

# JAVA SWING BASED – Mill Industry Management- SQL CONNECTIVITY USING JDBC

A Report Submitted in partial fulfilment of  
the Requirements for the award of the  
Degree of  
BACHELOR OF ENGINEERING IN  
INFORMATION TECHNOLOGY

By

B.Manikantaviswas <1602-22-737-089>  
Under the guidance of Ms Soume Sanyal



Department of Information Technology  
Vasavi College of Engineering (Autonomous)  
(Affiliated to Osmania University)  
Ibrahimbagh, Hyderabad-31  
2022-2023

## BONAFIDE CERTIFICATE

This is to certify that this project report titled  
‘Mill Industry Management’  
is a project work of B.Manikantaviswas  
bearing roll no. 1602-22-737-089 who carried  
out this project under my supervision in the IV  
semester of the academic year 2023- 2024

Signature External  
Examiner

Signature Internal  
Examiner

## CONTENTS:

- 1.Problem Statement
- 2.Abstract
- 3.Design Requirements
- 4.ER Diagram
- 5.DDL Commands
- 6.DML Commands
- 7.Implementation
- 8.Output
- 9.Result
- 10.Discussion and Future Work
- 11.References

## PROBLEM STATEMENT:

The Mill Industry Management System is designed to efficiently manage operations within a mill, including raw materials, products, inventory, customer orders, employees, suppliers, sales, and expenses. The system tracks raw material availability, product details, inventory levels, customer information, orders, employee data, shift timings, supplier contacts, sales transactions, and expenses. Developed using Oracle SQL, it ensures data integrity, performance, and scalability, providing a comprehensive solution for managing mill operations and supporting future growth.

## ABSTRACT

The Mill Industry Management System is a comprehensive database solution designed to streamline and manage various operations within a mill. Utilizing Oracle SQL, it handles raw materials, products, inventory, customer orders, employees, suppliers, sales, and expenses. The system ensures data integrity, efficiency, and scalability, offering robust tracking and management of all key operational aspects. This project aims to enhance the overall productivity and organization of mill industry processes through an integrated and efficient database management approach.

# Design Requirements:

List of tables, its attributes and their domains:

## 1. \*RawMaterials Table:\*

- \*raw\_material\_id:\* Unique identifier for each raw material.
- \*material\_name:\* Name of the raw material.
- \*quantity\_available:\* Quantity of raw material available.
- \*unit\_price:\* Price per unit of the raw material.
- \*supplier\_id:\* Identifier linking to the supplier providing the raw material.

## 2. \*Products Table:\*

- \*product\_id:\* Unique identifier for each product.
- \*product\_name:\* Name of the product.
- \*unit\_price:\* Price per unit of the product.
- \*category:\* Category to which the product belongs.

## 3. \*Inventory Table:\*

- \*inventory\_id:\* Unique identifier for the inventory record.
- \*product\_id:\* Identifier linking to the product in the inventory.
- \*quantity\_available:\* Quantity of the product available in inventory.
- \*last\_restock\_date:\* Date when the product was last restocked.

## 4. \*Customers Table:\*

- \*customer\_id:\* Unique identifier for each customer.
- \*customer\_name:\* Name of the customer.
- \*contact\_info:\* Contact information of the customer.

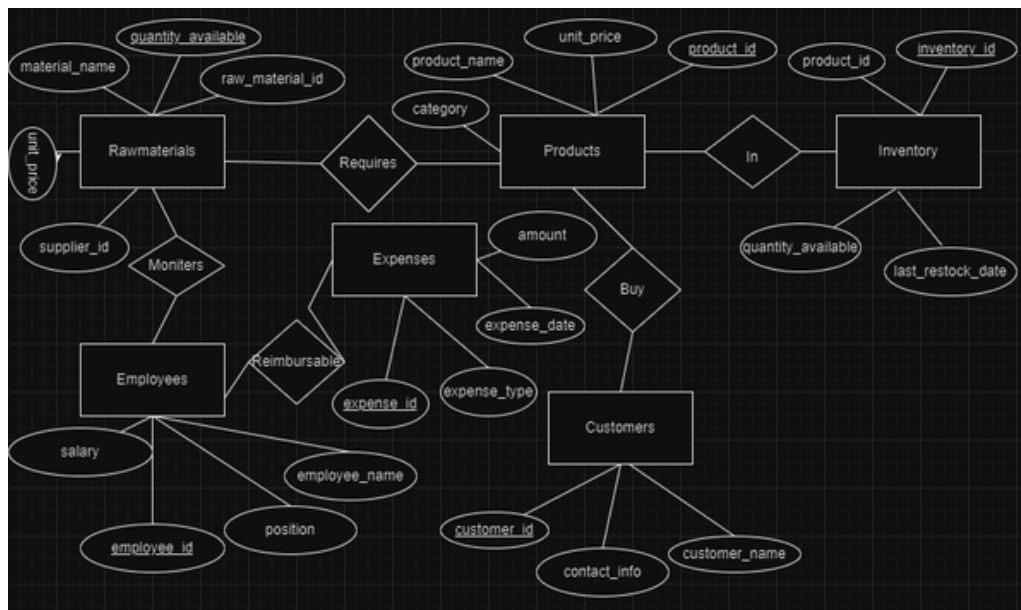
## 5. \*Employees Table:\*

- \*employee\_id:\* Unique identifier for each employee.
- \*employee\_name:\* Name of the employee.
- \*position:\* Position held by the employee.
- \*salary:\* Salary of the employee.

## 6. \*Expenses Table:\*

- \*expense\_id:\* Unique identifier for each expense.
- \*expense\_type:\* Type of expense (e.g., Utilities, Maintenance, Salaries).
- \*amount:\* Amount of the expense.
- \*expense\_date:\* Date when the expense was incurred.

## ER DIAGRAM:



## MAPPING CARDINALITIES

In the ER diagram, the relationship between \*Products\* and \*Inventory\* is One-to-Many, where each product can be referenced by multiple inventory records, indicating that a single product can be stored in various inventory locations or at different times. The \*RawMaterials\*, \*\*Employees, \*\*Expenses, and \*\*Customers\* tables do not have direct relationships with each other or with the \*Products\* and \*Inventory\* tables within this context, making them standalone entities. This design helps in maintaining clear and distinct records for each functional area of the mill industry, ensuring efficient management of raw materials, products, inventory, employee data, expenses, and customer information.

## DDL COMMANDS:

1.Creating Table for Rawmaterials:

QUERY:

```
CREATE TABLE RawMaterials (  
    raw_material_id INT PRIMARY KEY,  
    material_name VARCHAR(100),  
    quantity_available INT,  
    unit_price DECIMAL(10, 2),  
    supplier_id INT  
);
```



```
SQL> CREATE TABLE RawMaterials (
2   raw_material_id INT PRIMARY KEY,
3   material_name VARCHAR(100),
4   quantity_available INT,
5   unit_price DECIMAL(10, 2),
6   supplier_id INT
7 );
```

Table created.

```
SQL> desc rawmaterials;
```

Name	Null?	Type
RAW_MATERIAL_ID	NOT NULL	NUMBER(38)
MATERIAL_NAME		VARCHAR2(100)
QUANTITY_AVAILABLE		NUMBER(38)
UNIT_PRICE		NUMBER(10,2)
SUPPLIER_ID		NUMBER(38)

2.Creating Table for Products:

QUERY:

```
CREATE TABLE Products (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    unit_price DECIMAL(10, 2),
    category VARCHAR(50)
);
```

```
SQL> CREATE TABLE Products (
2   product_id INT PRIMARY KEY,
3   product_name VARCHAR(100),
4   unit_price DECIMAL(10, 2),
5   category VARCHAR(50)
6 );
```

Table created.

```
SQL> desc products;
```

Name	Null?	Type
PRODUCT_ID	NOT NULL	NUMBER(38)
PRODUCT_NAME		VARCHAR2(100)
UNIT_PRICE		NUMBER(10,2)
CATEGORY		VARCHAR2(50)

3.Creating Table for Inventory:

QUERY:

```
CREATE TABLE Inventory (
    inventory_id INT PRIMARY KEY,
    product_id INT,
    quantity_available INT,
    last_restock_date DATE
);
```

```
SQL> CREATE TABLE Inventory (
  2     inventory_id INT PRIMARY KEY,
  3     product_id INT,
  4     quantity_available INT,
  5     last_restock_date DATE
  6 );
```

Table created.

```
SQL> desc inventory;
```

Name	Null?	Type
INVENTORY_ID	NOT NULL	NUMBER(38)
PRODUCT_ID		NUMBER(38)
QUANTITY_AVAILABLE		NUMBER(38)
LAST_RESTOCK_DATE		DATE

4.Creating Table for  
Employees:  
QUERY:

```
CREATE TABLE Employees (
  employee_id INT PRIMARY KEY,
  employee_name VARCHAR(100),
  position VARCHAR(100),
  salary DECIMAL(10, 2)
);
```

```
SQL> CREATE TABLE Employees (
  2     employee_id INT PRIMARY KEY,
  3     employee_name VARCHAR(100),
  4     position VARCHAR(100),
  5     salary DECIMAL(10, 2)
  6 );
```

Table created.

```
SQL> desc employees;
```

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(38)
EMPLOYEE_NAME		VARCHAR2(100)
POSITION		VARCHAR2(100)
SALARY		NUMBER(10, 2)

5.Creating Table for Expenses:  
QUERY:

```
CREATE TABLE Expenses (
  expense_id INT PRIMARY KEY,
  expense_type VARCHAR(100),
  amount DECIMAL(10, 2),
  expense_date DATE
);
```

```
SQL> CREATE TABLE Expenses (
2     expense_id INT PRIMARY KEY,
3     expense_type VARCHAR(100),
4     amount DECIMAL(10, 2),
5     expense_date DATE
6 );
```

Table created.

```
SQL> desc expenses;
```

Name	Null?	Type
EXPENSE_ID	NOT NULL	NUMBER(38)
EXPENSE_TYPE		VARCHAR2(100)
AMOUNT		NUMBER(10,2)
EXPENSE_DATE		DATE

## 6.Creating Table for Customers:

QUERY:

```
CREATE TABLE Customers (
    customer_id INT PRIMARY KEY,
    customer_name VARCHAR(100),
    contact_info VARCHAR(100)
);
```

```
SQL> CREATE TABLE Customers (
2     customer_id INT PRIMARY KEY,
3     customer_name VARCHAR(100),
4     contact_info VARCHAR(100)
5 );
```

Table created.

```
SQL> desc customers;
```

Name	Null?	Type
CUSTOMER_ID	NOT NULL	NUMBER(38)
CUSTOMER_NAME		VARCHAR2(100)
CONTACT_INFO		VARCHAR2(100)

# DML COMMANDS:

1.Insert values into Rawmaterials:

QUERY:

```
INSERT INTO RawMaterials (raw_material_id, material_name,  
quantity_available, unit_price, supplier_id)  
VALUES (1, 'Iron', 1000, 10.50, 1);
```

```
INSERT INTO RawMaterials (raw_material_id, material_name,  
quantity_available, unit_price, supplier_id)  
VALUES (2, 'Steel', 800, 15.75, 2);
```

```
INSERT INTO RawMaterials (raw_material_id, material_name,  
quantity_available, unit_price, supplier_id)  
VALUES (3, 'Aluminum', 1200, 8.25, 3);
```

```
SQL> INSERT INTO RawMaterials (raw_material_id, material_name, quantity_available, unit_price, supplier_id)  
2 VALUES (1, 'Iron', 1000, 10.50, 1);  
  
1 row created.  
  
SQL> INSERT INTO RawMaterials (raw_material_id, material_name, quantity_available, unit_price, supplier_id)  
2 VALUES (2, 'Steel', 800, 15.75, 2);  
  
1 row created.  
  
SQL> INSERT INTO RawMaterials (raw_material_id, material_name, quantity_available, unit_price, supplier_id)  
2 VALUES (3, 'Aluminum', 1200, 8.25, 3);  
  
1 row created.
```

2.Insert values into Products:

QUERY:

```
INSERT INTO Products (product_id, product_name,  
unit_price, category)  
VALUES (1, 'Chair', 25.99, 'Furniture');
```

```
INSERT INTO Products (product_id, product_name,  
unit_price, category)  
VALUES (2, 'Table', 89.50, 'Furniture');
```

```
INSERT INTO Products (product_id, product_name,  
unit_price, category)  
VALUES (3, 'Lamp', 15.75, 'Lighting');
```

```

SQL> INSERT INTO Products (product_id, product_name, unit_price, category)
  2 VALUES (1, 'Chair', 25.99, 'Furniture');

1 row created.

SQL>
SQL> INSERT INTO Products (product_id, product_name, unit_price, category)
  2 VALUES (2, 'Table', 89.50, 'Furniture');

1 row created.

SQL>
SQL> INSERT INTO Products (product_id, product_name, unit_price, category)
  2 VALUES (3, 'Lamp', 15.75, 'Lighting');

1 row created.

```

### 3.Inserting values into Inventory:

QUERY:

```

INSERT INTO Inventory (inventory_id,
product_id, quantity_available, last_restock_date)
VALUES (1, 1, 50, TO_DATE('2024-05-12',
'YYYY-MM-DD'));

```

```

INSERT INTO Inventory (inventory_id,
product_id, quantity_available, last_restock_date)
VALUES (2, 2, 20, TO_DATE('2024-05-11',
'YYYY-MM-DD'));

```

```

INSERT INTO Inventory (inventory_id,
product_id, quantity_available, last_restock_date)
VALUES (3, 3, 100, TO_DATE('2024-05-10',
'YYYY-MM-DD'));

```

```

SQL> INSERT INTO Inventory (inventory_id, product_id, quantity_available, last_restock_date)
  2 VALUES (1, 1, 50, TO_DATE('2024-05-12', 'YYYY-MM-DD'));

1 row created.

SQL>
SQL> INSERT INTO Inventory (inventory_id, product_id, quantity_available, last_restock_date)
  2 VALUES (2, 2, 20, TO_DATE('2024-05-11', 'YYYY-MM-DD'));

1 row created.

SQL>
SQL> INSERT INTO Inventory (inventory_id, product_id, quantity_available, last_restock_date)
  2 VALUES (3, 3, 100, TO_DATE('2024-05-10', 'YYYY-MM-DD'));

1 row created.

```

#### 4.Insert values into Employees:

QUERY:

```
INSERT INTO Employees (employee_id,  
employee_name, position, salary)  
VALUES (101, 'John Doe', 'Manager', 50000);
```

```
INSERT INTO Employees (employee_id,  
employee_name, position, salary)  
VALUES (102, 'Jane Smith', 'Supervisor', 40000);
```

```
INSERT INTO Employees (employee_id,  
employee_name, position, salary)  
VALUES (103, 'Michael Johnson', 'Worker', 30000);
```

```
SQL> INSERT INTO Employees (employee_id, employee_name, position, salary)  
2 VALUES (101, 'John Doe', 'Manager', 50000);  
  
1 row created.  
  
SQL>  
SQL> INSERT INTO Employees (employee_id, employee_name, position, salary)  
2 VALUES (102, 'Jane Smith', 'Supervisor', 40000);  
  
1 row created.  
  
SQL>  
SQL> INSERT INTO Employees (employee_id, employee_name, position, salary)  
2 VALUES (103, 'Michael Johnson', 'Worker', 30000);  
  
1 row created.
```

#### 5.Inserting values into Expenses:

QUERY:

```
INSERT INTO Expenses (expense_id, expense_type, amount, expense_date)  
VALUES (1, 'Utilities', 1500.00, TO_DATE('2024-05-12', 'YYYY-MM-DD'));
```

```
INSERT INTO Expenses (expense_id, expense_type, amount, expense_date)  
VALUES (2, 'Maintenance', 800.00, TO_DATE('2024-05-11', 'YYYY-MM-DD'));
```

```
INSERT INTO Expenses (expense_id, expense_type, amount, expense_date)  
VALUES (3, 'Salaries', 12000.00, TO_DATE('2024-05-10', 'YYYY-MM-DD'));
```

```

SQL> INSERT INTO Expenses (expense_id, expense_type, amount, expense_date)
  2 VALUES (1, 'Utilities', 1500.00, TO_DATE('2024-05-12', 'YYYY-MM-DD'));

1 row created.

SQL>
SQL> INSERT INTO Expenses (expense_id, expense_type, amount, expense_date)
  2 VALUES (2, 'Maintenance', 800.00, TO_DATE('2024-05-11', 'YYYY-MM-DD'));

1 row created.

SQL>
SQL> INSERT INTO Expenses (expense_id, expense_type, amount, expense_date)
  2 VALUES (3, 'Salaries', 12000.00, TO_DATE('2024-05-10', 'YYYY-MM-DD'));

1 row created.

```

## 6.Inserting values into Customers:

QUERY:

```

INSERT INTO Customers (customer_id, customer_name, contact_info)
VALUES (1, 'ABC Corporation', 'abc@corp.com');

```

```

INSERT INTO Customers (customer_id, customer_name, contact_info)
VALUES (2, 'XYZ Ltd.', 'xyz@ltd.com');

```

```

INSERT INTO Customers (customer_id, customer_name, contact_info)
VALUES (3, '123 Industries', 'info@123.com');

```

```

SQL> INSERT INTO Customers (customer_id, customer_name, contact_info)
  2 VALUES (1, 'ABC Corporation', 'abc@corp.com');

1 row created.

SQL>
SQL> INSERT INTO Customers (customer_id, customer_name, contact_info)
  2 VALUES (2, 'XYZ Ltd.', 'xyz@ltd.com');

1 row created.

SQL>
SQL> INSERT INTO Customers (customer_id, customer_name, contact_info)
  2 VALUES (3, '123 Industries', 'info@123.com');

1 row created.

```

# IMPLEMENTATION

**JAVA-SQL CONNECTIVITY USING JDBC:** Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

## APPLICATION CODE:

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

    public Dashboard() {
        initComponents();
    }

    private void b1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        this.dispose();
        RawMaterials r=new RawMaterials();
        r.show();
    }

    private void b2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        this.dispose();
        Products p=new Products();
        p.show();
    }

    private void b3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        this.dispose();
        Inventory i=new Inventory();
        i.show();
    }

    private void b6ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        this.dispose();
        Expenses e =new Expenses();
        e.show();
    }

}
```



```

private void b4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.dispose();
    Customers c=new Customers();
    c.show();
}
private void b9ActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
    Expenses e=new Expenses();
    e.show();
}
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(Dashboard.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Dashboard.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Dashboard.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Dashboard.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 Dashboard().setVisible(true);
    }
});
}

```

```

// Variables declaration - do not modify
private javax.swing.JButton b1;
private javax.swing.JButton b2;
private javax.swing.JButton b3;
private javax.swing.JButton b4;
private javax.swing.JButton b6;
private javax.swing.JButton b9;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
// End of variables declaration
}

import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

public class LoginPage extends javax.swing.JFrame {

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

    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        username = new javax.swing.JTextField();
        password = new javax.swing.JPasswordField();
        jButton = new javax.swing.JButton();
        jLabel4 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        sign = new javax.swing.JButton();
        jLabel5 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
        jLabel1.setForeground(java.awt.Color.white);
        jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel1.setText("LOGIN");
        getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 20, 112, -1));

```

```

jLabel2.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
jLabel2.setForeground(java.awt.Color.white);
jLabel2.setText("USERNAME");
getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 70, 94, 27));
getContentPane().add(username, new org.netbeans.lib.awtextra.AbsoluteConstraints(220, 70, 161, 30));

password.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        passwordActionPerformed(evt);
    }
});
getContentPane().add(password, new org.netbeans.lib.awtextra.AbsoluteConstraints(220, 130, 161,
27));

jButton.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
jButton.setText("LOGIN");
jButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonActionPerformed(evt);
    }
});xt
getContentPane().add(jButton, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 180, 99, 31));

jLabel4.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
jLabel4.setForeground(java.awt.Color.white);
jLabel4.setText("PASSWORD");
getContentPane().add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 130, 94, 27));

jLabel3.setForeground(java.awt.Color.white);
jLabel3.setText("Don't have account?");
getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(140, 220, 129, -1));

sign.setText("Sign Up");
sign.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        signActionPerformed(evt);
    }
});
getContentPane().add(sign, new org.netbeans.lib.awtextra.AbsoluteConstraints(160, 250, -1, -1));

jLabel5.setIcon(new javax.swing.ImageIcon("C:\\Users\\DELL-PC\\OneDrive\\Desktop\\SQL
Project\\Screenshot 2024-05-16 121336.png")); // NOI18N
getContentPane().add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 430, 280));

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

```

```

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

    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "mani", "vasavi");

        String usr = username.getText();
        String pwd = new String(password.getPassword());

        String sql = "SELECT * FROM admin WHERE id = ? AND pass = ?";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setString(1, usr);
        pstmt.setString(2, pwd);
        ResultSet ra = pstmt.executeQuery();
        if (ra.next()) {
            this.dispose();
            Dashboard dsh = new Dashboard();
            dsh.show();
        } else {
            JOptionPane.showMessageDialog(this, "Username or password is incorrect");
        }

        ra.close();
        pstmt.close();
        con.close();
    } catch (ClassNotFoundException e) {
        System.out.print("Class not found");
    } catch (SQLException ex) {
        Logger.getLogger(LoginPage.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void signActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.dispose();
    Signup s=new Signup();
    s.show();
}

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

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

public static void main(String args[]) {

```

```

private void signActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.dispose();
    Signup s=new Signup();
    s.show();

}
private void passwordActionPerformed(java.awt.event.ActionEvent evt) {

public static void main(String args[]) {

    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(LoginPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(LoginPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(LoginPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(LoginPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater() -> {
    new LoginPage().setVisible(true);
});
}
// Variables declaration - do not modify
private javax.swing.JButton jButton;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JPasswordField password;
private javax.swing.JButton sign;
private javax.swing.JTextField username;
// End of variables declaration
}

```

```

import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
/**
 *
 * @author DELL-PC
 */
public class RawMaterials extends javax.swing.JFrame {

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

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        fetch = new javax.swing.JButton();
        back = new javax.swing.JButton();
        jButton1 = new javax.swing.JButton();
        id = new javax.swing.JTextField();
        quantity = new javax.swing.JTextField();
        sid = new javax.swing.JTextField();
        name = new javax.swing.JTextField();
        price = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jTable1.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null, null},
                {null, null, null, null, null},
                {null, null, null, null, null},
                {null, null, null, null, null}
            },
            new String [] {
                "ID", "Name", "Quantity", "price", "Supplier_id"
            }
        ));
        jScrollPane1.setViewportView(jTable1);

        fetch.setText("FETCH");

```

```
fetch.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        fetchActionPerformed(evt);  
    }  
});  
  
back.setText("BACK");  
back.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        backActionPerformed(evt);  
    }  
});  
  
jButton1.setText("ADD");  
jButton1.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButton1ActionPerformed(evt);  
    }  
});  
  
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());  
getContentPane().setLayout(layout);  
layout.setHorizontalGroup(  
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(layout.createSequentialGroup()  
            .addGap(10, 10, 10)  
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 593,  
Short.MAX_VALUE)  
            .addGap(10, 10, 10)  
            .addGroup(layout.createSequentialGroup()  
                .addGap(10, 10, 10)  
                .addComponent(id, javax.swing.GroupLayout.PREFERRED_SIZE, 90,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
                .addComponent(name, javax.swing.GroupLayout.PREFERRED_SIZE, 90,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
                .addComponent(quantity, javax.swing.GroupLayout.PREFERRED_SIZE, 90,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
                .addComponent(price, javax.swing.GroupLayout.PREFERRED_SIZE, 90,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)  
                .addComponent(sid, javax.swing.GroupLayout.PREFERRED_SIZE, 90,  
javax.swing.GroupLayout.PREFERRED_SIZE)  
                .addGap(10, 10, 10)  
            )  
        )  
        .addGroup(layout.createSequentialGroup()
```

```

.addGroup(layout.createSequentialGroup()
    .addGap(56, 56, 56)
    .addComponent(fetch)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(back)
    .addGap(54, 54, 54))
.addGroup(layout.createSequentialGroup()
    .addGap(261, 261, 261)
    .addComponent(jButton1)
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 182,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(id, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(quantity, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(sid, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(name, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(price, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addComponent(jButton1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(fetch)
            .addComponent(back))
        .addContainerGap(8, Short.MAX_VALUE))
    );

pack();
}

private void backActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.dispose();
    Dashboard dsh=new Dashboard();
    dsh.show();
}
private void fetchActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    DefaultTableModel model= (DefaultTableModel)jTable1.getModel();

    String q="select * from rawmaterials";
    try{

```



```

Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "mani",
"vasavi");
Statement smt=con.createStatement();
ResultSet ra=smt.executeQuery(q);
while(ra.next()){
    int ID=ra.getInt("raw_material_id");
    String Name=ra.getString("material_name");
    int Quantity=ra.getInt("quantity_available");
    int Price=ra.getInt("unit_price");
    int Supplier_id=ra.getInt("supplier_id");
    model.addRow(new Object[] {ID,Name,Quantity,Price,Supplier_id});
}
} catch (ClassNotFoundException e){
    System.out.print(e);
} catch (SQLException ex) {
    Logger.getLogger(RawMaterials.class.getName()).log(Level.SEVERE, null, ex);
}
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
try {
    int Id = Integer.parseInt(id.getText());
    String Name = name.getText();
    int Quant = Integer.parseInt(quantity.getText());
    int pri = Integer.parseInt(price.getText());
    int si = Integer.parseInt(sid.getText());
    String q = "INSERT INTO rawmaterials (raw_material_id, material_name,
quantity_available,unit_price, supplier_id) VALUES (?, ?, ?, ?, ?)";
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "mani",
"vasavi");
    con.setAutoCommit(false);
    PreparedStatement pstmt = con.prepareStatement(q);
    pstmt.setInt(1, Id);
    pstmt.setString(2, Name);
    pstmt.setInt(3, Quant);
    pstmt.setInt(4, pri);
    pstmt.setInt(5, si);
    int rowsAffected = pstmt.executeUpdate();
    if (rowsAffected > 0) {
        con.commit(); // Commit the transaction
        JOptionPane.showMessageDialog(this, "Record inserted successfully");
    } else {
        JOptionPane.showMessageDialog(this, "Failed to insert the record");
    }
}
pstmt.close();
con.close();
} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog(this, "Please enter valid numeric values");
} catch (ClassNotFoundException e) {
    System.out.print(e);
} catch (SQLException ex) {
    Logger.getLogger(RawMaterials.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

```

    }
    public static void main(String args[]) {

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

java.util.logging.Logger.getLogger(RawMaterials.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(RawMaterials.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(RawMaterials.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(RawMaterials.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 RawMaterials().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton back;
private javax.swing.JButton fetch;
private javax.swing.JTextField id;
private javax.swing.JButton jButton1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField name;
private javax.swing.JTextField price;
private javax.swing.JTextField quantity;
private javax.swing.JTextField sid;
// End of variables declaration
}

```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class Products extends javax.swing.JFrame {

    public Products() {
        initComponents();
    }

    @SuppressWarnings("unchecked")

    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        id = new javax.swing.JTextField();
        name = new javax.swing.JTextField();
        price = new javax.swing.JTextField();
        category = new javax.swing.JTextField();
        add = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jTable1.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null}
            },
            new String [] {
                "ID", "Name", "Price", "Category"
            }
        ));
        jScrollPane1.setViewportView(jTable1);

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

```

```

jButton2.setText("BACK");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});
add.setText("ADD");
add.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        addActionPerformed(evt);
    }
});
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 375,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(45, 45, 45))
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jButton1)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jButton2)
                    .addGap(98, 98, 98)))
            .addGap(10, 10, 10)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(add, javax.swing.GroupLayout.DEFAULT_SIZE, 89, Short.MAX_VALUE)
                .addComponent(category)
                .addComponent(id)
                .addComponent(name)
                .addComponent(price))
            .addGap(55, 55, 55))
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(id, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(29, 29, 29)
                    .addComponent(name, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(35, 35, 35)
                    .addComponent(price, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(35, 35, 35)
                    .addComponent(category, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(37, 37, 37)
                    .addComponent(add))
            .addGap(10, 10, 10))
    );

```

```

.addGroup(layout.createSequentialGroup()
    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 217,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(18, 18, 18)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jButton1)
        .addComponent(jButton2))))
    .addContainerGap(36, Short.MAX_VALUE))
);

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

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    DefaultTableModel model= (DefaultTableModel)jTable1.getModel();

    String q="select * from products";
    try{
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "mani",
"vasavi");
        Statement smt=con.createStatement();
        ResultSet ra=smt.executeQuery(q);
        while(ra.next()){
            int ID=ra.getInt("product_id");
            String Name=ra.getString("product_name");
            float Price=ra.getFloat("unit_price");
            String Category=ra.getString("category");

            model.addRow(new Object[] {ID,Name,Price,Category});
        }

    }catch(ClassNotFoundException e){
        System.out.print(e);
    } catch (SQLException ex) {
        Logger.getLogger(RawMaterials.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.dispose();
    Dashboard dsh=new Dashboard();
    dsh.show();
}

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

    try {
        int Id = Integer.parseInt(id.getText());
        String Name = name.getText();

        float pri = Float.parseFloat(price.getText());
        String cat = category.getText();
    }
}

```

```

String q = "INSERT INTO Products (product_id, product_name, unit_price, category) VALUES (?, ?, ?,
?)";
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "mani",
"vasavi");

    // Set auto-commit to false
    con.setAutoCommit(false);

    PreparedStatement pstmt = con.prepareStatement(q);
    pstmt.setInt(1, Id);
    pstmt.setString(2, Name);
    pstmt.setFloat(3, pri);
    pstmt.setString(4, cat);

    int rowsAffected = pstmt.executeUpdate();

    if (rowsAffected > 0) {
        con.commit(); // Commit the transaction
        JOptionPane.showMessageDialog(this, "Record inserted successfully");
    } else {
        JOptionPane.showMessageDialog(this, "Failed to insert the record");
    }

    pstmt.close();
    con.close();
} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog(this, "Please enter valid numeric values");
} catch (ClassNotFoundException e) {
    System.out.print(e);
} catch (SQLException ex) {
    Logger.getLogger(RawMaterials.class.getName()).log(Level.SEVERE, null, ex);
}
}

public static void main(String args[]) {
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(Products.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
} catch (InstantiationException ex) {
    java.util.logging.Logger.getLogger(Products.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
} catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(Products.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
    java.util.logging.Logger.getLogger(Products.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
}
}

```

```
java.awt.EventQueue.invokeLater(new Runnable() {  
    public void run() {  
        new Products().setVisible(true);  
    }  
});  
}
```

```
// Variables declaration - do not modify  
private javax.swing.JButton add;  
private javax.swing.JTextField category;  
private javax.swing.JTextField id;  
private javax.swing.JButton jButton1;  
private javax.swing.JButton jButton2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JTable jTable1;  
private javax.swing.JTextField name;  
private javax.swing.JTextField price;  
// End of variables declaration  
}
```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

import java.sql.*;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
public class Inventory extends javax.swing.JFrame {

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

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        invId = new javax.swing.JTextField();
        quant = new javax.swing.JTextField();
        date = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();
        prodId = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jTable1.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null}
            },
            new String [] {
                "Invent-ID", "Product-ID", "Quantity", "Restock-Date"
            }
        )

```



```

));
jScrollPane1.setViewportViewView(jTable1);

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

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

jButton3.setText("ADD");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(201, 201, 201)
            .addComponent(jButton3)
            .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(jButton2)
                .addComponent(invid, javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(proid, javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(quant, javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(date, javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(21, 21, 21)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                        .addComponent(jButton1)
                        .addGap(60, 60, 60)))
                .addComponent(jScrollPane1))
        );

```

```

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(12, 12, 12)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 171,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(date, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(quant, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(invid, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(proid, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addComponent(jButton3)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jButton1)
                .addComponent(jButton2))
            .addGap(14, 14, 14))
        );

```

```

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

```

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
    String q = "SELECT * FROM Inventory";
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "mani",
"vasavi");
        Statement smt = con.createStatement();
        ResultSet ra = smt.executeQuery(q);
        while (ra.next()) {
            int inventoryId = ra.getInt("inventory_id");
            int productId = ra.getInt("product_id");
            int quantityAvailable = ra.getInt("quantity_available");
            Date lastRestockDate = ra.getDate("last_restock_date");
            model.addRow(new Object[] {inventoryId, productId, quantityAvailable, lastRestockDate});
        }
        ra.close();
        smt.close();
        con.close();
    } catch (ClassNotFoundException e) {
        System.out.print(e);
    } catch (SQLException ex) {
        Logger.getLogger(RawMaterials.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.dispose();
    Dashboard dsh=new Dashboard();
    dsh.show();
}

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

    try {
        int inventoryId = Integer.parseInt(invid.getText());
        int productId = Integer.parseInt(proid.getText());
        int quantityAvailable = Integer.parseInt(quant.getText());
        java.sql.Date lastRestockDate = java.sql.Date.valueOf(date.getText());

        String q = "INSERT INTO Inventory (inventory_id, product_id, quantity_available,
last_restock_date) VALUES (?, ?, ?, ?)";
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "mani",
"vasavi");

        // Set auto-commit to false
        con.setAutoCommit(false);

        PreparedStatement pstmt = con.prepareStatement(q);
        pstmt.setInt(1, inventoryId);
        pstmt.setInt(2, productId);
        pstmt.setInt(3, quantityAvailable);
        pstmt.setDate(4, lastRestockDate);

        int rowsAffected = pstmt.executeUpdate();

        if (rowsAffected > 0) {
            con.commit(); // Commit the transaction
            JOptionPane.showMessageDialog(this, "Record inserted successfully");
        } else {
            JOptionPane.showMessageDialog(this, "Failed to insert the record");
        }

        pstmt.close();
        con.close();
    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(this, "Please enter valid numeric values");
    } catch (ClassNotFoundException e) {
        System.out.print(e);
    } catch (SQLException ex) {
        Logger.getLogger(RawMaterials.class.getName()).log(Level.SEVERE, null, ex);
    }

}

```

```

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(Inventory.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Inventory.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Inventory.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Inventory.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 Inventory().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JTextField date;
private javax.swing.JTextField invId;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField proid;
private javax.swing.JTextField quant;
// End of variables declaration
}

```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

import java.sql.*;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author DELL-PC
 */
public class Employees extends javax.swing.JFrame {

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

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

        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        id = new javax.swing.JTextField();
        position = new javax.swing.JTextField();
        salary = new javax.swing.JTextField();
        back = new javax.swing.JButton();
        fetch = new javax.swing.JButton();
        add = new javax.swing.JButton();
        name = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

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

```

```

{null, null, null, null},
    {null, null, null, null},
    {null, null, null, null},
    {null, null, null, null}
},
new String [] {
    "ID", "Name", "Position", "Salary"
}
));
jScrollPane1.setViewportViewView(jTable1);
back.setText("BACK");
back.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        backActionPerformed(evt);
    }
});
fetch.setText("FETCH");
fetch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        fetchActionPerformed(evt);
    }
});
add.setText("ADD");
add.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        addActionPerformed(evt);
    }
});
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(201, 201, 201)
            .addComponent(add)
            .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(fetch)
                .addComponent(id, javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(name, javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(position, javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(salary, javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(21, 21, 21))
            
```

```

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
    .addComponent(back)
    .addGap(60, 60, 60)))
.addComponent(jScrollPane1)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(12, 12, 12)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 171,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(salary, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(position, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(id, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(name, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addComponent(add)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(back)
            .addComponent(fetch))
        .addGap(14, 14, 14))
    );

```

```

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

```

```

private void backActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.dispose();
    Dashboard dsh=new Dashboard();
    dsh.show();
}

```

```

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

```

```

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

```

```

String q = "SELECT * FROM Employees";
try {

```

```

    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "mani",
"vasavi");
    Statement smt = con.createStatement();
    ResultSet ra = smt.executeQuery(q);
    while (ra.next()) {
        int id1 = ra.getInt("employee_id");

```

```

String name1 = ra.getString("employee_name");
    String pos = ra.getString("position");
    float sal = ra.getFloat("salary");
    model.addRow(new Object[] {id1, name1, pos, sal});
}
ra.close();
smt.close();
con.close();
} catch (ClassNotFoundException e) {
    System.out.print(e);
} catch (SQLException ex) {
    Logger.getLogger(Employees.class.getName()).log(Level.SEVERE, null, ex);
}

}

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

    try {
        int employeeId = Integer.parseInt(id.getText());
        String employeeName = name.getText();
        String pos = position.getText();
        float sal = Float.parseFloat(salary.getText());

        String q = "INSERT INTO Employees (employee_id, employee_name, position, salary) VALUES (?, ?, ?)";
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "mani",
"vasavi");

        // Set auto-commit to false
        con.setAutoCommit(false);

        PreparedStatement pstmt = con.prepareStatement(q);
        pstmt.setInt(1, employeeId);
        pstmt.setString(2, employeeName);
        pstmt.setString(3, pos);
        pstmt.setFloat(4, sal);

        int rowsAffected = pstmt.executeUpdate();

        if (rowsAffected > 0) {
            con.commit(); // Commit the transaction
            JOptionPane.showMessageDialog(this, "Record inserted successfully");
        } else {
            JOptionPane.showMessageDialog(this, "Failed to insert the record");
        }

        pstmt.close();
        con.close();
    } catch (NumberFormatException e) {

```



```

JOptionPane.showMessageDialog(this, "Please enter valid numeric values");
} catch (ClassNotFoundException e) {
System.out.print(e);
} catch (SQLException ex) {
Logger.getLogger(Employees.class.getName()).log(Level.SEVERE, null, ex);
}
}

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(Employees.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
} catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(Employees.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
} catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(Employees.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(Employees.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 Employees().setVisible(true);
}
});
}

// Variables declaration - do not modify
private javax.swing.JButton add;
private javax.swing.JButton back;
private javax.swing.JButton fetch;
private javax.swing.JTextField id;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField name;
private javax.swing.JTextField position;
private javax.swing.JTextField salary;
// End of variables declaration
}

```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

import java.sql.*;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author DELL-PC
 */
public class Expenses extends javax.swing.JFrame {

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

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

        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        id = new javax.swing.JTextField();
        amount = new javax.swing.JTextField();
        date = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();
        type = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

```

```

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "ID", "Type", "Amount", "Date"
    }
));
jScrollPane1.setViewportView(jTable1);

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

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

jButton3.setText("ADD");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(201, 201, 201)
            .addComponent(jButton3)
            .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jButton2)
            .addGap(18, 18, 18)
            .addComponent(id, javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(type, javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
        )
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(amount, javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(date, javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE)
        )
);

```

```

        .addComponent(amount, javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(date, javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(21, 21, 21))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())
            .addComponent(jButton1)
            .addGap(60, 60, 60)))
        .addComponent(jScrollPane1)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(12, 12, 12)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 171,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createSequentialGroup()
                .createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(date, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(amount, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(id, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(type, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)
                .addComponent(jButton3)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(layout.createSequentialGroup()
                    .createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jButton1)
                    .addComponent(jButton2))
                    .addGap(14, 14, 14))
            )
        );

    pack();

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        this.dispose();
        Dashboard dsh=new Dashboard();
        dsh.show();
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
        String q = "SELECT * FROM Expenses";
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "mani",
"vasavi");
            Statement smt = con.createStatement();
            ResultSet ra = smt.executeQuery(q);

```

```

while (ra.next()) {
    int expenseId = ra.getInt("expense_id");
    String expenseType = ra.getString("expense_type");
    float amt= ra.getFloat("amount");
    Date expenseDate = ra.getDate("expense_date");
    model.addRow(new Object[] {expenseId, expenseType, amt, expenseDate});
}
ra.close();
smt.close();
con.close();
} catch (ClassNotFoundException e) {
    System.out.print(e);
} catch (SQLException ex) {
    Logger.getLogger(Expenses.class.getName()).log(Level.SEVERE, null, ex);
}
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
try {
    int expenseId = Integer.parseInt(id.getText());
    String expenseType = type.getText();
    float amt = Float.parseFloat(amount.getText());
    java.sql.Date expenseDate = java.sql.Date.valueOf(date.getText());

    String q = "INSERT INTO Expenses (expense_id, expense_type, amount, expense_date) VALUES (?, ?, ?, ?)";
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "mani",
"vasavi");
    con.setAutoCommit(false);

    PreparedStatement pstmt = con.prepareStatement(q);
    pstmt.setInt(1, expenseId);
    pstmt.setString(2, expenseType);
    pstmt.setFloat(3, amt);
    pstmt.setDate(4, expenseDate);

    int rowsAffected = pstmt.executeUpdate();

    if (rowsAffected > 0) {
        con.commit(); // Commit the transaction
        JOptionPane.showMessageDialog(this, "Record inserted successfully");
    } else {
        JOptionPane.showMessageDialog(this, "Failed to insert the record");
    }

    pstmt.close();
    con.close();
} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog(this, "Please enter valid numeric values");
} catch (ClassNotFoundException e) {

```

```

System.out.print(e);
    } catch (SQLException ex) {
        Logger.getLogger(Expenses.class.getName()).log(Level.SEVERE, null, ex);
    }

}

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(Expenses.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Expenses.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Expenses.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Expenses.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Expenses().setVisible(true);
        }
    });
}

private javax.swing.JTextField amount;
private javax.swing.JTextField date;
private javax.swing.JTextField id;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField type;
// End of variables declaration
}

```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

import java.sql.*;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author DELL-PC
 */
public class Customers extends javax.swing.JFrame {

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

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

        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        id = new javax.swing.JTextField();
        cd = new javax.swing.JTextField();
        back = new javax.swing.JButton();
        fetch = new javax.swing.JButton();
        add = new javax.swing.JButton();
        name = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

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

```

```

new Object [] {
    {null, null, null},
    {null, null, null},
    {null, null, null},
    {null, null, null}
},
new String [] {
    "ID", "Name", "Contact Details"
}
));
jScrollPane1.setViewportView(jTable1);

back.setText("BACK");
back.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        backActionPerformed(evt);
    }
});
fetch.setText("FETCH");
fetch.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        fetchActionPerformed(evt);
    }
});
add.setText("ADD");
add.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        addActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 454,
Short.MAX_VALUE)
        .addGroup(layout.createSequentialGroup()
            .addGap(27, 27, 27)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(add)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                        .addComponent(fetch)
                        .addComponent(id, javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGap(54, 54, 54)
                    .addComponent(name, javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(cd, javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(back))
            )
        )

```



```

.addGap(21, 21, 21))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(12, 12, 12)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 171,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(cd, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(id, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(name, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 18,
Short.MAX_VALUE)
        .addComponent(add)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(back)
            .addComponent(fetch))
        .addGap(14, 14, 14))
);
pack();
} // </editor-fold>
private void backActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.dispose();
    Dashboard dsh=new Dashboard();
    dsh.show();
}
private void fetchActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
    String q = "SELECT * FROM Customers";
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "mani",
"vasavi");
        Statement smt = con.createStatement();
        ResultSet ra = smt.executeQuery(q);
        while (ra.next()) {
            int customerId = ra.getInt("customer_id");
            String customerName = ra.getString("customer_name");
            String contactInfo = ra.getString("contact_info");
            model.addRow(new Object[] {customerId, customerName, contactInfo});
        }
        ra.close();
        smt.close();
        con.close();
    }
}

```

```

} catch (ClassNotFoundException e) {
    System.out.print(e);
} catch (SQLException ex) {
    Logger.getLogger(Customers.class.getName()).log(Level.SEVERE, null, ex);
}

}

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

    try {
        int customerId = Integer.parseInt(id.getText());
        String customerName = name.getText();
        String contactInfo = cd.getText();

        String q = "INSERT INTO Customers (customer_id, customer_name, contact_info) VALUES (?, ?, ?)";
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "mani", "vasavi");

        // Set auto-commit to false
        con.setAutoCommit(false);

        PreparedStatement pstmt = con.prepareStatement(q);
        pstmt.setInt(1, customerId);
        pstmt.setString(2, customerName);
        pstmt.setString(3, contactInfo);

        int rowsAffected = pstmt.executeUpdate();

        if (rowsAffected > 0) {
            con.commit(); // Commit the transaction
            JOptionPane.showMessageDialog(this, "Record inserted successfully");
        } else {
            JOptionPane.showMessageDialog(this, "Failed to insert the record");
        }

        pstmt.close();
        con.close();
    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(this, "Please enter valid numeric values");
    } catch (ClassNotFoundException e) {
        System.out.print(e);
    } catch (SQLException ex) {
        Logger.getLogger(Customers.class.getName()).log(Level.SEVERE, null, ex);
    }

}

public static void main(String args[]) {

```

```

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(Customers.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Customers.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Customers.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Customers.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 Customers().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton add;
private javax.swing.JButton back;
private javax.swing.JTextField cd;
private javax.swing.JButton fetch;
private javax.swing.JTextField id;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField name;
// End of variables declaration
}

```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author DELL-PC
 */
public class Signup extends javax.swing.JFrame {

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

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

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        usr = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        pwd = new javax.swing.JPasswordField();
        su = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel1.setText("Password");

        jLabel2.setFont(new java.awt.Font("Segoe UI", 1, 18)); // NOI18N
        jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel2.setText("SIGN UP");
    }

```

```

jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel3.setText("User Name :");

pwd.setText("jPasswordField1");

su.setText("Sign up");
su.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        suActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(104, 104, 104)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 81,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 81,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(53, 53, 53)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(usr)
                .addComponent(pwd, javax.swing.GroupLayout.DEFAULT_SIZE, 108,
Short.MAX_VALUE))
            .addGap(54, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addGap(0, 100, Short.MAX_VALUE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 81,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(150, 150, 150))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                    .addComponent(su, javax.swing.GroupLayout.PREFERRED_SIZE, 93,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(141, 141, 141))))
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(15, 15, 15)
            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(36, 36, 36)
                    .addComponent(usr, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addGap(32, 32, 32)

```

```

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(41, 41, 41)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(pwd, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 62,
Short.MAX_VALUE)
            .addComponent(su, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(27, 27, 27))
        );

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

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

    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "mani",
"vasavi");

        String us = usr.getText();
        String pd = new String(pwd.getPassword());

        String sql = "INSERT INTO admin (ID, PASS) VALUES (?, ?)";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setString(1, us);
        pstmt.setString(2, pd);

        int rowsAffected = pstmt.executeUpdate();
        if (rowsAffected > 0) {
            this.dispose();
            Dashboard dsh = new Dashboard();
            dsh.show();
        } else {
            JOptionPane.showMessageDialog(this, "Failed to insert user");
        }

        pstmt.close();
        con.close();
    } catch (ClassNotFoundException e) {
        System.out.print("Class not found");
    } catch (SQLException ex) {
        Logger.getLogger(LoginPage.class.getName()).log(Level.SEVERE, null, ex);
    }

}

```

```

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(Signup.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Signup.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Signup.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Signup.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 Signup().setVisible(true);
        }
    });
}

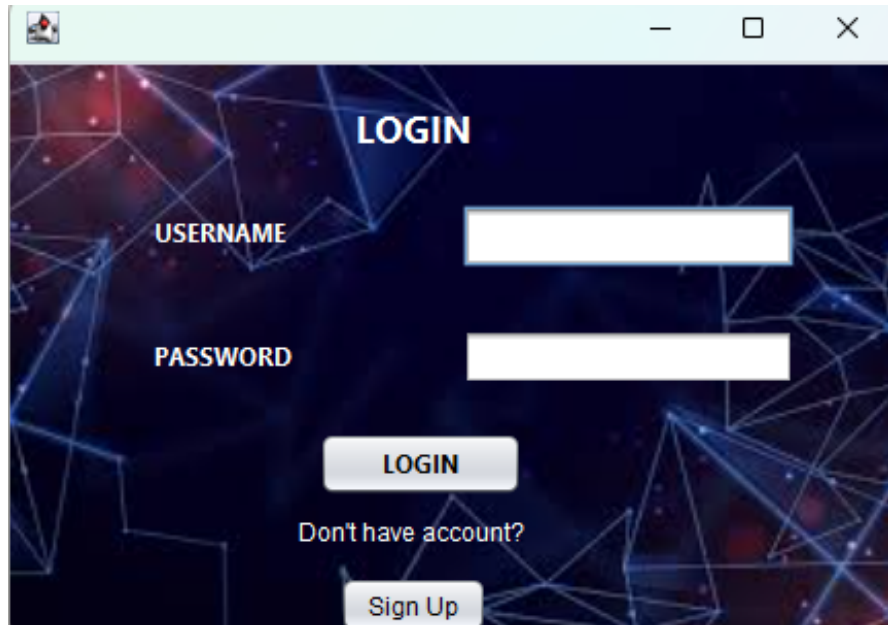
// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPasswordField pwd;
private javax.swing.JButton su;
private javax.swing.JTextField usr;
// End of variables declaration
}

```

# OUTPUT

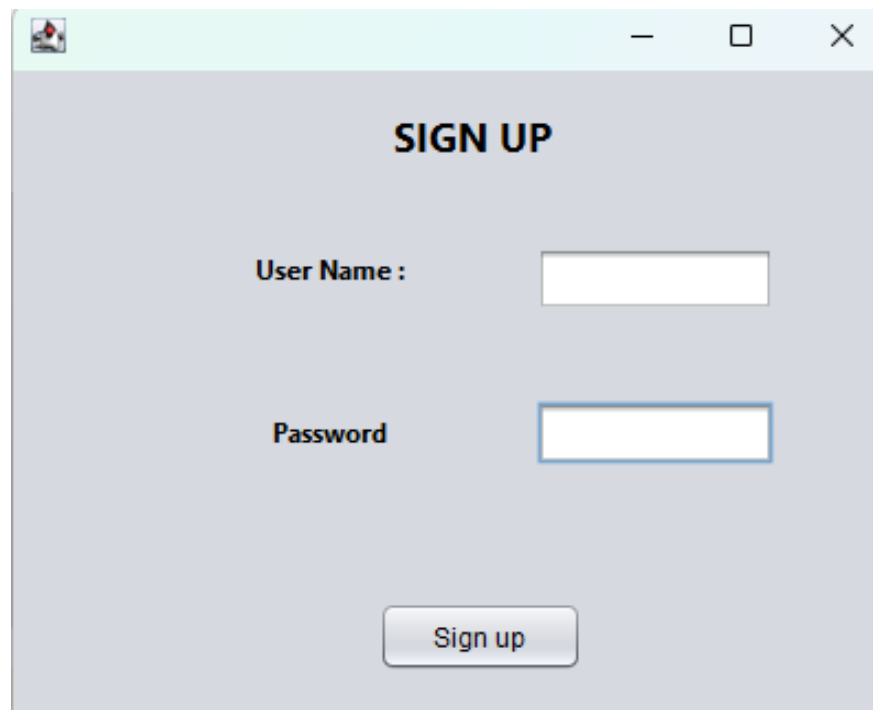
## SCREENSHOTS:

Login Page



A screenshot of a web application window titled "LOGIN". The window has a light blue title bar with standard minimize, maximize, and close buttons. The background of the page is dark blue with a network-like pattern of glowing blue lines and red dots. The word "LOGIN" is displayed in large, bold, white capital letters at the top right. Below it, the labels "USERNAME" and "PASSWORD" are in white capital letters, each followed by a white rectangular input field. A white "LOGIN" button is centered below the input fields. Below the button, the text "Don't have account?" is displayed in white. At the bottom, there is a white "Sign Up" button.

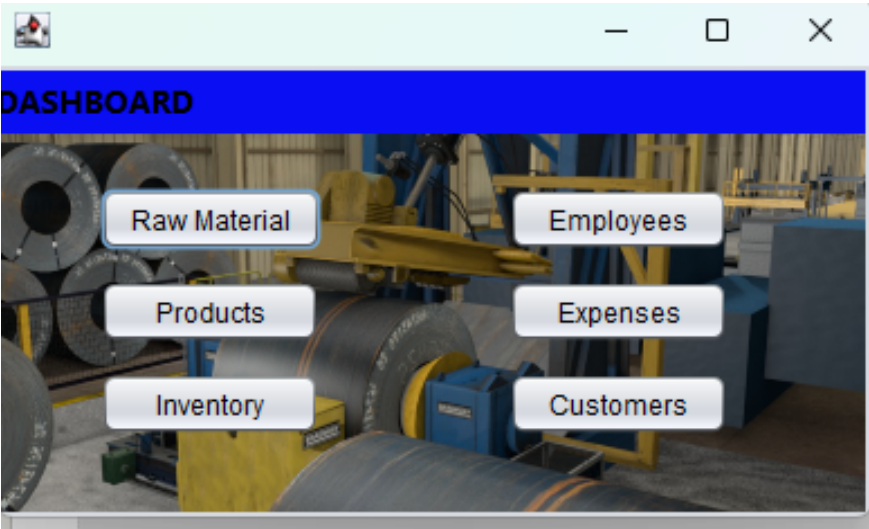
Sign Up Page:



A screenshot of a web application window titled "SIGN UP". The window has a light blue title bar with standard minimize, maximize, and close buttons. The background is a solid light gray. The word "SIGN UP" is displayed in large, bold, black capital letters at the top center. Below it, the labels "User Name :" and "Password" are in black, each followed by a white rectangular input field. A white "Sign up" button is centered at the bottom of the page.



DASHBOARD:



Raw Material (Details):

A screenshot of a software application window titled "Raw Material (Details)". The window has a standard Windows-style title bar. The main content area contains a table with five columns: "ID", "Name", "Quantity", "price", and "Supplier\_id". The table has four data rows. Below the table, there are five empty input fields for adding new data, followed by three buttons: "ADD", "FETCH", and "BACK".

ID	Name	Quantity	price	Supplier_id
1	Iron	1000	10	1
2	Steel	800	15	2
3	Aluminum	1200	8	3
14	stellplastic	242345	789	674

Products (Details):

—□×

ID	Name	Price	Category
1	Chair	25.0	Furniture
2	Table	89.0	Furniture
3	Lamp	15.0	Lighting
4	bulb	23.0	lighting
5	rod	45.0	lighting

FETCH

BACK

ADD

Inventory (Details):

—□×

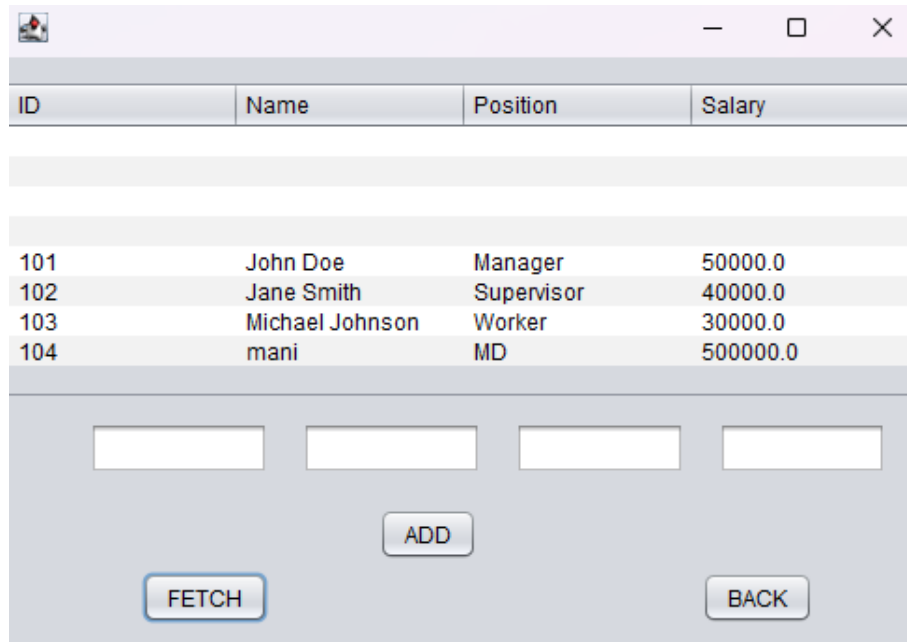
Invent-ID	Product-ID	Quantity	Restock-Date
1	1	50	2024-05-12
2	2	20	2024-05-11
3	3	100	2024-05-10
4	4	120	2024-05-11

ADD

FETCH

BACK

## Employees (Details):



ID	Name	Position	Salary
101	John Doe	Manager	50000.0
102	Jane Smith	Supervisor	40000.0
103	Michael Johnson	Worker	30000.0
104	mani	MD	500000.0

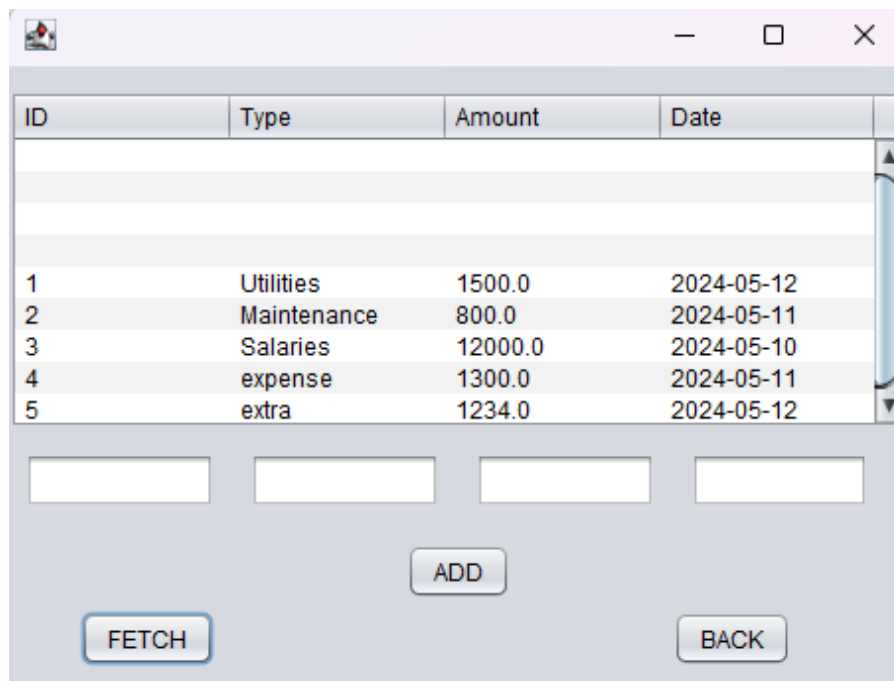
Four input fields for adding new employee data.

ADD

FETCH

BACK

## Expenses (Details):



ID	Type	Amount	Date	
1	Utilities	1500.0	2024-05-12	
2	Maintenance	800.0	2024-05-11	
3	Salaries	12000.0	2024-05-10	
4	expense	1300.0	2024-05-11	
5	extra	1234.0	2024-05-12	

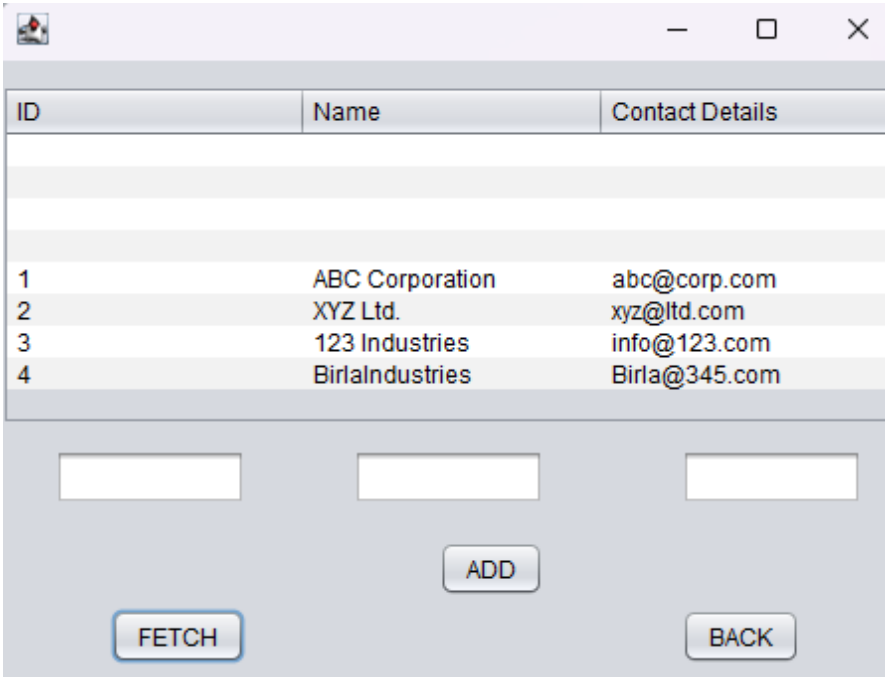
Four input fields for adding new expense data.

ADD

FETCH

BACK

## Customers (Details):



ID	Name	Contact Details
1	ABC Corporation	abc@corp.com
2	XYZ Ltd.	xyz@ltd.com
3	123 Industries	info@123.com
4	BirlaIndustries	Birla@345.com

# RESULTS I have successfully completed the mini-project “MILL INDUSTRY MANAGEMENT”.

## DISCUSSION AND FUTURE WORK

This project involves the development of a Mill Industry Management System using JDBC (Java Database Connectivity) to manage various operations such as raw materials, products, inventory, customer orders, employees, suppliers, sales, and expenses. The system includes a user-friendly interface featuring a login and signup mechanism, ensuring secure access for users. Upon successful login, users are presented with a dashboard that contains buttons for each table within the database, including Raw Materials, Products, Inventory, Employees, Expenses, and Customers. When a button is clicked, a new window opens displaying the respective table, providing functionalities to fetch and add records. These windows ensure that users can easily view the current data and input new entries seamlessly.

The implementation using JDBC allows for efficient interaction between the Java application and the Oracle SQL database, ensuring robust data manipulation and retrieval operations. Each table window includes 'Fetch' and 'Add' buttons, enabling users to retrieve the latest data from the database and add new records, respectively. The 'Back' button allows users to return to the main dashboard, ensuring easy navigation within the system.

### **Future Work**

In future enhancements, the system will incorporate 'Delete' buttons in each table window to allow users to remove records, enhancing data management capabilities. Additionally, an 'Update' button will be added to enable users to modify existing records directly from the interface. The expansion of the database to include more tables and relationships will further enrich the system, allowing for a more comprehensive management solution. Advanced features such as real-time data analytics, automated alerts for low inventory levels, and integration with other enterprise systems like ERP and SCM will be considered to enhance operational efficiency and decision-making processes. Enhancing the user interface for better usability and developing mobile access to the system will also be prioritized to ensure flexibility and ease of use for managers and employees. By implementing these future enhancements, the Mill Industry Management System will evolve into a powerful tool, providing greater value through improved efficiency, better decision-making capabilities, and enhanced operational control.

## REFERENCES

<https://docs.oracle.com/javase/7/docs/api/>  
<https://www.javatpoint.com/java-swing>  
<https://stackoverflow.com/>