

```
jLabel21.setText("Payment Method");
```

```
jComboBox1.addItemListener(new java.awt.event.ItemListener() {
```

```
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
```

```
        jComboBox1ItemStateChanged(evt);
```

```
    }
```

```
});
```

```
jLabel22.setText("Payment");
```

```
jTextField3.setEditable(false);
```

```
jTextField3.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jTextField3ActionPerformed(evt);  
    }  
});  
  
jTextField3.addKeyListener(new java.awt.event.KeyAdapter() {  
    public void keyReleased(java.awt.event.KeyEvent evt) {  
        jTextField3KeyReleased(evt);  
    }  
    public void keyTyped(java.awt.event.KeyEvent evt) {  
        jTextField3KeyTyped(evt);  
    }  
});  
  
jLabel23.setText("Balance :");  
  
jLabel24.setText("None");  
  
jButton5.setText("Print Invoice");  
jButton5.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButton5ActionPerformed(evt);  
    }  
});  
  
javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);  
jPanel5.setLayout(jPanel5Layout);
```

```

jPanel5Layout.setHorizontalGroup(
    jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel5Layout.createSequentialGroup())
            .addContainerGap()

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel5Layout.createSequentialGroup()
        .addComponent(jLabel19, javax.swing.GroupLayout.PREFERRED_SIZE, 105,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(fulltotal, javax.swing.GroupLayout.PREFERRED_SIZE, 166,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(jPanel5Layout.createSequentialGroup()
                .addComponent(jLabel21, javax.swing.GroupLayout.PREFERRED_SIZE, 107,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE, 245,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(64, 64, 64)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addComponent(jLabel23, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jLabel22, javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(jLabel24, javax.swing.GroupLayout.PREFERRED_SIZE, 129,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE, 239,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(jButton5, javax.swing.GroupLayout.PREFERRED_SIZE, 282,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap())
);

jPanel5Layout.setVerticalGroup(

jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel5Layout.createSequentialGroup())

.addContainerGap()

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)

.addComponent(jButton5, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addGroup(jPanel5Layout.createSequentialGroup())

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel19)

.addComponent(fulltotal)

.addComponent(jLabel22)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```
.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel21)
    .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel23)
    .addComponent(jLabel24))))))

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

);
```

```
kGradientPanel1.setkEndColor(new java.awt.Color(102, 204, 255));
```

```
jButton1.setText("Select Customer");

jButton1.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton1MouseClicked(evt);
    }
});

jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
```

```
jLabel1.setText("ID");
```

```
jLabel2.setText("Fiest Name");
```

```
jLabel3.setText("Last Name");
```

```
jLabel4.setText("Contact No");
```

```
id.setText("None");
```

```
jLabel6.setText("None");
```

```
jLabel7.setText("None");
```

```
jLabel8.setText("None");
```

```
javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
```

```
jPanel2.setLayout(jPanel2Layout);
```

```
jPanel2Layout.setHorizontalGroup(
```

```
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(jPanel2Layout.createSequentialGroup()
```

```
            .addGap()
```

```
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
            .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
            .addGroup(jPanel2Layout.createSequentialGroup()
```

```
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,  
false)
```

```

        .addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE, 101,
Short.MAX_VALUE)

        .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(id, javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(0, 64, Short.MAX_VALUE)))

.addContainerGap()

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

        .addContainerGap()

        .addComponent(jButton1)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

```



```

        .addComponent(jLabel1)
        .addComponent(id))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel2)
        .addComponent(jLabel6))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel3)
        .addComponent(jLabel7))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel4)
        .addComponent(jLabel8))
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

```

```

jLabel5.setIcon(new javax.swing.ImageIcon(getClass().getResource("/image/jewelry-logo-
design-template-cir_1.png"))); // NOI18N

```

```

    javax.swing.GroupLayout kGradientPanel1Layout = new
    javax.swing.GroupLayout(kGradientPanel1);

```

```

    kGradientPanel1.setLayout(kGradientPanel1Layout);

```

```

    kGradientPanel1Layout.setHorizontalGroup(

```

```

    kGradientPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(kGradientPanel1Layout.createSequentialGroup())

        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 126,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(31, 31, 31))
    );

    kGradientPanel1Layout.setVerticalGroup(

kGradientPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
kGradientPanel1Layout.createSequentialGroup())

        .addContainerGap()

.addGroup(kGradientPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TR
AILING)

        .addGroup(kGradientPanel1Layout.createSequentialGroup())

        .addGap(0, 0, Short.MAX_VALUE)

        .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 131,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

        .addGap(14, 14, 14))
    );

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(

```

```

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addComponent(kGradientPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addGroup(jPanel1Layout.createSequentialGroup())

    .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGap(0, 4, Short.MAX_VALUE))

    .addComponent(jPanel5, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel1Layout.createSequentialGroup())

    .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addComponent(kGradientPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

);

```

```

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE))
        );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );

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

```

```

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    if (evt.getClickCount() == 2) {
        jButton1.setEnabled(true);
        jButton1.setText("Select Supplier");

        id.setText("None");
        jLabel6.setText("None");
        jLabel7.setText("None");
    }
}

```

```
        jLabel8.setText("None");  
    }  
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    Customer_Registration sr = new Customer_Registration(this);  
    sr.setVisible(true);  
}
```

```
private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
  
    if (evt.getClickCount() == 2) {  
        int r = jTable1.getSelectedRow();  
  
        if (r == -1) {  
            JOptionPane.showMessageDialog(this, "Please select a invoice item", "Warning",  
JOptionPane.WARNING_MESSAGE);  
        } else {  
  
            int option = JOptionPane.showConfirmDialog(this, "Do you want to remove this  
item?", "Confirmation", JOptionPane.YES_NO_OPTION,  
JOptionPane.INFORMATION_MESSAGE);  
  
            if (option == JOptionPane.YES_OPTION) {  
  
                DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();
```

```

        dtm.removeRow(r);

        updateTotal();

        //payments
        fulltotal.setText("0.00");
        jTextField3.setText("");
        jTextField3.setEditable(false);
        jLabel24.setText("0.00");
        jComboBox1.setSelectedIndex(0);
        //payments

        JOptionPane.showMessageDialog(this, "GRN item removed", "Success",
JOptionPane.INFORMATION_MESSAGE);

    }
}
}
}

private void jTextField1KeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:

    String qty = jTextField1.getText();
    String text = qty + evt.getKeyChar();

    if (!Pattern.compile("[1-9][0-9]*").matcher(text).matches()) {
        evt.consume();
    }
}

```

```
}  
}
```

```
private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void jTextField2KeyTyped(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:
```

```
    String price = jTextField2.getText();  
    String text = price + evt.getKeyChar();
```

```
    if (!Pattern.compile("(0|0[.]|0[.][0-9]*)| [1-9]| [1-9][0-9]*| [1-9][0-9]*[.]| [1-9][0-9]*[.][0-9]*").matcher(text).matches()) {  
        evt.consume();  
    }  
}
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    Add_juwellari pr = new Add_juwellari(this);  
    pr.setVisible(true);  
}
```

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:
```

```

String sid = id.getText();

String pid = jLabel10.getText();

String qty = jTextField1.getText();


//Update

String selling_price = jTextField2.getText();

Date mfd = jDateChooser1.getDate();


//Update

if (sid.equals("None")) {

    JOptionPane.showMessageDialog(this, "Please select Customer", "Warning",
JOptionPane.WARNING_MESSAGE);

} else if (pid.equals("None")) {

    JOptionPane.showMessageDialog(this, "Please select Jewellery", "Warning",
JOptionPane.WARNING_MESSAGE);

} else if (!Pattern.compile("[1-9][0-9]*").matcher(qty).matches()) {

    JOptionPane.showMessageDialog(this, "Invalid quantity", "Warning",
JOptionPane.WARNING_MESSAGE);

    /*} else if (!Pattern.compile("([1-9][0-9]*)|(([1-9][0-9]*)[.][0]*[1-9][0-9]*)|([0][.][0]*[1-9][0-9][0-9]*)").matcher(buyingPrice).matches()) {

        JOptionPane.showMessageDialog(this, "Invalid buying price", "Warning",
JOptionPane.WARNING_MESSAGE);*/

} //Update

else if (!Pattern.compile("([1-9][0-9]*)|(([1-9][0-9]*)[.][0]*[1-9][0-9]*)|([0][.][0]*[1-9][0-9][0-9]*)").matcher(selling_price).matches()) {

    JOptionPane.showMessageDialog(this, "Invalid selling price", "Warning",
JOptionPane.WARNING_MESSAGE);

    /* } else if (Double.parseDouble(buyingPrice) > Double.parseDouble(selling_price)) {

```



```
        JOptionPane.showMessageDialog(this, "Invalid buying price & selling price", "Warning",  
JOptionPane.WARNING_MESSAGE);*/
```

```
    } else if (mfd == null) {
```

```
        JOptionPane.showMessageDialog(this, "Invalid MFD", "Warning",  
JOptionPane.WARNING_MESSAGE);
```

```
    } else if (mfd.after(new Date())) {
```

```
        JOptionPane.showMessageDialog(this, "Invalid MFD", "Warning",  
JOptionPane.WARNING_MESSAGE);
```

```
    } //Update
```

```
    else {
```

```
        DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();
```

```
        boolean isFound = false;
```

```
        int x = -1;
```

```
        for (int i = 0; i < dtm.getRowCount(); i++) {
```

```
            String id = jTable1.getValueAt(i, 1).toString();
```

```
            if (id.equals(pid)) {
```

```
                isFound = true;
```

```
                x = i;
```

```
                break;
```

```
            }
```

```
        }
```

```
        if (isFound) {
```

```
int option = JOptionPane.showConfirmDialog(this, "This product is already added. Do  
you want to update?", "Confirmation", JOptionPane.YES_NO_OPTION,  
JOptionPane.INFORMATION_MESSAGE);
```

```
if (option == JOptionPane.YES_NO_OPTION) {
```

```
int oldQty = Integer.parseInt(jTable1.getValueAt(x, 4).toString());
```

```
int finalQty = oldQty + Integer.parseInt(qty);
```

```
jTable1.setValueAt(String.valueOf(finalQty), x, 4);
```

```
jTable1.setValueAt(selling_price, x, 5);
```

```
double UpdatedItemTotal = finalQty * Double.parseDouble(selling_price);
```

```
jTable1.setValueAt(String.valueOf(UpdatedItemTotal), x, 8);
```

```
updateTotal();
```

```
//double total = 0;
```

```
//for (int i = 0; i < dtm.getRowCount(); i++) {
```

```
//String t = jTable1.getValueAt(i, 6).toString();
```

```
//total = total + Double.parseDouble(t);
```

```
//}
```

```
//jLabel20.setText(df.format(total));
```

```
}
```

```
resetFields();
```

```
} else {

    Vector v = new Vector();
    v.add(jLabel5.getText());
    v.add(pid);
    v.add(jLabel14.getText());
    v.add(jLabel12.getText());
    v.add(qty);
    v.add(selling_price);

    //Update

    v.add(sdf.format(mfd));

    //Update

    double itemtotal = Integer.parseInt(qty) * Double.parseDouble(selling_price);
    v.add(String.valueOf(itemtotal));

    dtm.addRow(v);

    //double total = 0;
    //for (int i = 0; i < dtm.getRowCount(); i++) {
    //String t = jTable1.getValueAt(i, 6).toString();
    //total = total + Double.parseDouble(t);
    //}

    //jLabel20.setText(df.format(total));

    updateTotal();
```

```
resetFields();
```

```
JOptionPane.showMessageDialog(this, "Jwellery added to the Invoice", "Success",  
JOptionPane.INFORMATION_MESSAGE);
```

```
if (jTable1.getRowCount() == 1) {  
    jButton1.setText("Update Supplier");  
    jButton1.setEnabled(false);  
}
```

```
}
```

```
}
```

```
}
```

```
private void jComboBox1ItemStateChanged(java.awt.event.ItemEvent evt) {  
    // TODO add your handling code here:
```

```
String text = jComboBox1.getSelectedItem().toString();
```

```
if (text.equals("Select")) {  
    jTextField3.setEditable(false);  
    jTextField3.setText("");  
    jLabel24.setText("0.00");  
    jLabel24.setForeground(Color.BLACK);  
} else {  
    jTextField3.setEditable(true);
```

```
}  
}
```

```
private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void jTextField3KeyReleased(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:
```

```
    if (jTextField3.getText().isEmpty()) {  
        jLabel24.setText("0.00");  
        jLabel24.setForeground(Color.BLACK);  
    } else {  
        String total = fulltotal.getText();  
        String payment = jTextField3.getText();
```

```
        double balance = Double.parseDouble(payment) - Double.parseDouble(total);
```

```
        if (balance < 0) {  
            jLabel24.setForeground(Color.RED);  
        } else {  
            jLabel24.setForeground(Color.BLACK);  
        }
```

```
        jLabel24.setText(String.valueOf(balance));  
    }
```

```
}
```

```
private void jTextField3KeyTyped(java.awt.event.KeyEvent evt) {
```

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

```
    String price = jTextField3.getText();
```

```
    String text = price + evt.getKeyChar();
```

```
    if (!Pattern.compile("(0|0[.]|0[.][0-9]*)|[[1-9]]|[[1-9][0-9]*|[[1-9][0-9]*[.]]|[[1-9][0-9]*[.][0-9]*)").matcher(text).matches()) {
```

```
        evt.consume();
```

```
    }
```

```
}
```

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
```

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

```
    String paymentType = jComboBox1.getSelectedItem().toString();
```

```
    String payment = jTextField3.getText();
```

```
    if (jTable1.getRowCount() == 0) {
```

```
        JOptionPane.showMessageDialog(this, "Please add a product ", "Warning",  
JOptionPane.WARNING_MESSAGE);
```

```
    } else if (paymentType.equals("Select")) {
```

```
        JOptionPane.showMessageDialog(this, "Please select payment method", "Warning",  
JOptionPane.WARNING_MESSAGE);
```

```
    } else if (!Pattern.compile("(0)|([1-9][0-9]*)|(((1-9)[0-9]*[.])([0]*[1-9][0-9]*))|([0][.])([0]*[1-9][0-9]*)").matcher(payment).matches()) {
```

```
        JOptionPane.showMessageDialog(this, "Invalid payment", "Warning",  
JOptionPane.WARNING_MESSAGE);
```

```

} else {

    //GRN Insert

    long mTime = System.currentTimeMillis();

    String uniqueId = mTime + "-" + SignUp.userId;

    String sid = id.getText();

    SimpleDateFormat sdf2 = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    String dNow = sdf2.format(new Date());

    try {

        MySql.iud("INSERT INTO `invoice`(`date_time`,`user_id`,`employee_id`,`unique_id`)
VALUES('" + dNow + "'," + Integer.parseInt(sid) + "," + SignUp.userId + "," + uniqueId + "')");

        //GRN Payment Insert

        ResultSet rs = MySql.search("SELECT * FROM `invoice` WHERE `unique_id`='" +
uniqueId + "'");

        rs.next();

        String id = rs.getString("id");

        ResultSet rs2 = MySql.search("SELECT * FROM `payment_type` WHERE `name`='" +
paymentType + "'");

        rs2.next();

        String paymentTupId = rs2.getString("id");

```

```
String balance = jLabel24.getText();
```

```
MySQL.iud("INSERT INTO  
`invoice_payment`(`invoice_id`,`payment_type_id`,`payment`,`balance`) VALUES('"  
Integer.parseInt(id) + "','" + Integer.parseInt(paymentTupeld) + "','" + payment + "','" + balance  
+ "')");
```

```
//GRN payment Insert
```

```
//GRN item Insert & stock Insert or update
```

```
for (int i = 0; i < jTable1.getRowCount(); i++) {
```

```
String pid = jTable1.getValueAt(i, 1).toString();
```

```
System.out.println(pid);
```

```
String qty = jTable1.getValueAt(i, 4).toString();
```

```
String sellingPrice = jTable1.getValueAt(i, 5).toString();
```

```
// String buyingPrice = jTable1.getValueAt(i, 5).toString();
```

```
String mfd = jTable1.getValueAt(i, 6).toString();
```

```
// String exd = jTable1.getValueAt(i, 8).toString();
```

```
//      grn_id = id;
```

```
//      stock_id = ?
```

```
ResultSet rs3 = MySQL.search("SELECT * FROM `stock` WHERE `jewellery_id`='" +  
Integer.parseInt(pid) + "'AND `selling_price` = '" + sellingPrice + "' AND `mfd` = '" + mfd + "' ");
```

```
{
```

```
String stock_id;
```

```
if (rs3.next()) {
```

```
//update
```

```
stock_id = rs3.getString("id");
```



```

        String stock_qty = rs3.getString("quantity");
//        String stock_status = rs3.getString("Status");

        int updatedQty = Integer.parseInt(stock_qty) + Integer.parseInt(qty);

        MySql.iud("UPDATE `stock` SET `quantity`='" + updatedQty + "' WHERE `id`='" +
stock_id + "'");
//        MySql.iud("UPDATE `stock` SET `Status`='Unavailble' WHERE `quantity`='" +
updatedQty + "'");
//        System.out.println(updatedQty);

    } else {
        //insert
//        System.out.println(pid);

        MySql.iud("UPDATE `stock` SET `Status`='Unavailble' WHERE `jewellery_id`='" +
Integer.parseInt(pid) + "'");

        System.out.println(pid);

        MySql.iud("INSERT INTO `stock`(`jewellery_id`,`quantity`,`selling_price`,`mfd`)
VALUES('" + Integer.parseInt(pid) + "','" + qty + "','" + sellingPrice + "','" + mfd + "')");

        ResultSet rs4 = MySql.search("SELECT * FROM `stock` WHERE `jewellery_id`='" +
Integer.parseInt(pid) + "' AND `selling_price`='" + sellingPrice + "' AND `mfd`='" + mfd + "'");

        rs4.next();

        stock_id = rs4.getString("id");

    }

    // MySql.iud("INSERT INTO `invoice_item`(`stock_id`,`qty`,`invoice_id`) VALUES('"
+ Integer.parseInt(stock_id) + "','" + qty + "','" + Integer.parseInt(id) + "')");

```

```
    }  
}
```

```
resetFields();
```

```
//supplier
```

```
 jButton1.setEnabled(true);
```

```
 jButton1.setText("Select Customer");
```

```
 this.id.setText("none");
```

```
 jLabel6.setText("none");
```

```
 jLabel7.setText("none");
```

```
 jLabel8.setText("none");
```

```
//supplier
```

```
 JOptionPane.showMessageDialog(this, "New Invoice Created", "Success",  
JOptionPane.INFORMATION_MESSAGE);
```

```
//
```

```
//      String filePath = "src//reports//Invoice14.jrxml";
```

```
//      JasperReport jr = JasperCompileManager.compileReport(filePath);
```

```
////      InputStream stream = Invoice.class.getResourceAsStream(filePath);
```

```
////      JasperReport jr = JasperCompileManager.compileReport(stream);
```

```
//
```

```
////      JasperReport jr = JasperCompileManager.compileReport(filePath);
```

```
//      HashMap parameters = new HashMap();
```

```
//
//      //parameters.put("date", Date);
//      // parameters.put("grnid", uniqueId);
//
//      parameters.put("Parameter1", jComboBox1);
//      parameters.put("Parameter2", jTextField3);
//
//      TableModel tm = jTable1.getModel();
//      JTableModelDataSource dataSource = new JTableModelDataSource(tm);
////      JEmptyDataSource dataSource = new JEmptyDataSource();
//
//      JasperPrint jp = JasperFillManager.fillReport(jr, parameters, dataSource);
//      JasperViewer.viewReport(jp,false);
```

```
//payments
```

```
fulltotal.setText("0.00");
```

```
jTextField3.setText("");
```

```
jTextField3.setEditable(false);
```

```
jLabel24.setText("0.00");
```

```
jComboBox1.setSelectedIndex(0);
```

```
//payments
```

```
String filePath =
"C:\\Users\\DELL\\JaspersoftWorkspace\\MyReports\\Invoice14.jrxml";
JasperReport jr = JasperCompileManager.compileReport(filePath);
```

```

HashMap parameters = new HashMap();

//      JREmptyDataSource dataSource = new JREmptyDataSource();
TableModel tm = jTable1.getModel();
JRTableModelDataSource dataSource = new JRTableModelDataSource(tm);

JasperPrint jp = JasperFillManager.fillReport(jr, parameters, dataSource);

JasperViewer.viewReport(jp,false);

DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();
dtm.setRowCount(0);

//      ResultSet rs3 = MySql.search("SELECT * FROM `stock` WHERE `jewellery_id`='" +
Integer.parseInt(pid) + "'AND `selling_price` = '" + sellingPrice + "' AND `mfd` = '" + mfd + "' ");
//
//      if (rs3.next()) {
//          //update
//          stock_id = rs3.getString("id");
//          String stock_qty = rs3.getString("quantity");
//
//          int updatedQty = Integer.parseInt(stock_qty) + Integer.parseInt(qty);
//
//          MySql.iud("UPDATE `stock` SET `quantity`='" + updatedQty + "' WHERE `id` = '"
+ stock_id + "' ");

```

```

//
//      }

//MySQL.iud("UPDATE `stock` SET `Status`='Unavailble' WHERE `id`= '" + stock_id + "' ");

//      JOptionPane.showMessageDialog(this, "New Invoice Created", "Success",
JOptionPane.INFORMATION_MESSAGE);

//GRN item Insert & stock Insert or update
} catch (Exception e) {
    e.printStackTrace();
}

}

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see
    http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

```

```

        */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Invoice.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Invoice.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Invoice.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Invoice.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    }
}

//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {

```

```
        public void run() {  
            new Invoice().setVisible(true);  
        }  
    });  
}
```

```
// Variables declaration - do not modify  
private javax.swing.JLabel fulltotal;  
public javax.swing.JLabel id;  
public javax.swing.JButton jButton1;  
private javax.swing.JButton jButton2;  
private javax.swing.JButton jButton3;  
private javax.swing.JButton jButton5;  
private javax.swing.JComboBox<String> jComboBox1;  
private com.toedter.calendar.JDateChooser jDateChooser1;  
private javax.swing.JLabel jLabel1;  
public javax.swing.JLabel jLabel10;  
private javax.swing.JLabel jLabel11;  
public javax.swing.JLabel jLabel12;  
private javax.swing.JLabel jLabel13;  
public javax.swing.JLabel jLabel14;  
private javax.swing.JLabel jLabel15;  
public javax.swing.JLabel jLabel16;  
private javax.swing.JLabel jLabel17;  
private javax.swing.JLabel jLabel19;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel21;
```

```
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
public javax.swing.JLabel jLabel6;
public javax.swing.JLabel jLabel7;
public javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel5;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
public javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private com.k33ptoo.components.KGradientPanel kGradientPanel1;

// End of variables declaration
}
```



## Employee Report Code –

Employee Details									
ID	Name	User N...	Passwo...	NIC	Email	Contact...	Address	Catego...	Emp C...
1	maduka	maduka4...	4336	98456315...	Mduka@...	0728954363	no.32/ka...	Ring	reception
2	kasun sa...	kasun4@	55869	20015896...	kasun@...	0778658823	No.04,Col...	Neclous	Workers
3	dilshan	dilshan3@	589745	20016589...	dilshan...	0778965585	no.03,ka...	Ring	Workers
4	maduka	maduka4...	4336	98456315...	Mduka@...	0728954363	no.32/ka...	Ring	reception
5	kasu	kasu@	45569	56972585	kslskjsj...	0758996585	kandy ro...	Neclous	Workers

Print

```
package gui;
```

```
import java.sql.ResultSet;
```

```
import java.util.HashMap;
```

```
import java.util.Vector;
```

```
import javax.swing.table.DefaultTableModel;
```

```
import javax.swing.table.TableModel;
```

```

import model.MySql;
import net.sf.jasperreports.engine.JasperCompileManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.data.JRTableModelDataSource;
import net.sf.jasperreports.view.JasperViewer;

/**
 *
 * @author DELL
 */
public class Employee_report extends javax.swing.JFrame {

    public void loademployee() {

        try {
            ResultSet rs = MySql.search("SELECT
`employee`.`id`,`employee`.`name`,`employee`.`user_name`,`employee`.`password`,`employee`.`NIC`,`employee`.`email`,`employee`.`Contact_No`,`employee`.`Address`,`category`.`name` AS
`category`,`emp_category`.`name` AS `emp_category` FROM `employee` INNER JOIN `category`
ON `employee`.`category_id`=`category`.`id` INNER JOIN `emp_category` ON
`employee`.`emp_category_id`=`emp_category`.`id`;");

            //DefaultTableModel dtm = new DefaultTableModel();

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

            dtm.setRowCount(0);

            while (rs.next()) {

```

```

        Vector v = new Vector();
        v.add(rs.getString("id"));
        v.add(rs.getString("name"));
        v.add(rs.getString("user_name"));
        v.add(rs.getString("password"));
        v.add(rs.getString("NIC"));
        v.add(rs.getString("email"));
        v.add(rs.getString("Contact_No"));
        v.add(rs.getString("Address"));
        v.add(rs.getString("category"));
        v.add(rs.getString("emp_category"));
        dtm.addRow(v);

    }

    jTable1.setModel(dtm);

} catch (Exception e) {
    e.printStackTrace();
}

}

/**
 * Creates new form Employee_report
 */
public Employee_report() {

```

```

        initComponents();
        loademployee();
    }

    /**
     * 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() {

        jPanel1 = new javax.swing.JPanel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        jButton1 = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setResizable(false);

        jPanel1.setBackground(new java.awt.Color(255, 255, 255));

        jTable1.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 1, 1, 1, new
        java.awt.Color(255, 0, 0)));

        jTable1.setFont(new java.awt.Font("Dialog", 1, 11)); // NOI18N

```

```

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

    },
    new String [] {
        "ID", "Name", "User Name", "Password", "NIC", "Email", "Contact_No", "Address",
        "Category ", "Emp Category"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false, false, true, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});

jTable1.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jTable1MouseClicked(evt);
    }
});

jScrollPane1.setViewportViewView(jTable1);

jButton1.setText("Print");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        jButton1ActionPerformed(evt);
    }
});

jLabel1.setFont(new java.awt.Font("Elephant", 1, 18)); // NOI18N
jLabel1.setForeground(new java.awt.Color(153, 153, 153));
jLabel1.setText("Employee Details");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 682,
Short.MAX_VALUE)
        .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(241, 241, 241)
                        .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 180,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGap(254, 254, 254)
                            .addComponent(jLabel1)))
                        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addGap(10, 10, 10))
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addComponent(jLabel1)
                .addGap(10, 10, 10))
        )
    );
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 280,
Short.MAX_VALUE)
            .addGap(10, 10, 10)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
Short.MAX_VALUE)
            .addGap(10, 10, 10)
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
Short.MAX_VALUE)
            .addGap(10, 10, 10)
        )
    );

```

```

        .addGroup(jPanel1Layout.createSequentialGroup())
        .addGap(23, 23, 23)
        .addComponent(jLabel1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 33,
Short.MAX_VALUE)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 362,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(12, 12, 12)
        .addComponent(jButton1)
        .addContainerGap())
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

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

```

```
private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
  
}  
  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try {  
        String filePath = "C:\\\\Users\\\\DELL\\\\JaspersoftWorkspace\\\\MyReports\\\\Employee.jrxml";  
        JasperReport jr = JasperCompileManager.compileReport(filePath);  
  
        HashMap parameters = new HashMap();  
  
        //      JREmptyDataSource dataSource = new JREmptyDataSource();  
        TableModel tm = jTable1.getModel();  
        JRTableModelDataSource dataSource = new JRTableModelDataSource(tm);  
  
        JasperPrint jp = JasperFillManager.fillReport(jr, parameters, dataSource);  
  
        JasperViewer.viewReport(jp,false);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```



```

        this.dispose();

    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see
         * http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Windows".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

            java.util.logging.Logger.getLogger(Employee_report.class.getName()).log(java.util.logging.Level.
                SEVERE, null, ex);

        } catch (InstantiationException ex) {

```

```
java.util.logging.Logger.getLogger(Employee_report.class.getName()).log(java.util.logging.Level.  
SEVERE, null, ex);
```

```
    } catch (IllegalAccessException ex) {
```

```
java.util.logging.Logger.getLogger(Employee_report.class.getName()).log(java.util.logging.Level.  
SEVERE, null, ex);
```

```
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(Employee_report.class.getName()).log(java.util.logging.Level.  
SEVERE, null, ex);
```

```
    }
```

```
//</editor-fold>
```

```
/* Create and display the form */
```

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

```
    public void run() {
```

```
        new Employee_report().setVisible(true);
```

```
    }
```

```
});
```

```
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JButton jButton1;
```

```
private javax.swing.JLabel jLabel1;
```

```
private javax.swing.JPanel jPanel1;
```

```
private javax.swing.JScrollPane jScrollPane1;
```

```
private javax.swing.JTable jTable1;
```

```
// End of variables declaration
```

```
}
```

## Supplier Report Code –

Supplier Details				
ID	Name	Contact No	Email	Company Branch
1	Rocchana Rathnaya...	0769815988	Rochana@gmail.com	colombo branch
2	Thewindu Munnaa	0776358954	Thevindu@gmail. c...	colombo branch
3	Rocchana Rathnaya...	0769815988	Rochana@gmail.com	colombo branch
4	Rocchana Rathnaya...	0769815989	Rochana@gmail.com	colombo branch
5	Rocchana Rathnaya...	0769815988	Rochana@gmail.com	colombo branch
6	Tashila kumaraya	0713573654	tashilakumaraya@...	jewell kandy
7	dasun sanku	0783658923	dk12@gmailo.com	jewell kandy
8	Mendis soisa	0786589562	soisa@gmail.com	colombo branch
9	amal perera	0782365589	amal@gmail.com	colombo branch
10	mawan	0753269854	mawan8@	trinco branch

Print

```
package gui;
```

```
import java.sql.ResultSet;
```

```
import java.util.HashMap;
```

```
import java.util.Vector;
```

```
import javax.swing.table.DefaultTableModel;
```

```
import javax.swing.table.TableModel;
```

```

import model.MySql;
import net.sf.jasperreports.engine.JasperCompileManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.data.JRTableModelDataSource;
import net.sf.jasperreports.view.JasperViewer;

/**
 *
 * @author DELL
 */
public class Supplier_Report extends javax.swing.JFrame {

    public void loadsupplier() {

        try {

            ResultSet rs = MySql.search("SELECT
`supplier`.`id`,`supplier`.`name`,`supplier`.`contact_number`,`supplier`.`email`,`company_branch`.`name` AS `branch_name` FROM `supplier` INNER JOIN `company_branch` ON
`Supplier`.`company_branch_id`=`company_branch`.`id`");

            //DefaultTableModel dtm = new DefaultTableModel();

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

            dtm.setRowCount(0);

```

```

while (rs.next()) {

    Vector v = new Vector();
    v.add(rs.getString("id"));
    v.add(rs.getString("name"));
    v.add(rs.getString("contact_number"));
    v.add(rs.getString("email"));
    v.add(rs.getString("branch_name"));
    dtm.addRow(v);

}

jTable1.setModel(dtm);

} catch (Exception e) {
    e.printStackTrace();
}

}

/**
 * Creates new form Supplier_Report
 */
public Supplier_Report() {
    initComponents();

```

```

        loadsupplier();
    }

    /**
     * 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() {

        jPanel1 = new javax.swing.JPanel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        jButton1 = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setResizable(false);

        jPanel1.setBackground(new java.awt.Color(255, 255, 255));

        jTable1.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 1, 1, 1, new
java.awt.Color(255, 0, 0)));
        jTable1.setFont(new java.awt.Font("Dialog", 1, 11)); // NOI18N
        jTable1.setModel(new javax.swing.table.DefaultTableModel(

```

```

new Object [][] {

},

new String [] {
    "ID", "Name", "Contact No", "Email", "Company Branch"
}
) {
    boolean[] canEdit = new boolean [] {
        false, false, false, true, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});

jTable1.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jTable1MouseClicked(evt);
    }
});

jScrollPane1.setViewportViewView(jTable1);


jButton1.setText("Print");

jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
}

```

```
});
```

```
jLabel1.setFont(new java.awt.Font("Dialog", 1, 18)); // NOI18N
```

```
jLabel1.setForeground(new java.awt.Color(153, 153, 153));
```

```
jLabel1.setText("Supplier Details");
```

```
javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
```

```
jPanel1.setLayout(jPanel1Layout);
```

```
jPanel1Layout.setHorizontalGroup(
```

```
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING,  
javax.swing.GroupLayout.DEFAULT_SIZE, 634, Short.MAX_VALUE)
```

```
        .addGroup(jPanel1Layout.createSequentialGroup()
```

```
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
                .addGroup(jPanel1Layout.createSequentialGroup()
```

```
                    .addGap(245, 245, 245)
```

```
                    .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 142,  
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
                .addGroup(jPanel1Layout.createSequentialGroup()
```

```
                    .addGap(242, 242, 242)
```

```
                    .addComponent(jLabel1)))
```

```
            .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
```

```
);
```

```
jPanel1Layout.setVerticalGroup(
```

```
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(jPanel1Layout.createSequentialGroup()
```

```
            .addGap()
```



```

        .addComponent(jLabel1)

        .addGap(51, 51, 51)

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 291,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(31, 31, 31)

        .addComponent(jButton1)

        .addContainerGap(37, Short.MAX_VALUE))

    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 0, Short.MAX_VALUE))
    );

    pack();

    setLocationRelativeTo(null);
} // </editor-fold>

```

```
private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:
```

```
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:
```

```
    try {
```

```
        String filePath = "C:\\\\Users\\DELL\\JaspersoftWorkspace\\MyReports\\supplier.jrxml";
```

```
        JasperReport jr = JasperCompileManager.compileReport(filePath);
```

```
        HashMap parameters = new HashMap();
```

```
//        JREmptyDataSource dataSource = new JREmptyDataSource();
```

```
        TableModel tm = jTable1.getModel();
```

```
        JRTableModelDataSource dataSource = new JRTableModelDataSource(tm);
```

```
        JasperPrint jp = JasperFillManager.fillReport(jr, parameters, dataSource);
```

```
        JasperViewer.viewReport(jp,false);
```

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

```

        e.printStackTrace();
    }

    this.dispose();
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see
     * http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Windows".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Supplier_Report.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}

```

```

    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Supplier_Report.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);

    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Supplier_Report.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Supplier_Report.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);

    }
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Supplier_Report().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;

```

```
// End of variables declaration
}
```

### Stock Report Code –

Stock									
Stock id	Jewellery...	Category	Name	Caret No	Quantity	Buying P...	Selling_P...	Mfd	Status
67	7	Neclous	GOLD R...	24	2	6000.0	7000.0	2022-08-...	Availble
38	7	Neclous	GOLD R...	24	2	800.0	9000.0	2022-08-...	Availble
49	7	Neclous	GOLD R...	24	2	6000.0	7000.0	2022-08-...	Availble
55	5	Neclous	golden e...	25	4	40000.0	50000.0	2022-08-...	Unavailb...
17	5	Neclous	golden e...	25	8	60000.0	80000.0	2022-08-...	Unavailb...
26	1	Ring	golden n...	06	3	555.0	5555.0	2022-08-...	Unavailb...
27	1	Ring	golden n...	06	2	2222.0	50000.0	2022-08-...	Unavailb...
30	1	Ring	golden n...	06	2	60000.0	70000.0	2022-08-...	Unavailb...
35	1	Ring	golden n...	06	2	6000.0	7000.0	2022-08-...	Unavailb...
50	1	Ring	golden n...	06	2	2000.0	3000.0	2022-08-...	Unavailb...
66	1	Ring	golden n...	06	2	6000.0	7000.0	2022-08-...	Availble
19	1	Ring	golden n...	06	6	65000.0	65100.0	2022-08-...	Unavailb...

Print

```
package gui;
```

```
import java.sql.ResultSet;
```

```
import java.util.HashMap;
```

```
import java.util.Vector;
```

```
import javax.swing.table.DefaultTableModel;
```

```
import javax.swing.table.TableModel;
import model.MySql;
import net.sf.jasperreports.engine.JasperCompileManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.data.JRTableModelDataSource;
import net.sf.jasperreports.view.JasperViewer;
```

```
/**
```

```
*
```

```
* @author DELL
```

```
*/
```

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

```
    public void loadStock() {
```

```
        try {
```

```
            ResultSet rs = MySql.search("SELECT DISTINCT
`stock`.`id`,`stock`.`Status`,`jewellery`.`id`,`category`.`name`,`jewellery`.`caret_no`,`jewellery`.`name`,`stock`.`quantity`,`grn_item`.`buying_price`,`stock`.`selling_price`,`stock`.`mfd` FROM `stock`
INNER JOIN `grn_item` ON `grn_item`.`stock_id`=`stock`.`id` INNER JOIN `jewellery` ON
`stock`.`jewellery_id`=`jewellery`.`id` INNER JOIN `category` ON
`jewellery`.`category_id`=`category`.`id` ORDER BY `jewellery`.`name` ASC");
```

```
            DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();
```

```
            dtm.setRowCount(0);
```

```

while (rs.next()) {
    Vector v = new Vector();
    v.add(rs.getString("stock.id"));
    v.add(rs.getString("jewellery.id"));
    v.add(rs.getString("category.name"));
    v.add(rs.getString("jewellery.name"));
    v.add(rs.getString("jewellery.caret_no"));
    v.add(rs.getString("stock.quantity"));
    v.add(rs.getString("grn_item.buying_price"));
    v.add(rs.getString("stock.selling_price"));
    v.add(rs.getString("stock.mfd"));
    v.add(rs.getString("stock.Status"));

    dtm.addRow(v);
}
jTable1.setModel(dtm);

} catch (Exception e) {
    e.printStackTrace();
}

}

/**
 * Creates new form stock_report

```

```

*/
public stock_report() {
    initComponents();
    loadStock();
}

/**
 * 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() {

    jPanel1 = new javax.swing.JPanel();
    jButton1 = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

    jPanel1.setBackground(new java.awt.Color(255, 255, 255));

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

```



```
jButton1.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButton1ActionPerformed(evt);  
    }  
});
```

```
jLabel1.setFont(new java.awt.Font("Dubai Medium", 1, 18)); // NOI18N  
jLabel1.setForeground(new java.awt.Color(102, 102, 102));  
jLabel1.setText("Stock");
```

```
jTable1.setModel(new javax.swing.table.DefaultTableModel(  
    new Object [][] {  
  
    },  
    new String [] {  
        "Stock id", "Jewellery id", "Category", "Name", "Caret No", "Quantity", "Buying Price",  
"Selling_Price", "Mfd", "Status"  
    }  
){  
    boolean[] canEdit = new boolean [] {  
        false, false, false, false, false, false, false, false, false, false  
    };  
  
    public boolean isCellEditable(int rowIndex, int columnIndex) {  
        return canEdit [columnIndex];  
    }  
});
```

```

jTable1.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jTable1MouseClicked(evt);
    }
});
jScrollPane1.setViewportView(jTable1);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 793,
Short.MAX_VALUE)
        .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(352, 352, 352)
            .addComponent(jLabel1))
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(335, 335, 335)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 99,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(352, 352, 352)
                    .addComponent(jLabel1))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(335, 335, 335)
                    .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 99,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

```

```
.addContainerGap()
.addComponent(jLabel1)
.addGap(18, 18, 18)
.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 334,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(39, 39, 39)
.addComponent(jButton1)
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
);
```

```
pack();
setLocationRelativeTo(null);
} // </editor-fold>
```

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

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

```
try {
```

```
    String filePath = "C:\\Users\\DELL\\JaspersoftWorkspace\\MyReports\\stock.jrxml";
```

```
    JasperReport jr = JasperCompileManager.compileReport(filePath);
```

```
    HashMap parameters = new HashMap();
```

```
//    JREmptyDataSource dataSource = new JREmptyDataSource();
```

```
    TableModel tm = jTable1.getModel();
```

```
    JRTableModelDataSource dataSource = new JRTableModelDataSource(tm);
```

```
    JasperPrint jp = JasperFillManager.fillReport(jr, parameters, dataSource);
```

```
    JasperViewer.viewReport(jp,false);
```

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

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

```
}
```

```
this.dispose();
```

```
}
```

```
private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
```

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

```
////    int selectedRow = jTable1.getSelectedRow();
```

```

////    if(selectedRow!=-1) {
////
////        String buying_price = jTable1.getValueAt(selectedRow, 6).toString();
////        jLabel3.setText(buying_price);
//    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see
     * http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

```

```
java.util.logging.Logger.getLogger(stock_report.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
    } catch (InstantiationException ex) {
```

```
java.util.logging.Logger.getLogger(stock_report.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
    } catch (IllegalAccessException ex) {
```

```
java.util.logging.Logger.getLogger(stock_report.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(stock_report.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
    }
```

```
//</editor-fold>
```

```
/* Create and display the form */
```

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

```
    public void run() {
```

```
        new stock_report().setVisible(true);
```

```
    }
```

```
});
```

```
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JButton jButton1;
```

```
private javax.swing.JLabel jLabel1;
```

```
private javax.swing.JPanel jPanel1;
```

```

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

// End of variables declaration
}

```

## 9. Exchange Item

The screenshot shows the 'Jewelry Shop' application. The sidebar on the left contains the following menu items: Customer Registration, Supplier Registration, Employee Registration, Branch Registration, Owner Registration, Add Jewellery, Stock, Reports, Exchanged Item (highlighted), and Logout. The main content area is titled 'Exchanged Item' and contains a table with the following data:

ID	Date	NIC	Jewellery Name	Category
1	2022-08-09	962223345V	golden nec	Neclous
2	2022-08-09	987452485V	golden siver	Neclous

Below the table, there are input fields for 'Date', 'NIC', 'Jewellery Name' (a dropdown menu), and 'Category' (a dropdown menu). There are also 'Add' and 'Update' buttons.

**To load data into the table –**

```

public void loadexchange() {

    try {
        ResultSet rs = MySql.search("SELECT
`exchange_item`.`id`,`exchange_item`.`Date`,`exchange_item`.`NIC`,`jewellery`.`name` AS
`jewellery`,`category`.`name` AS `category` FROM `exchange_item` INNER JOIN `jewellery`

```

```
ON `exchange_item`.`jewellery_id`=`jewellery`.`id` INNER JOIN `category` ON  
`exchange_item`.`category_id`=`category`.`id`);
```

```
//DefaultTableModel dtm = new DefaultTableModel();  
DefaultTableModel dtm = (DefaultTableModel) jTable10.getModel();  
dtm.setRowCount(0);  
  
while (rs.next()) {  
  
    Vector v = new Vector();  
    v.add(rs.getString("id"));  
    v.add(rs.getString("Date"));  
    v.add(rs.getString("NIC"));  
    v.add(rs.getString("jewellery"));  
    v.add(rs.getString("category"));  
    dtm.addRow(v);  
  
}  
  
jTable10.setModel(dtm);  
  
} catch (Exception e) {  
    e.printStackTrace();  
}  
}
```

### **Table Mouse Clicked-**

```
private void jTable10MouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
  
    if(evt.getClickCount() == 1){  
        int r = jTable10.getSelectedRow();  
  
        if (r== -1) {  
            JOptionPane.showMessageDialog(this, "Please select a  
Jewellery","Warning",JOptionPane.WARNING_MESSAGE);  
        } else {
```



```

String Date = jTable10.getValueAt(r, 1).toString();
String NIC = jTable10.getValueAt(r, 2).toString();
String j_name = jTable10.getValueAt(r, 3).toString();
String category = jTable10.getValueAt(r, 4).toString();

jTextField33.setText(NIC);
//          jDateChooser1.setDate();
jComboBox19.setSelectedItem(category);
jComboBox18.setSelectedItem(j_name);
jButton12.setEnabled(false);

}

}
}

```

#### **Jwellery Name (ComboBox) –**

```

public void loadJwellerys() {

    try {

        ResultSet rs = MySql.search("SELECT * FROM `jewellery`");

        Vector v = new Vector();
        v.add("Select");

        while (rs.next()) {
            v.add(rs.getString("name"));
        }

        DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);
        jComboBox18.setModel(dcm);

    } catch (Exception e) {
        e.printStackTrace();
    }

}

```

### **Category (ComboBox) –**

```
public void loadcateg() {  
  
    try {  
  
        ResultSet rs = MySql.search("SELECT * FROM `category`");  
  
        Vector v = new Vector();  
        v.add("Select");  
  
        while (rs.next()) {  
            v.add(rs.getString("name"));  
        }  
  
        DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);  
        jComboBox19.setModel(dcm);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
}
```

### **Add button –**

```
private void jButton12ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    Date d = jDateChooser1.getDate();  
  
    String NIC = jTextField33.getText();  
    String JC = jComboBox19.getSelectedItem().toString();  
    String jname = jComboBox18.getSelectedItem().toString();  
  
    if (d == null) {  
        JOptionPane.showMessageDialog(this, "Please Select a Date", "warning",  
JOptionPane.WARNING_MESSAGE);
```

```

    } else if (d.after(new Date())) {
        JOptionPane.showMessageDialog(this, "Invalid MFD", "Warning",
JOptionPane.WARNING_MESSAGE);
    } else if (NIC.isEmpty()) {
        JOptionPane.showMessageDialog(this, "Please Enter NIC", "warning",
JOptionPane.WARNING_MESSAGE);
    } else if (jname.equals("Select")) {
        JOptionPane.showMessageDialog(this, "Please Select Jewellery", "warning",
JOptionPane.WARNING_MESSAGE);
    } else if (JC.equals("Select")) {
        JOptionPane.showMessageDialog(this, "Please Select Category", "warning",
JOptionPane.WARNING_MESSAGE);
    } else {

        SimpleDateFormat sdf2 = new SimpleDateFormat("yyyy-MM-dd");
        String dNow = sdf2.format(new Date());

        try {

            ResultSet rs1 = MySql.search("SELECT `id` FROM `category` WHERE `name` = '"
+ JC + "'");
            rs1.next();

            ResultSet rs2 = MySql.search("SELECT `id` FROM `jewellery` WHERE `name` = '" +
jname + "'");
            rs2.next();

            String j_id = rs1.getString("id");
            String e_id = rs2.getString("id");

            MySql.iud("INSERT INTO
`exchange_item`(`Date`,`NIC`,`jewellery_id`,`category_id`) VALUES ('" + dNow + "','" + NIC
+ "','" + Integer.parseInt(e_id) + "','" + Integer.parseInt(j_id) + ');");

            loadexchange();
            // jDateChooser1.setDate();
            jTextField33.setText("");
            jComboBox18.setSelectedIndex(0);
            jComboBox19.setSelectedIndex(0);

```

```

        JOptionPane.showMessageDialog(this, "New Employee Registered", "Success",
JOptionPane.INFORMATION_MESSAGE);
    } catch (Exception e) {
        e.printStackTrace();
    }

}

}

```

### Update Button -

```

private void jButton15ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    int r = jTable10.getSelectedRow();

    String id = jTable10.getValueAt(r, 0).toString();

    Date d = jDateChooser1.getDate();

    String NIC = jTextField33.getText();
    String JC = jComboBox19.getSelectedItem().toString();
    String jname = jComboBox18.getSelectedItem().toString();

    if (d == null) {
        JOptionPane.showMessageDialog(this, "Please Select a Date", "warning",
JOptionPane.WARNING_MESSAGE);
    } else if (d.after(new Date())) {
        JOptionPane.showMessageDialog(this, "Invalid MFD", "Warning",
JOptionPane.WARNING_MESSAGE);
    } else if (NIC.isEmpty()) {
        JOptionPane.showMessageDialog(this, "Please Enter NIC", "warning",
JOptionPane.WARNING_MESSAGE);
    } else if (jname.equals("Select")) {
        JOptionPane.showMessageDialog(this, "Please Select Jewellery", "warning",
JOptionPane.WARNING_MESSAGE);
    } else if (JC.equals("Select")) {

```

```

        JOptionPane.showMessageDialog(this, "Please Select Category", "warning",
JOptionPane.WARNING_MESSAGE);
    } else {

        SimpleDateFormat sdf2 = new SimpleDateFormat("yyyy-MM-dd");
        String dNow = sdf2.format(new Date());

        try {

            ResultSet rs1 = MySql.search("SELECT `id` FROM `category` WHERE `name` = '"
+ JC + "'");
            rs1.next();

            ResultSet rs2 = MySql.search("SELECT `id` FROM `jewellery` WHERE `name` = '"
+ jname + "'");
            rs2.next();

            String j_id = rs1.getString("id");
            String e_id = rs2.getString("id");

            MySql.iud("Update exchange_item set Date='" + dNow + "',NIC='" + NIC +
            "',jewellery_id='" + Integer.parseInt(e_id) + "',category_id='" + Integer.parseInt(j_id) + "'
            where id='" + id + "'");

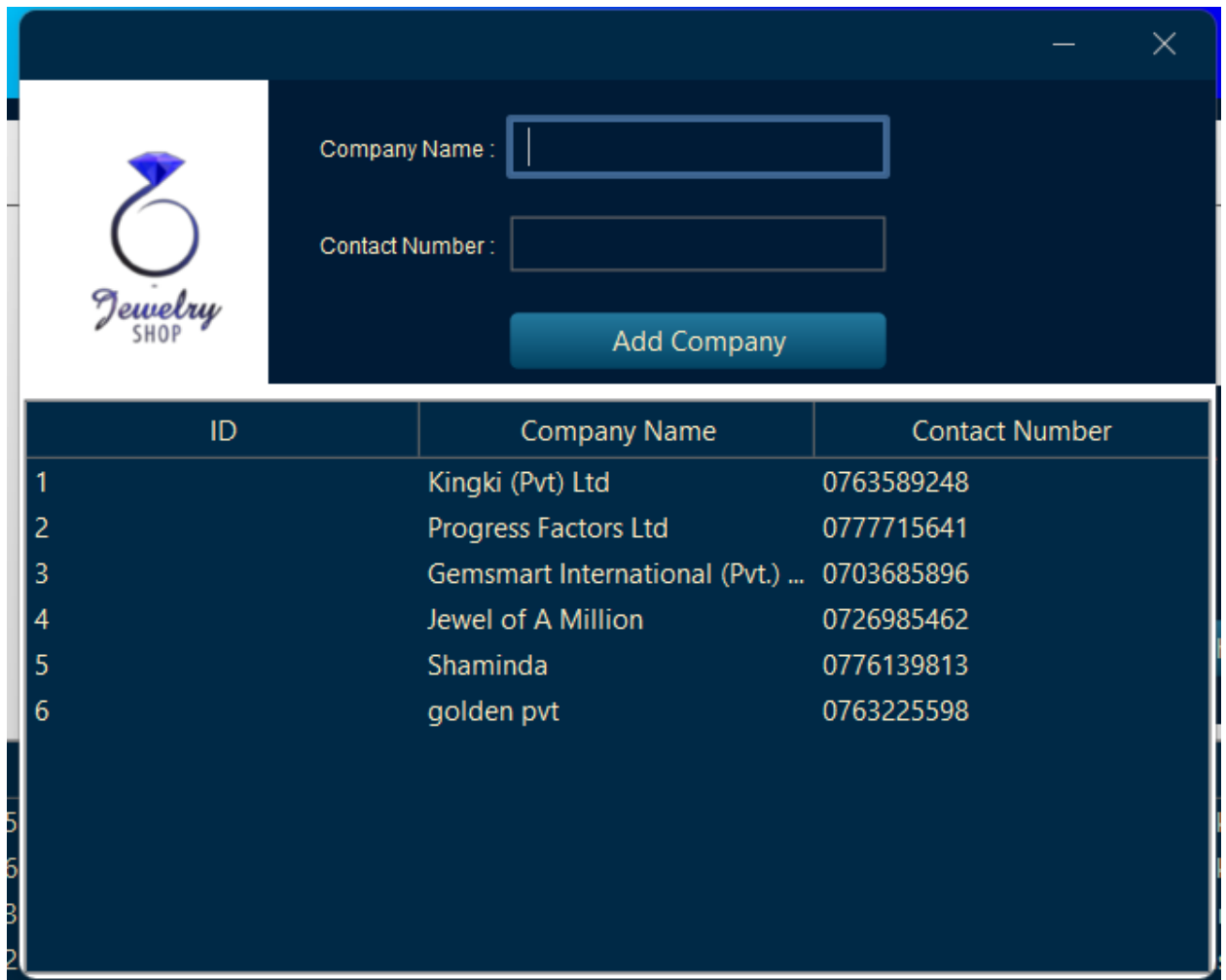
            loadexchange();
            // jDateChooser1.setDate();
            jTextField33.setText("");
            jComboBox18.setSelectedIndex(0);
            jComboBox19.setSelectedIndex(0);

            JOptionPane.showMessageDialog(this, "Update Succses", "Success",
JOptionPane.INFORMATION_MESSAGE);
        } catch (Exception e) {
            e.printStackTrace();
        }

    }
}

```

## 10. Company Registration



Company Name :

Contact Number :

Add Company

ID	Company Name	Contact Number
1	Kingki (Pvt) Ltd	0763589248
2	Progress Factors Ltd	0777715641
3	Gemsmart International (Pvt.) ...	0703685896
4	Jewel of A Million	0726985462
5	Shaminda	0776139813
6	golden pvt	0763225598

```
package gui;
```

```
import java.sql.ResultSet;
```

```
import java.util.Vector;
```

```
import java.util.regex.Pattern;
```

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

```
import javax.swing.table.DefaultTableModel;

import model.MySql;

/**
 *
 * @author DELL
 */
public class Company_Registration extends javax.swing.JFrame {

    public void loadCompanies(){

        try{

            ResultSet rs = MySql.search("SELECT * FROM `company` ORDER BY `id` ASC");
            DefaultTableModel dtm = (DefaultTableModel)jTable1.getModel();
            dtm.setRowCount(0);

            while (rs.next()) {
                Vector v = new Vector();
                v.add(rs.getString("id"));
                v.add(rs.getString("name"));
                v.add(rs.getString("contact_number"));
                dtm.addRow(v);
            }

        }catch(Exception e){
            e.printStackTrace();
        }
    }
}
```

```

    }

}

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

/**
 * 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() {

    jPanel1 = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jPanel2 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jTextField1 = new javax.swing.JTextField();

```



```

jLabel3 = new javax.swing.JLabel();
jTextField2 = new javax.swing.JTextField();
jButton1 = new javax.swing.JButton();
jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setResizable(false);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

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

    },
    new String [] {
        "ID", "Company Name", "Contact Number"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, true, true
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
jScrollPane1.setViewportViewView(jTable1);

```

```
jLabel2.setFont(new java.awt.Font("Dialog", 0, 12)); // NOI18N
```

```
jLabel2.setText("Company Name :");
```

```
jTextField1.addKeyListener(new java.awt.event.KeyAdapter() {
```

```
    public void keyTyped(java.awt.event.KeyEvent evt) {
```

```
        jTextField1KeyTyped(evt);
```

```
    }
```

```
});
```

```
jLabel3.setFont(new java.awt.Font("Dialog", 0, 12)); // NOI18N
```

```
jLabel3.setText("Contact Number :");
```

```
jTextField2.addActionListener(new java.awt.event.ActionListener() {
```

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

```
        jTextField2ActionPerformed(evt);
```

```
    }
```

```
});
```

```
jTextField2.addKeyListener(new java.awt.event.KeyAdapter() {
```

```
    public void keyTyped(java.awt.event.KeyEvent evt) {
```

```
        jTextField2KeyTyped(evt);
```

```
    }
```

```
});
```

```
jButton1.setText("Add Company");
```

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

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

```

        jButton1ActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(28, 28, 28)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)

            .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(jLabel2)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 210,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(jLabel3)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                    .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jTextField2))))
            .addGap(178, 178, 178)

        );
jPanel2Layout.setVerticalGroup(

```

```

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(19, 19, 19)

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel2)

            .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel3)

            .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)

            .addComponent(jButton1)

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

```

```

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/image/jewelry-logo-
design-template-cir_1.png"))); // NOI18N

```

```

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()

```

```

        .addGap(20, 20, 20)

        .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    );

    jPanel1Layout.setVerticalGroup(

        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup())

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

            .addGroup(jPanel1Layout.createSequentialGroup()

                .addGap(0, 0, Short.MAX_VALUE)

                .addComponent(jLabel1)))

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 319,
javax.swing.GroupLayout.PREFERRED_SIZE))

    );

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

    layout.setHorizontalGroup(

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

        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

);

layout.setVerticalGroup(

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

        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

);

```

```

pack();

setLocationRelativeTo(null);

} // </editor-fold>

```

```

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

    // TODO add your handling code here:

    String name = jTextField1.getText();

    String contact_number = jTextField2.getText();

    if(name.isEmpty()){

        JOptionPane.showMessageDialog(this, "Please enter Company
Name", "Warning", JOptionPane.WARNING_MESSAGE);

    }else if(!Pattern.compile("07[01245678][0-9]{7}").matcher(contact_number).matches()){

        JOptionPane.showMessageDialog(this, "Invalid Contact
Number", "Warning", JOptionPane.WARNING_MESSAGE);

    }else {

        try {

            ResultSet rs = MySql.search("SELECT * FROM `company` WHERE `name`='"+name+"'");

            if(rs.next()){

```

```
        JOptionPane.showMessageDialog(this, "Company already  
exist", "Warning", JOptionPane.WARNING_MESSAGE);
```

```
    }else{
```

```
        MySql.iud("INSERT INTO `company`(`name`,`contact_number`) VALUES('"+ name  
+"','"+contact_number+"')");
```

```
        JOptionPane.showMessageDialog(this, "New Company  
Added", "Success", JOptionPane.INFORMATION_MESSAGE);
```

```
        loadCompanies();
```

```
        jTextField1.setText("");
```

```
        jTextField2.setText("");
```

```
        jTextField1.grabFocus();
```

```
    }
```

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

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

```
    }
```

```
}
```

```
}
```

```
private void jTextField1KeyTyped(java.awt.event.KeyEvent evt) {
```

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

```
    char m = evt.getKeyChar();
```

```
    if(Character.isLetter(m) || Character.isWhitespace(m) || Character.isISOControl(m)){
```

```
        jTextField1.setEditable(true);
```

```
}else{

    jTextField1.setEditable(false);
}

}

private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jTextField2KeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:

    System.out.println("1");
    String mobile = jTextField2.getText();
    String text = mobile + evt.getKeyChar();

    if (text.length() == 1) {
        if (!text.equals("0")) {
            evt.consume();
        }
    } else if (text.length() == 2) {
        if (!text.equals("07")) {
            evt.consume();
        }
    }
}
```



```

    } else if (text.length() == 3) {
        if (!Pattern.compile("07[01245678]").matcher(text).matches()) {
            evt.consume();
        }
    } else if (text.length() <= 10) {
        if (!Pattern.compile("07[01245678][0-9]+").matcher(text).matches()) {
            evt.consume();
        }
    } else {
        evt.consume();
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see
     * http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Windows".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
            }
        }
    }
}

```

```
        break;
    }
}
} catch (ClassNotFoundException ex) {
```

```
java.util.logging.Logger.getLogger(Company_Registration.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
```

```
    } catch (InstantiationException ex) {
```

```
java.util.logging.Logger.getLogger(Company_Registration.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
```

```
    } catch (IllegalAccessException ex) {
```

```
java.util.logging.Logger.getLogger(Company_Registration.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
```

```
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(Company_Registration.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
```

```
    }
```

```
//</editor-fold>
```

```
/* Create and display the form */
```

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

```
    public void run() {
```

```
        new Company_Registration().setVisible(true);
```

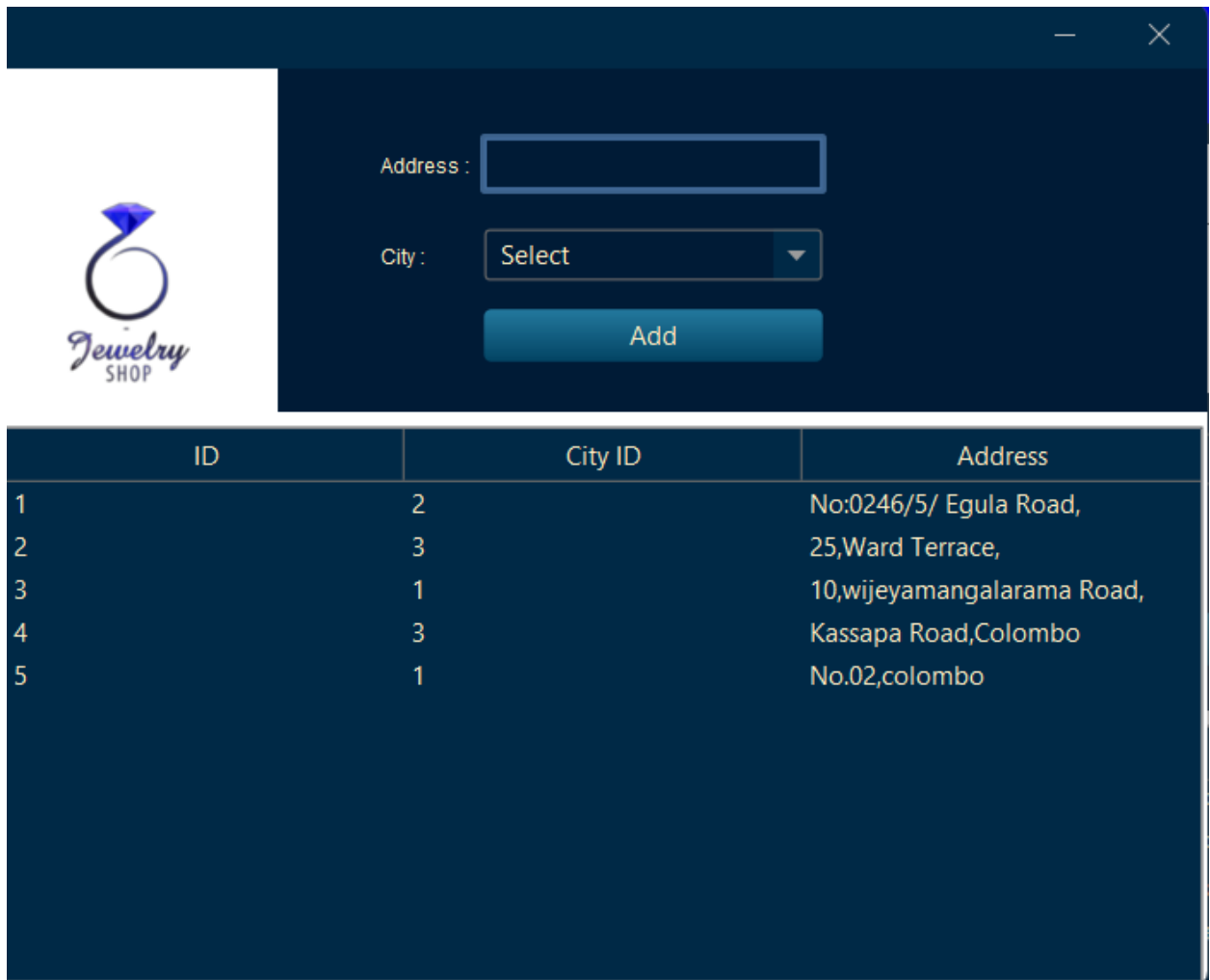
```
    }
```

```
});
```

```
}
```

```
// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
// End of variables declaration
}
```

## 11. Branch address



The screenshot shows a Java Swing window titled "Jewelry SHOP". On the left is a logo featuring a diamond ring and the text "Jewelry SHOP". On the right is a form with the following fields:

- "Address :" followed by a text input field.
- "City :" followed by a dropdown menu currently showing "Select".
- An "Add" button below the dropdown.

Below the form is a table with three columns: "ID", "City ID", and "Address". The table contains five rows of data:

ID	City ID	Address
1	2	No:0246/5/ Egula Road,
2	3	25,Ward Terrace,
3	1	10,wijeyamangalarama Road,
4	3	Kassapa Road,Colombo
5	1	No.02,colombo

```
package gui;
```

```
import java.sql.ResultSet;
import java.util.Vector;
import java.util.regex.Pattern;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import model.MySql;
```

```

/**
 *
 * @author DELL
 */
public class company_branch_address extends javax.swing.JFrame {

    public void loadbranchaddress(){

        try{

            ResultSet rs = MySql.search("SELECT * FROM `company_branch_address` ORDER
BY `id` ASC");
            DefaultTableModel dtm = (DefaultTableModel)jTable1.getModel();
            dtm.setRowCount(0);

            while (rs.next()) {
                Vector v = new Vector();
                v.add(rs.getString("id"));
                v.add(rs.getString("city_id"));
                v.add(rs.getString("Address"));
                dtm.addRow(v);
            }

        }catch(Exception e){
            e.printStackTrace();
        }

    }

    public void loadcity() {

        try {

            ResultSet rs = MySql.search("SELECT * FROM `city`");

            Vector v = new Vector();
            v.add("Select");

            while (rs.next()) {
                v.add(rs.getString("name"));
            }

        }

    }

}

```

```

    }

    DefaultComboBoxModel dcm = new DefaultComboBoxModel(v);
    city.setModel(dcm);

} catch (Exception e) {
    e.printStackTrace();
}

}

/**
 * Creates new form company_branch_address
 */
public company_branch_address() {
    initComponents();
    loadbranchaddress();
    loadcity();
}

/**
 * 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() {

    jPanel1 = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jPanel2 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    address = new javax.swing.JTextField();
    jLabel3 = new javax.swing.JLabel();
    city = new javax.swing.JComboBox<>();
    jButton1 = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

```

```

setResizable(false);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

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

        },
        new String [] {
            "ID", "City ID", "Address"
        }
    ) {
        boolean[] canEdit = new boolean [] {
            false, false, false
        };

        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    });
jScrollPane1.setViewportView(jTable1);

jLabel2.setFont(new java.awt.Font("Dialog", 0, 12)); // NOI18N
jLabel2.setText("Address :");

jLabel3.setFont(new java.awt.Font("Dialog", 0, 12)); // NOI18N
jLabel3.setText("City :");

jButton1.setText("Add");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(60, 60, 60)

```

```

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel2)
    .addComponent(jLabel3))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE, 202, Short.MAX_VALUE)
    .addComponent(address)
    .addComponent(city, 0, 202, Short.MAX_VALUE))
.addContainerGap(222, Short.MAX_VALUE)
);
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(38, 38, 38)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel2)
    .addComponent(address, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(city, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel3))
    .addGap(12, 12, 12)
    .addComponent(jButton1)
    .addContainerGap(27, Short.MAX_VALUE)
);

jLabel1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/image/jewelry-logo-design-template-cir_1.png"))); // NOI18N

```



```

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
            .addGap(20, 20, 20)
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 29,
Short.MAX_VALUE)
            .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        );
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
            .addPreferredGap(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 127,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 329,
javax.swing.GroupLayout.PREFERRED_SIZE))
                );

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );

```

```

        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );

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

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        String addr = address.getText();
        String cit = city.getSelectedItem().toString();

        if(addr.isEmpty()){
            JOptionPane.showMessageDialog(this, "Please enter address
", "Warning", JOptionPane.WARNING_MESSAGE);
        }else if(cit.isEmpty()){
        }else {
            try {
                ResultSet rs = MySql.search("SELECT * FROM `city` WHERE `name`='"+cit+"'");
                rs.next();

                String cit_id = rs.getString("id");

                MySql.iud("INSERT INTO `jewellery_shop`.`company_branch_address`
(`Address`, `city_id`) VALUES ('" + addr + "'," + Integer.parseInt(cit_id) + ");");
                JOptionPane.showMessageDialog(this, "New Address
Added", "Success", JOptionPane.INFORMATION_MESSAGE);
                loadbranchaddress();
                address.setText("");
                city.setSelectedIndex(0);
                JOptionPane.showMessageDialog(this, "Please Select
City", "Warning", JOptionPane.WARNING_MESSAGE);
                address.grabFocus();

            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }

```

```

    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional)
">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Windows".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(company_branch_address.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(company_branch_address.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(company_branch_address.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(company_branch_address.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

```

```
/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new company_branch_address().setVisible(true);
    }
});
}
```

```
// Variables declaration - do not modify
private javax.swing.JTextField address;
private javax.swing.JComboBox<String> city;
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
// End of variables declaration
}
```