**Experiment No. 4**

**Title:** Implementation of CRUD operation.

**Batch:** A2 **Roll No.:** 1914072 **Experiment No.:**4

**Aim:** Create a form using Swing form to accept the user input and store it in database. Perform and demonstrate CRUD operation on the button click.

---

**Resources needed:** Java Development kit, Netbeans, Mysql Server(XAMPP)

---

**Theory:**

Swing API is set of extensible GUI Components to ease developer's life to create JAVA based Front End/ GUI Applications. It is built upon top of AWT API and acts as replacement of AWT API as it has almost every control corresponding to AWT controls. It has more powerful and flexible components than AWT. Swing component follows a Model-View-Controller architecture. Java Swing is Light Weight; Rich in controls and Highly Customizable. Swing controls inherit properties from following Component class hierarchy. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

| No. | Class | Description |
|---|---|---|
| 1 | **Component** | A Container is the abstract base class for the non menu user-interface controls of SWING. Component represents an object with graphical representation |
| 2 | **Container** | A Container is a component that can contain other SWING components. |
| 3 | **JComponent** | A JComponent is a base class for all swing UI components. In order to use a swing component that inherits from JComponent, component must be in a containment hierarchy whose root is a top-level Swing container. |

Following is the list of commonly used controls while designed GUI using SWING.

| No. | Control | Description |
|---|---|---|
| 1 | **JLabel** | A JLabel object is a component for placing text in a container. |
| 2 | **JButton** | This class creates a labeled button. |

| | | |
|---|---|---|
| 3 | **JColorChooser** | A JColorChooser provides a pane of controls designed to allow a user to manipulate and select a color. |
| 4 | **JCheck Box** | A JCheckBox is a graphical component that can be in either an **on**(true) or **off** (false) state. |
| 5 | **JRadioButton** | The JRadioButton class is a graphical component that can be in either an **on** (true) or **off** (false) state. in a group. |
| 6 | **JList** | A JList component presents the user with a scrolling list of text items. |
| 7 | **JComboBox** | A JComboBox component presents the user with a to show up menu of choices. |
| 8 | **JTextField** | A JTextField object is a text component that allows for the editing of a single line of text. |
| 9 | **JPasswordField** | A JPasswordField object is a text component specialized for password entry. |
| 10 | **JTextArea** | A JTextArea object is a text component that allows for the editing of a multiple lines of text. |
| 11 | **ImageIcon** | A ImageIcon control is an implementation of the Icon interface that paints Icons from Images |
| 12 | **JScrollbar** | A Scrollbar control represents a scroll bar component in order to enable user to select from range of values. |
| 13 | **JOptionPane** | JOptionPane provides set of standard dialog boxes that prompt users for a value or informs them of something. |
| 14 | **JFileChooser** | A JFileChooser control represents a dialog window from which the user can select a file. |
| 15 | **JProgressBar** | As the task progresses towards completion, the progress bar displays the task's percentage of completion. |
| 16 | **JSlider** | A JSlider lets the user graphically select a value by sliding a knob within a bounded interval. |
| 17 | **JSpinner** | A JSpinner is a single line input field that lets the user select a number or an object value from an ordered sequence. |



**java.awt.GridLayout Package**: The GridLayout class is a **layout manager** that lays out a container's components in a rectangular grid. The container is divided into equal-sized rectangles, and one component is placed in each rectangle, like rows and columns.

**Event Handling**

Any program that uses GUI (graphical user interface) such as Java application written for windows, is event driven. Changing the state of an object is known as an event. For example, click on button, dragging mouse etc. Event Handling is the mechanism that controls the event and decides what should happen if an event occurs. Event handling has three main components.

- **Events :** An event is a change of state of an object.
- **Events Source :** Event source is an object that generates an event. Source is responsible for providing information of the occurred event to it's handler.
- **Listeners :** A listener is an object that listens to the event. A listener gets notified when an event occurs.**Listener** also known as event handler is responsible for generating response to an event. Listener waits until it receives an event. The java.awt.event package provides many event classes and Listener interfaces for event handling. In this package we use **ActionListener Interface**. The listener interface is for receiving action events. The class that is interested in processing an action event implements this interface, and the object created with that class is registered with a component, using the component's**addActionListener** method. When the action event occurs, that object's **actionPerformed** method is invoked.Following steps are required to perform event handling:

  1. Implement the Listener interface and overrides its methods
  2. Register the component with the Listener

For registering the component with the Listener, many classes provide the registration methods. For example:

**Button**: public void addActionListener(ActionListener a){}

We can put the event handling code in Same class or Other class or Annonymous class.

**Important Event Classes and Interfaces**

| Event Classes | Description | Listener Interface |
|---|---|---|
| **ActionEvent** | generated when button is pressed, menu-item is selected, list-item is double clicked | ActionListener |
| **MouseEvent** | generated when mouse is dragged, moved,clicked,pressed or released also when the enters or exit a component | MouseListener |
| **KeyEvent** | generated when input is received from keyboard | KeyListener |

| | | |
|---|---|---|
| **ItemEvent** | generated when check-box or list item is clicked | ItemListener |
| **TextEvent** | generated when value of textarea or textfield is changed | TextListener |
| **MouseWheelEvent** | generated when mouse wheel is moved | MouseWheelListener |
| **WindowEvent** | generated when window is activated, deactivated, deiconified, iconified, opened or closed | WindowListener |
| **ComponentEvent** | generated when component is hidden, moved, resized or set visible | ComponentEventListener |
| **ContainerEvent** | generated when component is added or removed from container | ContainerListener |
| **AdjustmentEvent** | generated when scroll bar is manipulated | AdjustmentListener |
| **FocusEvent** | generated when component gains or loses keyboard focus | FocusListener |

**Connecting to the Database**

Input obtained from GUI can be stored in database using the database connectivity code provided. Validation process is included before storing data into database. Java JDBC is a java API to connect and execute query with the database. JDBC API uses jdbc drivers to connect with the database. There are 6 steps to connect any java application with the database in java using JDBC. They are as follows:

1. **Import the packages:** Requires that you include the packages containing the JDBC classes needed for database programming. These 3 import statements should be present.
    - **import java.sql.Connection;**
    - **import java.sql.DriverManager;**
    - **import java.sql.SQLException;**
2. **Register the driver class:** The forName() method of Class class is used to register the driver class. This method is used to dynamically load the driver class. Eg: Class.forName("com.mysql.jdbc.Driver");
3. **Creating connection:** To connect to a database you need a Connection object. The Connection object uses a **DriverManager**. The DriverManager passes in your database username, your password, and the location of the database. The getConnection() method of DriverManager class is used to establish connection with the database. For the mysql

database the driver class is **com.mysql.jdbc.Driver**., connection is **jdbc:mysql://localhost:3306/sonoo** where jdbc is the API, mysql is the database, localhost is the server name on which mysql is running, we may also use IP address, 3306 is the port number and sonoo is the database name. The default username for the mysql database is **root**. Password is given by the user at the time of installing the mysql database. Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/pl2db","root","root");

4. **Creating statement:** The createStatement() method of Connection interface is used to create statement. The object of statement is responsible. The JDBC*Statement, CallableStatement,* and *PreparedStatement* interfaces define the methods and properties that enable you to send SQL or PL/SQL commands and receive data from your database.They also define methods that help bridge data type differences between Java and SQL data types used in a database.
5. **Executing queries:** The executeQuery() method of Statement interface is used to execute queries to the database. This method returns the object of ResultSet that can be used to get all the records of a table.

   Eg: ResultSet rs=stmt.executeQuery("select * from emp");

   **while**(rs.next()){

   System.out.println(rs.getInt(1)+" "+rs.getString(2));

   }

6. **Closing connection:** By closing connection object statement and ResultSet will be closed automatically. The close() method of Connection interface is used to close the connection. Eg: con.close();

**Running the Application/Form**

For running the application following files are required:

- open project then right click open new -> jFrame-> name your Frame and untick the main class.
- Right click on project and select properties add mysql-connector.bin jar file to the library.
- Design your application page by Drag and drop components from Palette
- open xampp control panel and Start apache and mysql
- Type **localhost/phpmyadmin** in browser.
- Click on Database tab ,Name your database and click on create
- Database will be listed then click on that to create table

- Name your table add no. of columns required for your table. (can be modified later)—click on GO
- Add attributes as per application requirement.Save this table
- Table will be created inside your database

**Results: (Code and Screen shots of GUI developed with Output )**

**Code:**

```
package newpackage;

import java.sql.*;
import java.lang.*;
import javax.swing.JOptionPane;
import java.text.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import javax.swing.JScrollPane;
import javax.swing.JTable;

public class NewJForm extends javax.swing.JFrame {

   public NewJForm() {
      initComponents();
   }

  private void initComponents() {

      buttonGroup1 = new javax.swing.ButtonGroup();
      jPanel1 = new javax.swing.JPanel();
      jScrollPane1 = new javax.swing.JScrollPane();
      jTable1 = new javax.swing.JTable();
```

```
jLabel1 = new javax.swing.JLabel();
jTextField1 = new javax.swing.JTextField();
jButton1 = new javax.swing.JButton();
jLabel2 = new javax.swing.JLabel();
jTextField2 = new javax.swing.JTextField();
jLabel3 = new javax.swing.JLabel();
jTextField3 = new javax.swing.JTextField();
jButton2 = new javax.swing.JButton();
jTextField4 = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();
jRadioButton1 = new javax.swing.JRadioButton();
jRadioButton2 = new javax.swing.JRadioButton();
jLabel5 = new javax.swing.JLabel();
jComboBox1 = new javax.swing.JComboBox<>();
jLabel6 = new javax.swing.JLabel();
jCheckBox3 = new javax.swing.JCheckBox();
jButton3 = new javax.swing.JButton();
jButton4 = new javax.swing.JButton();
jButton5 = new javax.swing.JButton();

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 [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jScrollPane1.setViewportView(jTable1);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
            .addContainerGap(15, Short.MAX_VALUE)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 352,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
    );
    jPanel1Layout.setVerticalGroup(
       jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
       .addGroup(jPanel1Layout.createSequentialGroup()
          .addContainerGap()
          .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 289,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addContainerGap(15, Short.MAX_VALUE))
    );

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabel1.setText("Name");

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

    jLabel2.setText("Roll No");

    jLabel3.setText("Email");

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

```
        jTextField4.setBackground(new java.awt.Color(0, 255, 255));
        jTextField4.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
        jTextField4.setHorizontalAlignment(javax.swing.JTextField.CENTER);
        jTextField4.setText("Registration Form");
        jTextField4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jTextField4ActionPerformed(evt);
            }
        });

        jLabel4.setText("Gender");

        buttonGroup1.add(jRadioButton1);
        jRadioButton1.setText("Male");

        buttonGroup1.add(jRadioButton2);
        jRadioButton2.setText("Female");

        jLabel5.setText("Department");

        jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"IT", "COMP", "MECH", "ETRX", "EXTC" }));

        jLabel6.setText("If Hostelite");

        jCheckBox3.setText("Hostelite");

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

        jButton4.setText("Delete");
        jButton4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            jButton4ActionPerformed(evt);
        }
    });

    jButton5.setText("Show All records");
    jButton5.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton5ActionPerformed(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(26, 26, 26)
                    .addComponent(jButton1)
                    .addGap(15, 15, 15)
                    .addComponent(jButton2)
                    .addGap(18, 18, 18)
                    .addComponent(jButton3)
                    .addGap(18, 18, 18)
                    .addComponent(jButton4)
                    .addGap(18, 18, 18)
                    .addComponent(jButton5))
                .addGroup(layout.createSequentialGroup()
                    .addGap(61, 61, 61)

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

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                        .addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
                .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE,
65, Short.MAX_VALUE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
                .addComponent(jLabel5,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                .addComponent(jLabel4,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
            .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE,
54, javax.swing.GroupLayout.PREFERRED_SIZE))

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

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

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                .addComponent(jTextField1,
javax.swing.GroupLayout.DEFAULT_SIZE, 143, Short.MAX_VALUE)
                .addComponent(jTextField2)
                .addComponent(jTextField3))
            .addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE, 94,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jRadioButton1)
                .addGap(18, 18, 18)
                .addComponent(jRadioButton2))
            .addComponent(jCheckBox3,
javax.swing.GroupLayout.PREFERRED_SIZE, 79,
javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        .addComponent(jTextField4)
```

```
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED_SIZE, 45,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel1)
                    .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(20, 20, 20)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel2)
                    .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel3))
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel4)
                    .addComponent(jRadioButton1)
                    .addComponent(jRadioButton2))
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel5)
                    .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel6)
                .addComponent(jCheckBox3))
            .addGap(26, 26, 26)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jButton1)
                .addComponent(jButton2)
                .addComponent(jButton3)
                .addComponent(jButton4)
                .addComponent(jButton5))
            .addContainerGap(34, Short.MAX_VALUE))
        );

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

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String str1=jTextField1.getText();
        String str2=jTextField2.getText();
        String str3=jTextField3.getText();
        String gender="Male";
        if(jRadioButton2.isSelected()){
            gender="Female";
        }
        String Dept="";
        Dept=jComboBox1.getSelectedItem().toString();
        String Type="Non Hostelite";
        if(jCheckBox3.isSelected()){
            Type="Hostelite";
        }
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con=(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/form?zeroDateTimeBehavior=C
ONVERT_TO_NULL","root","");
            PreparedStatement ps = (PreparedStatement) con.prepareStatement("insert into info
```

```
values(?,?,?,?,?,?)");
            ps.setString(1, str1);
            ps.setString(2, str2);
            ps.setString(3, str3);
            ps.setString(4, gender);
            ps.setString(5, Dept);
            ps.setString(6, Type);
            int row=ps.executeUpdate();
             con.close();
            JOptionPane.showMessageDialog(this,"Record Is Successfully Inserted");
        }
        catch(Exception e){}
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        String str2=jTextField2.getText();
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con=(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/form?zeroDateTimeBehavior=C
ONVERT_TO_NULL","root","");
            ResultSet ps1 = ((PreparedStatement) con.prepareStatement("SELECT * FROM
info where RollNo="+str2)).executeQuery();
            String res="";
            while(ps1.next()){
               res+="Name:"+ps1.getString(1)+"\n"+"Roll
No:"+ps1.getString(2)+"\n"+"Email:"+ps1.getString(3)+"\n"+"Gender:"+ps1.getString(4)+"\
n"+"Department:"+ps1.getString(5)+"\n"+"Type:"+ps1.getString(6);
            }
            if(res==""){
              JOptionPane.showMessageDialog(this,"NO record Found");
            }
            else{
            JOptionPane.showMessageDialog(this,res);
            }
             con.close();

        }
```

```
    catch(Exception e){}
  }

  private void jTextField4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
  }

  private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    String str1=jTextField1.getText();
    String str2=jTextField2.getText();
    String str3=jTextField3.getText();
    String gender="Male";
    if(jRadioButton2.isSelected()){
      gender="Female";
    }
    String Dept="";
    Dept=jComboBox1.getSelectedItem().toString();
    String Type="Non Hostelite";
    if(jCheckBox3.isSelected()){
      Type="Hostelite";
    }
    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con=(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/form?zeroDateTimeBehavior=C
ONVERT_TO_NULL","root","");
        PreparedStatement ps = (PreparedStatement) con.prepareStatement("update info set
fname=?,email=?,Gender=?,Department=?,Type=? where RollNo=?");
        ps.setString(1, str1);
        ps.setString(2, str3);
        ps.setString(3, gender);
        ps.setString(4, Dept);
        ps.setString(5, Type);
        ps.setString(6, str2);
        int row=ps.executeUpdate();
         con.close();
         if(row!=0){
         JOptionPane.showMessageDialog(this,"Record Is Successfully Updated");
```

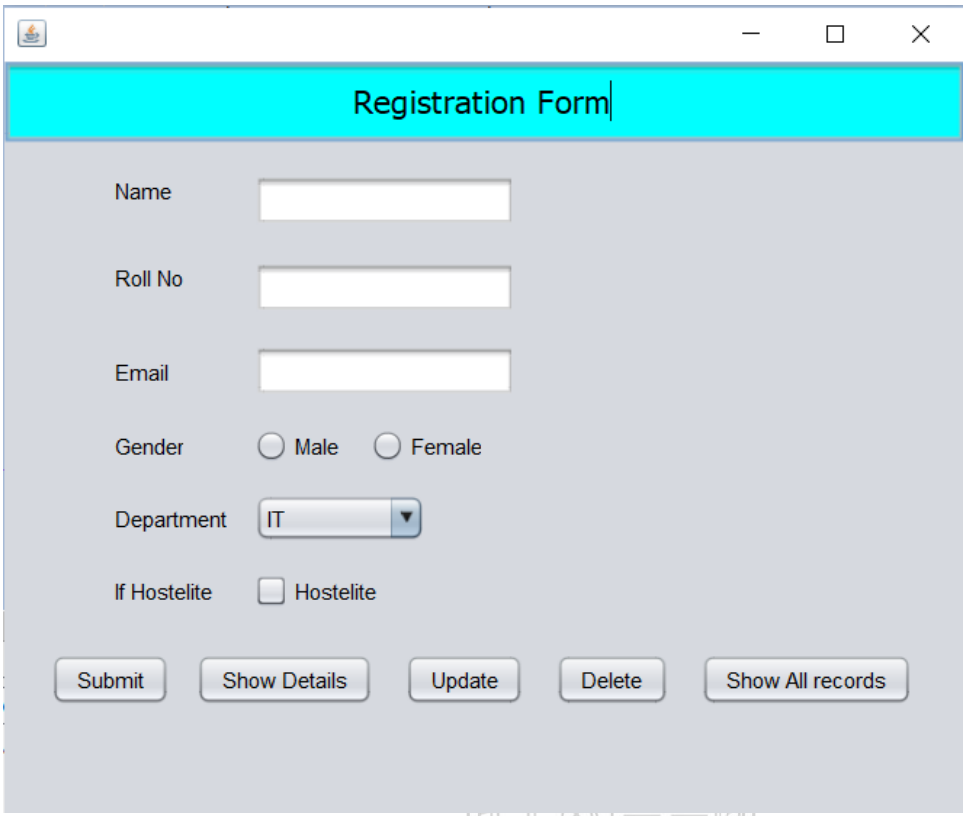
```
            }
            else{
               JOptionPane.showMessageDialog(this,"No record found");
            }
      }
      catch(Exception e){}
   }

   private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
      String str1=jTextField1.getText();
      String str2=jTextField2.getText();
      try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con=(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/form?zeroDateTimeBehavior=C
ONVERT_TO_NULL","root","");
            PreparedStatement ps = (PreparedStatement) con.prepareStatement("delete from
info where RollNo=?");

            ps.setString(1, str2);
            int row=ps.executeUpdate();
             con.close();
             if(row!=0){
             JOptionPane.showMessageDialog(this,"Record Is Successfully Deleted");
             }
             else{
               JOptionPane.showMessageDialog(this,"No record found");
             }
      }
      catch(Exception e){}
   }

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

      try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con=(Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/form?zeroDateTimeBehavior=C
```

```
ONVERT_TO_NULL","root","");
            ResultSet ps1 = ((PreparedStatement) con.prepareStatement("SELECT * FROM
info")).executeQuery();
          String res ="";
          String[] tableHeader = {"Name", "RollNo", "Email",
"Gender","Department","Type"};
          ResultSet rs1 = (((PreparedStatement) con.prepareStatement("SELECT COUNT(*)
FROM info")).executeQuery());
          rs1.next();
          int n = Integer.parseInt(rs1.getString(1));
          String [][] str = new String[n][];
          int i=0;
          while(ps1.next())
          {
             str[i] = new String[6];
             str[i][0] = ps1.getString(1);
             str[i][1] = ps1.getString(2);
             str[i][2] = ps1.getString(3);
             str[i][3] = ps1.getString(4);
             str[i][4] = ps1.getString(5);
             str[i][5] = ps1.getString(6);
             i++;
          }

          JOptionPane.showMessageDialog(this,new JScrollPane(new
JTable(str,tableHeader)));

       }
       catch(Exception e){}
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
       /* Set the Nimbus look and feel */
       //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
       /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
```

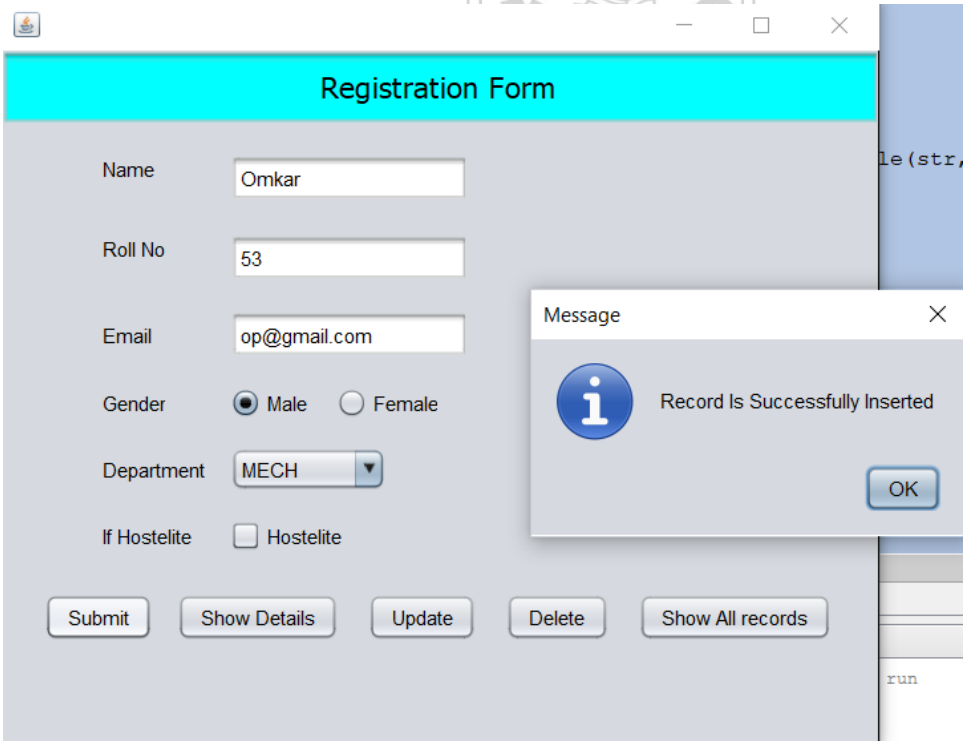
```
feel.
        * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
        */
       try {
           for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
               if ("Nimbus".equals(info.getName())) {
                   javax.swing.UIManager.setLookAndFeel(info.getClassName());
                   break;
               }
           }
       } catch (ClassNotFoundException ex) {

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

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

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

java.util.logging.Logger.getLogger(NewJForm.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
       }
       //</editor-fold>

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

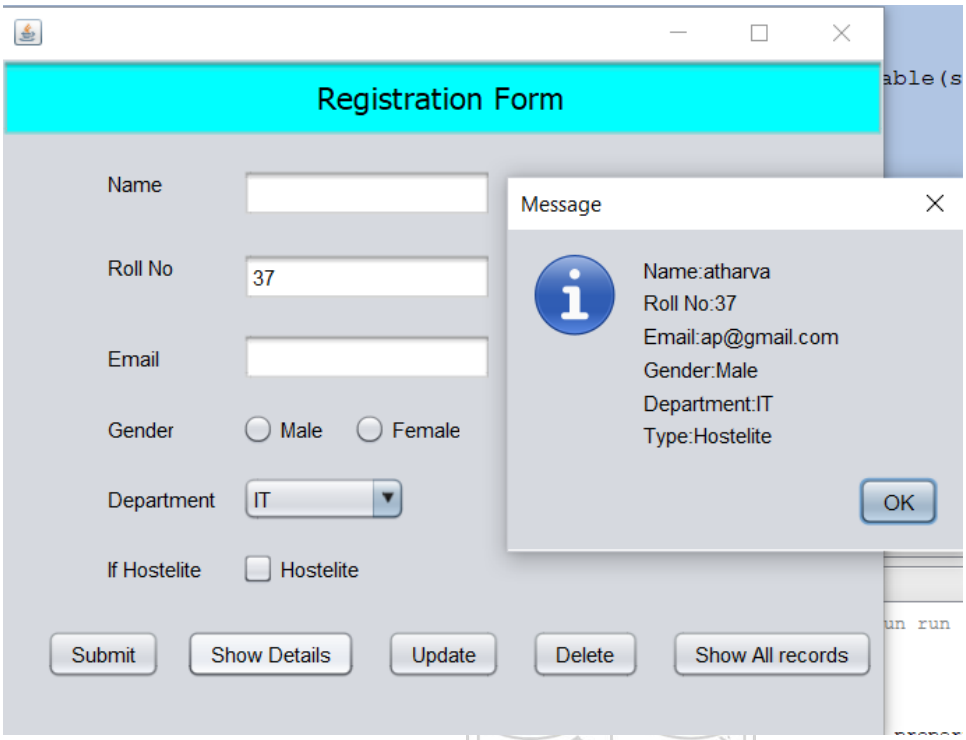
```
    // Variables declaration - do not modify
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JButton jButton5;
    private javax.swing.JCheckBox jCheckBox3;
    private javax.swing.JComboBox<String> jComboBox1;
    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.JLabel jLabel6;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JRadioButton jRadioButton1;
    private javax.swing.JRadioButton jRadioButton2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable jTable1;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField4;
    // End of variables declaration
}
```


**Screen Shots of GUI:**

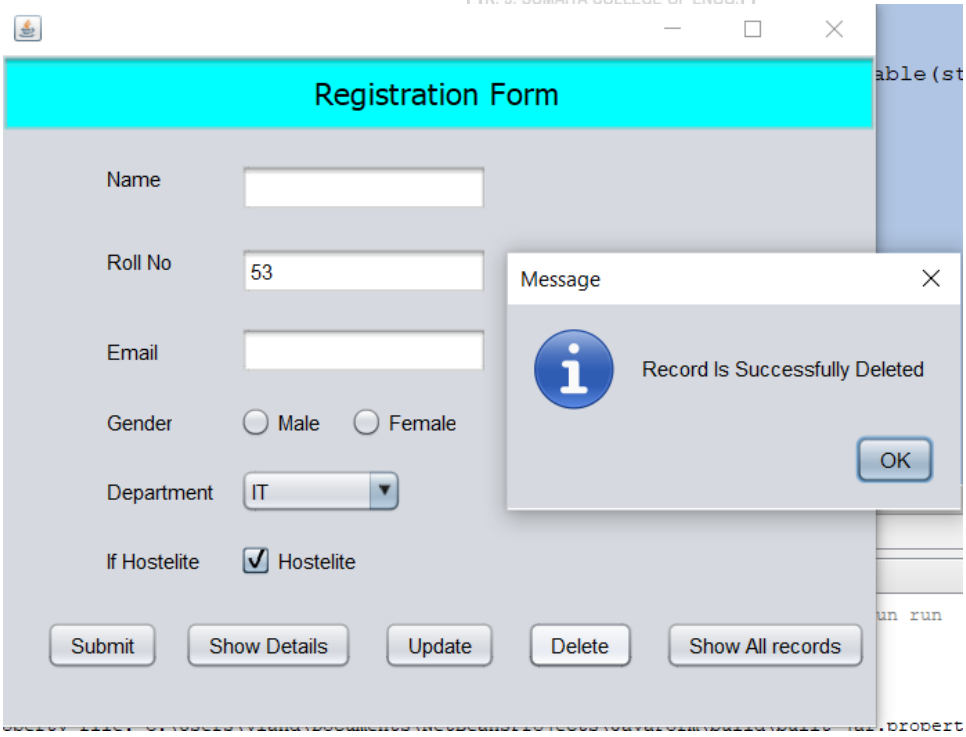


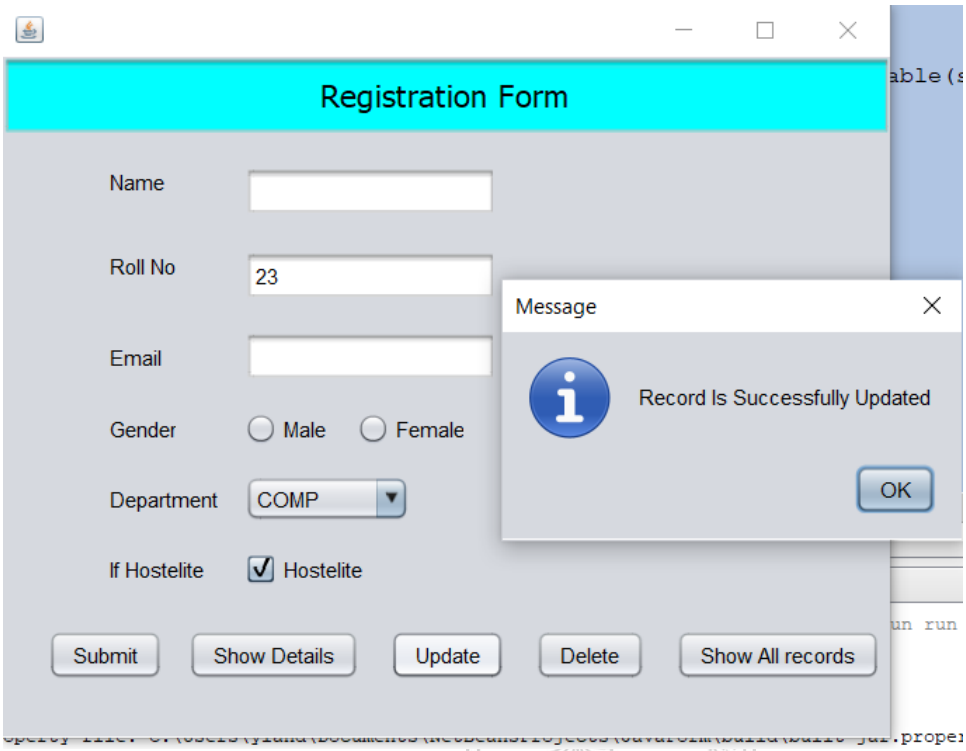


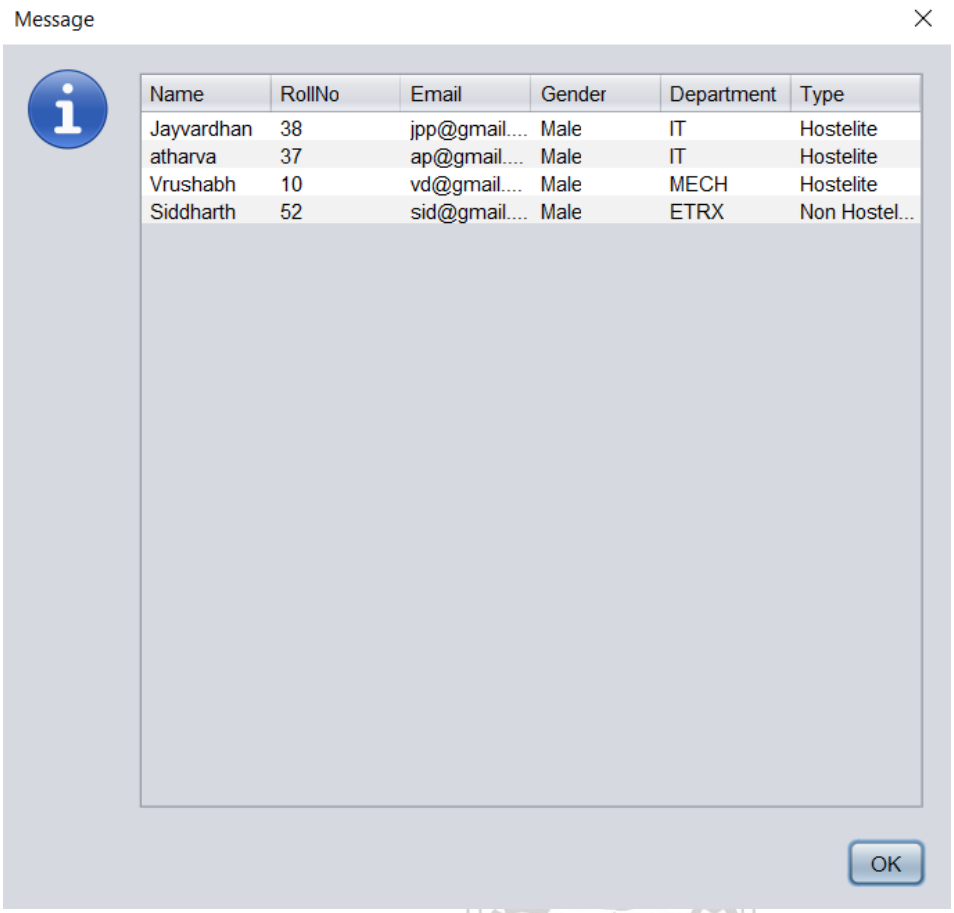








Registration Form
Name
Roll No
23
Email
Gender
Male
Female
Department
COMP
If Hostelite
Hostelite
Submit
Show Details
Update
Delete
Show All records
Message
Record Is Successfully Updated
OK




---

**Questions:**

**1. What is the content pane?**

**Ans:** The content pane is an object created by the Java run time environment. A container has several layers in it. You can think of a layer as a transparent film that overlays the container. In Java Swing, the layer that is used to hold objects is called the content pane. Objects are added to the content pane layer of the container. The getContentPane() method retrieves the content pane layer so that you can add an object to it. The content pane is an object created by the Java run time environment. You do not have to know the name of the content pane to use it. When you use getContentPane(), the content pane object then is substituted there so that you can apply a method to it

---

**Outcomes:** Illustrate the wiring of front end with back end of Java application using database connectivity

**Conclusion: (Conclusion to be based on the objectives and outcomes achieved)**

Designing a form using functions of Swing class was learnt, connection of database with jdbc and about importance of jdbc was also understood and its implementation was also done.

**Grade: AA / AB / BB / BC / CC / CD/DD**

**Signature of faculty in-charge with date**

**References:**

1.By Herbert Schildt ; ” Java The Complete Reference”; 10th Edition 2017; McGraw-Hill Publication
2.By D.T. Editorial Services; “Java 8 Programming Black Book” ; Edition 2015 ; Dreamtech Press2004