```
package User;

import javax.swing.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.ResultSet;

public class TestDesignLoin extends JFrame {

    private Connection cn;

    public TestDesignLoin() {

        setTitle("Registration Form");
        setLayout(null);
        setSize(500, 450);

        // Create labels
        JLabel label_id = new JLabel("Enter ID:");
        JLabel label_name = new JLabel("Enter Name:");
        JLabel label_sex = new JLabel("Enter Sex:");
        JLabel label_salary = new JLabel("Enter Salary:");

        // Create text fields
        JTextField jtextfield_id = new JTextField(30);
        JTextField jtextfield_name = new JTextField(30);
        JTextField jtextfield_sex = new JTextField(30);
```

```
JTextField jtextfield_salary = new JTextField(30);
JTextField jtextfield_delet_by_name = new JTextField(30);
JTextField jtextfield_delet_by_id = new JTextField(30);

// Create buttons
JButton jbutton_insert = new JButton("Insert");
JButton jbutton_show = new JButton("Show Data");
JButton jbutton_delete_by_name = new JButton("Delete by Name");
JButton jbutton_delete_by_id = new JButton("Delete by ID");

// Set bounds for labels, text fields, and buttons
label_id.setBounds(50, 30, 100, 30);
label_name.setBounds(50, 70, 100, 30);
label_sex.setBounds(50, 110, 100, 30);
label_salary.setBounds(50, 150, 100, 30);

jtextfield_id.setBounds(180, 30, 200, 30);
jtextfield_name.setBounds(180, 70, 200, 30);
jtextfield_sex.setBounds(180, 110, 200, 30);
jtextfield_salary.setBounds(180, 150, 200, 30);

jbutton_insert.setBounds(50, 200, 130, 30);
jbutton_show.setBounds(50, 250, 130, 30);
jbutton_delete_by_name.setBounds(50, 300, 130, 30);
jtextfield_delet_by_name.setBounds(190, 300, 130, 30);
jbutton_delete_by_id.setBounds(50, 350, 130, 30);
jtextfield_delet_by_id.setBounds(190, 350, 130, 30);

// Add components to the frame
```

```
add(label_id);
add(jtextfield_id);
add(label_name);
add(jtextfield_name);
add(label_sex);
add(jtextfield_sex);
add(label_salary);
add(jtextfield_salary);
add(jbutton_insert);
add(jbutton_show);
add(jbutton_delete_by_name);
add(jbutton_delete_by_id);
add(jtextfield_delet_by_name);
add(jtextfield_delet_by_id);

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLocationRelativeTo(null);
setVisible(true);

// button insert
jbutton_insert.addActionListener(
      e -> insertData(
            jtextfield_id.getText(),
            jtextfield_name.getText(),
            jtextfield_sex.getText(),
            jtextfield_salary.getText())
);

// button show
```

```
    jbutton_show.addActionListener(e -> showData());

    // button delete by name
    jbutton_delete_by_name.addActionListener(
            e -> deleteByName(jtextfield_delet_by_name.getText()));

    // button delete by id
    jbutton_delete_by_id.addActionListener(
            e -> deleteById(jtextfield_delet_by_id.getText()));
}

private void connectionDatabase() {
    try {

        cn = DriverManager.getConnection("jdbc:mysql://localhost:3306/stock", "root", "");
        JOptionPane.showMessageDialog(null, "Connected successfully to the database.");
    } catch (SQLException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(null, "Error: " + e.getMessage());
    }
}

private void createTable() {

    String createTable_employee = "CREATE TABLE IF NOT EXISTS employee ("
            + "id INT PRIMARY KEY, "
            + "name VARCHAR(255), "
            + "sex VARCHAR(30), "
            + "salary DOUBLE)";
```

```
        Statement stmt = null;

        try {
            if (cn == null || cn.isClosed()) {
                connectionDatabase();
            }

            stmt = cn.createStatement();
            stmt.executeUpdate(createTable_employee);

            JOptionPane.showMessageDialog(null, "Table 'employee' created successfully.");

        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(null, "Error creating table: " + e.getMessage());
        }
    }

    // Method  insert data into the database
    private void insertData(String id, String name, String sex, String salary) {
        try {
            Statement sm_object = cn.createStatement();
            String insert_query = "INSERT INTO employee (id, name, sex, salary) VALUES (" + id + ", '" + name
+ "', '" + sex + "', " + salary + ")";
            sm_object.executeUpdate(insert_query);
            JOptionPane.showMessageDialog(null, "Data inserted successfully.");
        } catch (SQLException e) {
            e.printStackTrace();
```

```
        JOptionPane.showMessageDialog(null, "Error inserting data: " + e.getMessage());
    }
}

// Method to show data from the database
private void showData() {
    try {
        Statement sm_object = cn.createStatement();
        String query_showData = "SELECT * FROM employee";
        ResultSet rs = sm_object.executeQuery(query_showData);
        StringBuilder sb = new StringBuilder();
        while (rs.next()) {
            sb.append("ID: ").append(rs.getInt("id"))
                    .append(", Name: ").append(rs.getString("name"))
                    .append(", Sex: ").append(rs.getString("sex"))
                    .append(", Salary: ").append(rs.getFloat("salary"))
                    .append("\n");
        }
        JOptionPane.showMessageDialog(null, sb.toString());
    } catch (SQLException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(null, "Error retrieving data: " + e.getMessage());
    }
}

// Method to delete data by name
private void deleteByName(String name) {
    try {
        Statement sm_object = cn.createStatement();
```

```
        String query_deleteByName = "DELETE FROM employee WHERE name = '" + name + "'";
        int rowsDeleted = sm_object.executeUpdate(query_deleteByName);
        if (rowsDeleted > 0) {
            JOptionPane.showMessageDialog(null, "Record deleted successfully.");
        } else {
            JOptionPane.showMessageDialog(null, "No record found with the given name.");
        }
    } catch (SQLException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(null, "Error deleting record: " + e.getMessage());
    }
}

// Method to delete data by ID
private void deleteById(String id) {
    try {
        Statement sm_object = cn.createStatement();
        String query_deleteById = "DELETE FROM employee WHERE id = " + id;
        int rowsDeleted = sm_object.executeUpdate(query_deleteById);
        if (rowsDeleted > 0) {
            JOptionPane.showMessageDialog(null, "Record deleted successfully.");
        } else {
            JOptionPane.showMessageDialog(null, "No record found with the given ID.");
        }
    } catch (SQLException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(null, "Error deleting record: " + e.getMessage());
    }
}
```

```
    public static void main(String[] args) {
        TestDesignLoin table = new TestDesignLoin();
        table.connectionDatabase();
        table.createTable();
    }
}

/*
+ port 80
  HTTP Communication:When a web browser sends a request to a server
    (ex: when you enter a URL in your browser), request is sent to port 80 on the web server

+ port 3306
    Database Communication: Port 3306 is used for client-server communication in MySQL database systems.

+ class connection

   Connection  cn = DriverManager.getConnection("jdbc:mysql://localhost:3306/stock", "root", "");

*/
```