

Java Development:

1. Form Design:

The screenshot shows a Java Swing application window titled "Medical Research Center & Hospital". The menu bar includes "File" and "Edit". A vertical toolbar on the left has buttons for "Home", "Patient Entry" (which is selected), and "Receipt". The main panel contains a "Patient entry form" with fields for Registration number, Registration date, Patient Name, Patient Address (with a scroll bar), City, Contact number, Gender (with radio buttons for male and female), and Age. Below these fields is a "Events" button group containing "Update", "Delete", "Save", "Search", "Clear", and "Exit". To the right of the form is a table displaying patient data:

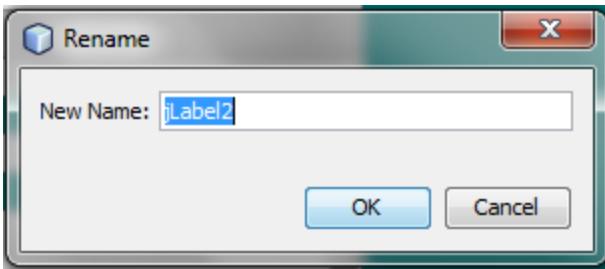
Reg_no	Reg_D...	Patient...	Address	City	Contact...	Gender	Age
16	02.04...	Afsana	aman ...	Ctg	017***	female	26
17	02.04...	Saif	khulsi	Ctg	016***	male	23
15	01.04....	taslima	south ...	ctg	01670...	female	24

2. Edit Text:

Right click on text field and JLabel, have to click on edit text.

3. Change Variable Name:

If we do right click on text field and select this item we will see the following window.

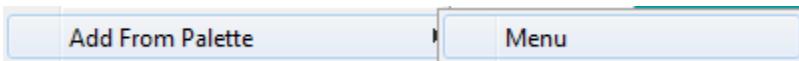


After typing new name have to click 'OK' to change the variable name.

4. Menu bar and sub-menu:

To set the menu item we have to click on File from Swing Menus and drag in form.

If we want to add sub menu we've to right click any of menus and click on



5. Insert information on combo box:

Click on properties keeping mouse on combo box, select model, delete default item from model, insert data if need and click close.

6. Button:

Double click on click button and have to code as follows:

Variable name.setText(""); //for Text Field and tx Text Area.

Variable name.setSelectedItem(""); //for Combo Box.

Action performed code in button for the given form is given below:

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

```
//clear window  
  
regNoText.setText("");  
  
regdateText.setText("");  
  
nameText.setText("");  
  
addrsText.setText("");  
  
cityText.setText("");  
  
numberText.setText("");  
  
genderText.setText("");  
  
ageText.setText("");  
  
}  
  
}
```

7. **Button:**

Double click on close button and place the code “this.dispose();” into the class.

8. **Save Button:**

Action performed code in Save button for the given form is given below:

```
private void saveBtnActionPerformed(java.awt.event.ActionEvent evt) {  
  
    try {  
  
        String str = "insert into  
patient_entry(reg_no,reg_date,name,address,city,contact,gender,age) values(?,  
?,?,?,?,?,?);";  
  
        st = con.prepareStatement(str);  
  
        st.setString(1, regNoText.getText());  
  
        st.setString(2, regdateText.getText());
```

```

st.setString(3, nameText.getText());
st.setString(4, addrsText.getText());
st.setString(5, cityText.getText());
st.setString(6, numberText.getText());
st.setString(7, genderText.getText());
st.setString(8, ageText.getText());
int i = st.executeUpdate();
if (i != 0) {
    JOptionPane.showMessageDialog(this, "Saved");
} else {
    JOptionPane.showMessageDialog(this, "Not Saved");
}
} catch (SQLException ex) {
    JOptionPane.showMessageDialog(null, ex.toString());
}

```

9.Update Button :

```

private void updateBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // update
    try {
        String reg_no = regNoText.getText();

```

```

String str = "UPDATE patient_entry set
reg_no=?,reg_date=?,name=?,address=?,city=?,contact=?,gender=?,age=?
WHERE reg_no=?";

st = (PreparedStatement) con.prepareStatement(str);

st.setString(1, reg_no);
st.setString(2, regdateText.getText());
st.setString(3, nameText.getText());
st.setString(4, addrsText.getText());
st.setString(5, cityText.getText());
st.setString(6, numberText.getText());
st.setString(7, genderText.getText());
st.setString(8, ageText.getText());
st.setString(9, reg_no);

st.executeUpdate();

JOptionPane.showMessageDialog(null, "Data Successfully updated!!!!");

} catch (SQLException ex) {
    JOptionPane.showMessageDialog(null, ex.toString());
}
}

```

10.Search Button:

```
private void searchBtnActionPerformed(java.awt.event.ActionEvent evt) {  
    //search  
    try {  
        String regNo = regNoText.getText();  
        String str = "select * from patient_entry where reg_no = ?";  
        st = con.prepareStatement(str);  
        st.setString(1, regNo);  
        rs = st.executeQuery();  
        while (rs.next()) {  
            String regdate = rs.getString("reg_date");  
            regdateText.setText(regdate);  
  
            String name = rs.getString("name");  
            nameText.setText(name);  
  
            String addrs = rs.getString("address");  
            addrsText.setText(addrs);  
  
            String city = rs.getString("city");  
            cityText.setText(city);  
  
            String phone = rs.getString("contact");  
    }  
}
```

```

        numberText.setText(phone);

        String gendr = rs.getString("gender");
        genderText.setText(gendr);

        String age = rs.getString("age");
        ageText.setText(age);
    }

} catch (SQLException e) {

}

}

```

11.Delete Button:

```

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

    //delete

    try {

        String regNo = regNoText.getText();

        String str = "DELETE FROM patient_entry where reg_no = ? ";

        st = con.prepareStatement(str);

        st.setString(1, regNo);

        int i = st.executeUpdate();

        if (i != 0) {

            JOptionPane.showMessageDialog(this, "Deleted");
        }
    }
}

```

```

} else {

    JOptionPane.showMessageDialog(this, "Not Deleted");

}

} catch (SQLException ex) {

    JOptionPane.showMessageDialog(null, ex.toString());

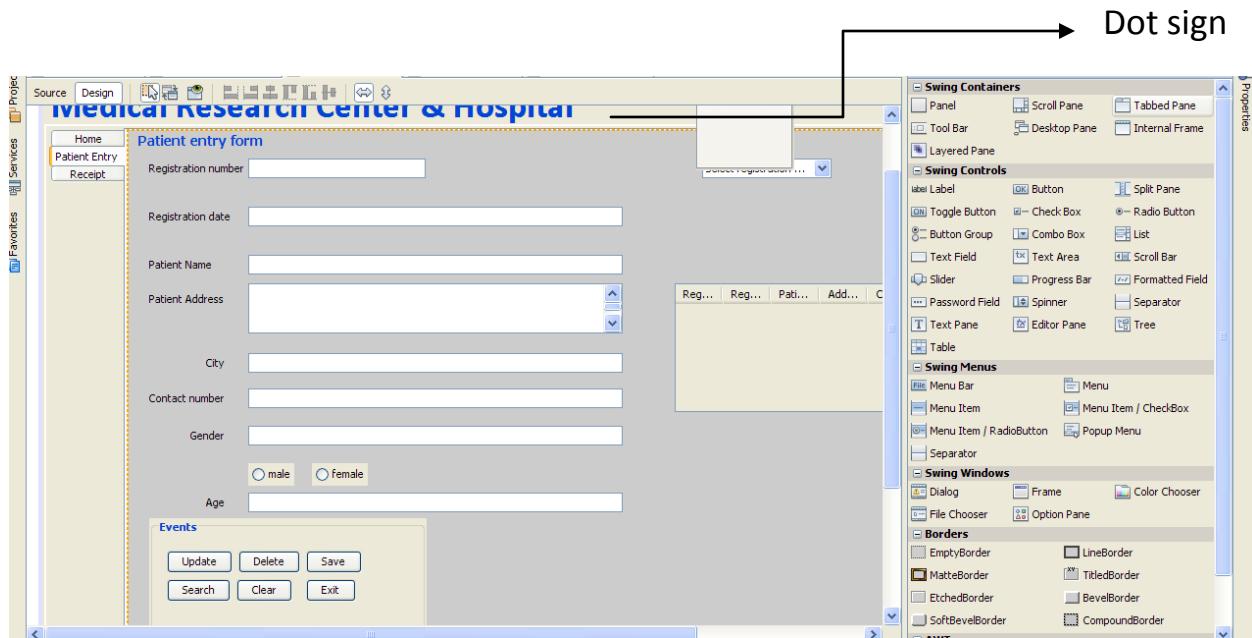
}

}

```

12.Adding tabbed pane:

Draw panel,select tabbed pane palette from swing containers,drag it into the panel and release mouse after seeing the dot sign around panel.



For placing the tabbed panes,right click on tab,click properties and select **tabPlacement** .

13.Combo box searching:

```
private void regComboBoxActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        String regNo = regComboBox.getSelectedItem().toString();  
        String str = "select * from patient_entry where reg_no = ?";  
        st = con.prepareStatement(str);  
        st.setString(1, regNo);  
        rs = st.executeQuery();  
        while (rs.next()) {  
            String regdate = rs.getString("reg_date");  
            regdateText.setText(regdate);  
  
            String name = rs.getString("name");  
            nameText.setText(name);  
  
            String addrs = rs.getString("address");  
            addrsText.setText(addrs);  
  
            String city = rs.getString("city");  
            cityText.setText(city);  
  
            String phone = rs.getString("contact");  
    }  
}
```

```

numberText.setText(phone);

String gendr = rs.getString("gender");
genderText.setText(gendr);

String age = rs.getString("age");
ageText.setText(age);

test = 0;
break;
}

if (test == 1) {
    JOptionPane.showMessageDialog(this, "Not Found");
}

} catch (SQLException e) {
}

}

```

14.Key released searching:

Same code as action performed except different event(key released event).

15.Radio Button select:

In given form,we use two radio button for male and female.

/.....Select female button and place it in text field “Gender”.....*/*

```

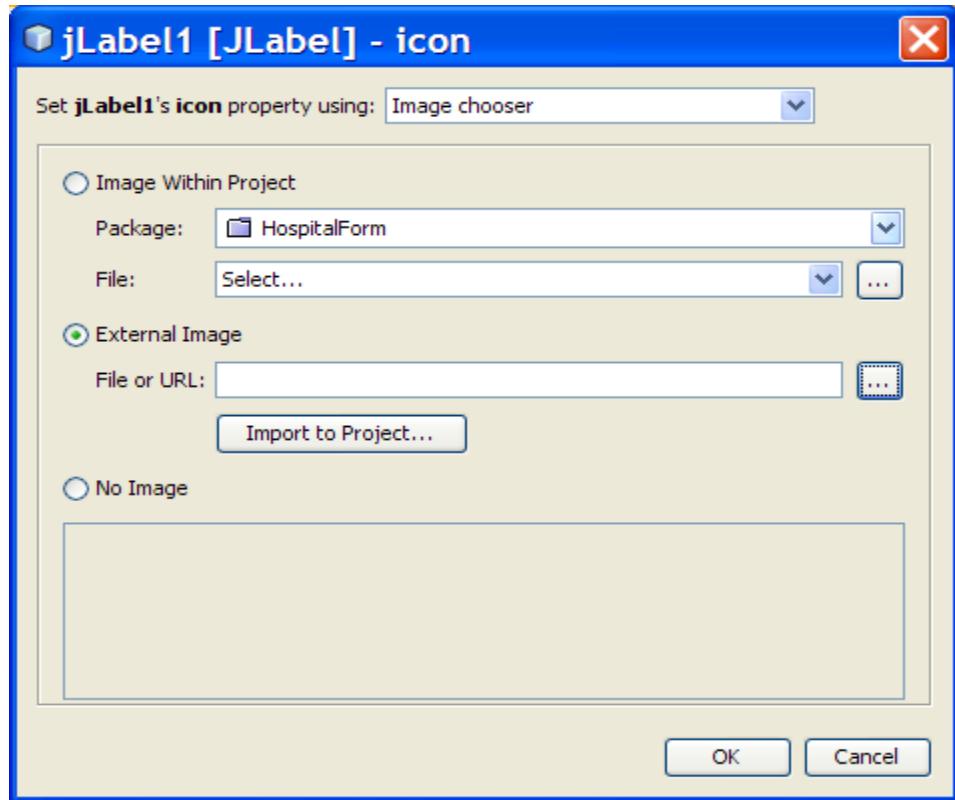
private void femalebtnActionPerformed(java.awt.event.ActionEvent evt) {
    malebtn.setSelected(false);
    genderText.setText("female");
}

/*.....Select male button and place it in text field “Gender”.....*/
private void malebtnActionPerformed(java.awt.event.ActionEvent evt) {
    femalebtn.setSelected(false);
    genderText.setText("male");
}

```

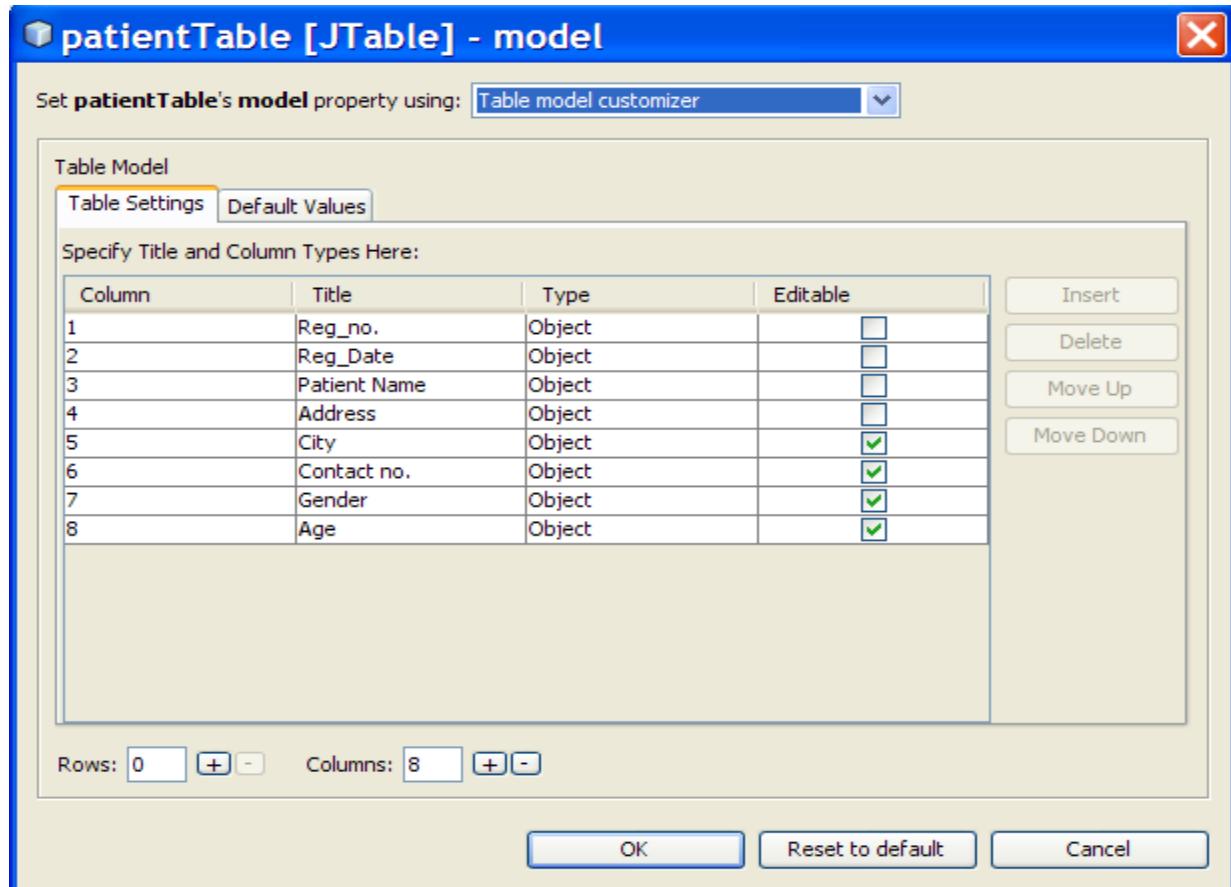
16.Insert image:

Select properties ->icon and follow



17.Insert table:

Select table palette and insert into panel. For inserting column and row we have to right click on table, select properties, click model and select table item.



18.Operation on table:

a.to show table data opening window following code are added within constructor.

```
try {
```

```
    DefaultTableModel model = (DefaultTableModel) patientTable.getModel();  
  
    String sql = "select * from patient_entry";  
  
    st = con.prepareStatement(sql);  
  
    rs = st.executeQuery();
```

```

        while (rs.next()) {

            String regno = rs.getString("reg_no");

            String regdate = rs.getString("reg_date");

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

            String addrs = rs.getString("address");

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

            String phone = rs.getString("contact");

            String gendr = rs.getString("gender");

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

            model.addRow(new Object[]{regno, regdate, name, addrs, city, phone,
gendr, age});

        }

    } catch (Exception e) {

}

```

b.For removing table data following codes are needed.

```

int tbaleClear = patientTable.getRowCount();

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

for (int i = tbaleClear - 1; i >= 0; i--) {

    model.removeRow(i);

}

```

c.Following code shows table data when combo box searching is used.

```
try {
```

```
model = (DefaultTableModel) patientTable.getModel();

String sqr = "select * from patient_entry where reg_no = ?";
st = con.prepareStatement(sqr);
st.setString(1, regComboBox.getSelectedItem().toString());
rs = st.executeQuery();
while (rs.next()) {
    String regno = rs.getString("reg_no");
    String regdate = rs.getString("reg_date");
    String name = rs.getString("name");
    String addrs = rs.getString("address");
    String city = rs.getString("city");
    String phone = rs.getString("contact");
    String gendr = rs.getString("gender");
    String age = rs.getString("age");
    model.addRow(new Object[]{regno, regdate, name, addrs, city, phone,
        gendr, age});
}
} catch (Exception e) {
}
```

19.Merging:

a.Class holding code for database connection:

```
package DBConnection;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import javax.swing.JOptionPane;

public class DBConnection {
    static Connection con;
    public static Connection DBConnect(){
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital",
"root", "");
            System.out.println("Connection Successful");
        } catch (ClassNotFoundException ex) {
            JOptionPane.showMessageDialog(null, ex.toString());
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, ex.toString());
        }
    }
}
```

```
    return con;  
}  
}
```

b.Main class:

main class refers a dialog form “login”.

```
package Main;  
  
public class MainClass {  
  
    public static void main(String arg[]){  
  
        LoginForm Lg = new LoginForm();  
  
        Lg.setVisible(true);  
  
    }  
  
}
```

c.Login form:

login form is opened as dialog frame in same package where main class is situated. We convert it into JFrame.

```
package Main;  
  
import HRMForm.Home;  
  
import dbConnection.DBConnection;  
  
import java.sql.Connection;  
  
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;

public class LoginForm extends javax.swing.JFrame {

    Connection con;
    PreparedStatement st;
    ResultSet rs = null;

    public LoginForm() {
        initComponents();
        /*.....connection.....*/
        con=(Connection) DBConnection.DBConnect();
    }

    private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
        try{
            String username=(String) userComboBox.getSelectedItem();
            String psswrd = psswrdText.getText();

            String str = "select * from login where user= ? and password= ?";
            st = (PreparedStatement) con.prepareStatement(str);
            st.setString(1, username);
            st.setString(2, psswrd);
        }
    }
}
```

```

rs=st.executeQuery();

if(rs.next())

{

    Home hm = new Home(); //refers home page.

    hm.setVisible(true);

    this.dispose();           //disposing login window

}

else JOptionPane.showMessageDialog(null,"Invalid user name or
password.");

}

catch (SQLException ex) {

    JOptionPane.showMessageDialog(null, ex.toString());

}

}

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

/*.....cancel login.....*/

this.dispose();

}

/*.....view window at center....*/

private void formWindowOpened(java.awt.event.WindowEvent evt) {

    this.setLocationRelativeTo(null);

}

```

```
private void psswrdTActionPerformed(java.awt.event.ActionEvent evt) {  
    try{  
        String username=(String) userComboBox.getSelectedItem();  
        String psswrD = psswrDText.getText();  
  
        String str = "select * from login where user= ? and password= ?";  
        st = (PreparedStatement) con.prepareStatement(str);  
  
        st.setString(1, username);  
        st.setString(2, psswrD);  
  
        rs=st.executeQuery();  
        if(rs.next())  
        {  
            Home hm = new Home();  
            hm.setVisible(true);  
        }  
        else JOptionPane.showMessageDialog(null,"Invalid user name or  
password.");  
    }  
}
```

```
        catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, ex.toString());
        }
    }

    public static void main(String args[]) {
        /* Create and display the dialog */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                LoginForm dialog = new LoginForm();

                dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                    @Override
                    public void windowClosing(java.awt.event.WindowEvent e) {
                        System.exit(0);
                    }
                });
                dialog.setVisible(true);
            }
        });
    }
}
```

```

// Variables declaration - do not modify

private javax.swing.JButton btnLogin;

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JPasswordField psswrdTText;

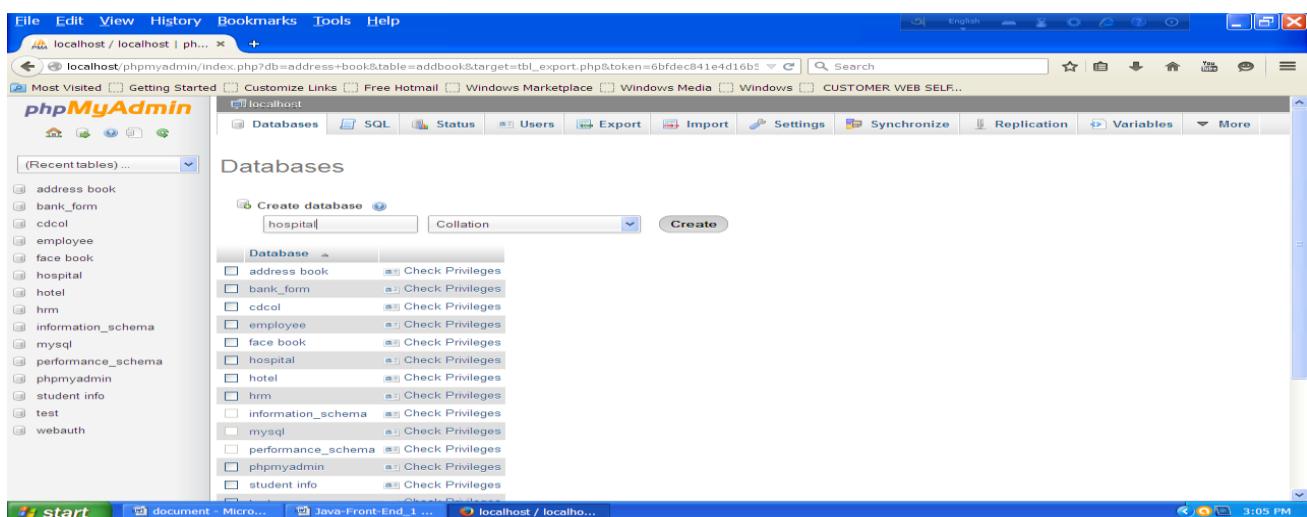
private javax.swing.JComboBox userComboBox;

// End of variables declaration
}

```

20. Link Java Form to the MySQL Database:

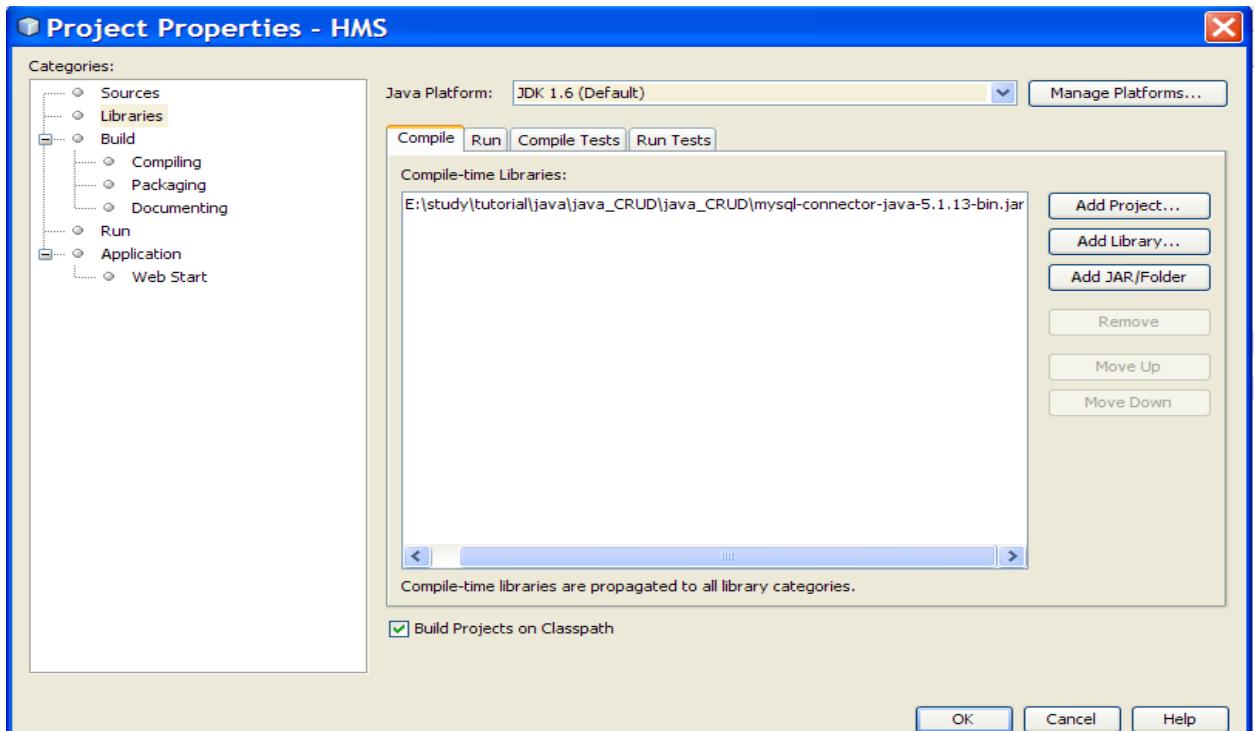
To link java form with MySQL Database, at first have to create a database which have various table name.



Attribute names of table have to be same as attributes/fields of java form.

Now a jar file is needed to connect database with java form.jar file needs to be added with main project. For adding, right click on project, select properties, click libraries and select **Add JAR/Folder** placed in right side.

Then have to select **mysql-connector-java-5.1.13-bin.jar** and click OK.



A Simple Java Program :

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;
public class hospital extends javax.swing.JFrame {
    Connection con;
    ResultSet rs;
    PreparedStatement pst;
    public hospital() {
        setTitle("Hospital Management System");
        initComponents();
    }
    /*.....Connection.....*/
    try {
        Class.forName("com.mysql.jdbc.Driver");
con = DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital",
"root", "");
        System.out.println("Connection Successful");
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, ex.toString());
    }
}
```

//.....combo box data for patient.....//

```
try {  
  
    String str = "select distinct(reg_no) from patient_entry";  
  
    st = con.prepareStatement(str);  
  
    rs = st.executeQuery();  
  
    while (rs.next()) {  
  
        String regno = rs.getString("reg_no");  
  
        regComboBox.addItem(regno);  
  
    }  
  
} catch (SQLException e) {  
  
}  
  
}
```

//worker show table data

```
try {  
  
    DefaultTableModel model = (DefaultTableModel) patientTable.getModel();  
  
    String sql = "select * from patient_entry";  
  
    st = con.prepareStatement(sql);  
  
    rs = st.executeQuery();  
  
    while (rs.next()) {  
  
        String regno = rs.getString("reg_no");  
  
        String regdate = rs.getString("reg_date");  
  
        String name = rs.getString("name");  
  
    }  
  
}  
  
}
```

```

String addrs = rs.getString("address");

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

String phone = rs.getString("contact");

String gendr = rs.getString("gender");

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

model.addRow(new Object[]{regno, regdate, name, addrs, city, phone,
gendr, age});

}

} catch (Exception e) {

}

}

```

Reg_no	Reg_D...	Patient...	Address	City	Contact...	Gender	Age
16	02.04....	Afsana	aman ...	Ctg	017***...	female	26
17	02.04...	Saif	khulsi	Ctg	016***...	male	23
15	01.04....	taslima	south ...	ctg	01670...	female	24

/.....View new window.....*/*

```

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

Info io = new Info();

io.setVisible(true);

}

```

```

/*.....Calculate total amount....*/

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

    try {

        int total=
        Integer.parseInt(DText.getText())+Integer.parseInt(ecgText.getText())+Integer.par
        seInt(testText.getText())+Integer.parseInt(injctnText.getText());

        feesText.setText(String.valueOf(total));

    } catch (NumberFormatException ex) {

        //do something like telling the user the input is not a number

    }

}

/*.....Clear Button....*/

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

    // regText.setText("");
    Textname.setText("");
    DText.setText("0");
    ecgText.setText("0");
    injctnText.setText("0");
    testText.setText("0");
    feesText.setText("");

}

/*.....Save Button.....*/

```

```
private void addBtnActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        String str = "insert into  
receipt_entry(reg_no,name,diagnosis_fees,ECG_fees,inj_fees,test_fees,total)  
values(?, ?, ?, ?, ?, ?, ?)";  
  
        st = con.prepareStatement(str);  
        regText.getText();  
        st.setString(1, regText.getText());  
        st.setString(2, Textname.getText());  
        st.setString(3, DText.getText());  
        st.setString(4, ecgText.getText());  
        st.setString(5, injctnText.getText());  
        st.setString(6, testText.getText());  
        st.setString(7, feesText.getText());  
        int i = st.executeUpdate();  
  
        if (i != 0) {  
            JOptionPane.showMessageDialog(this, "Saved");  
            int j=Integer.parseInt(regText.getText());  
            j++;  
            regText.setText(String.valueOf(j));  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

```

} else {

    JOptionPane.showMessageDialog(this, "Not Saved");

}

} catch (SQLException ex) {

    JOptionPane.showMessageDialog(null, ex.toString());

}

}

/*.....Exit Button.....*/

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

    this.dispose();

}

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

    JOptionPane.showMessageDialog(null, "Wait for printing.....");

}

/*.....find total fees in text field “total”.....*/

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

    try {

        int total=

        Integer.parseInt(DText.getText())+Integer.parseInt(ecgText.getText())+Integer.par-
        selInt(testText.getText())+Integer.parseInt(injctnText.getText());
    }
}

```

```

feesText.setText(String.valueOf(total));

} catch (NumberFormatException ex) {

//do something like telling the user the input is not a number

}

}

/*.....find total fees in text field “total” by clicking mouse.....*/

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

try {

int total=

Integer.parseInt(DText.getText())+Integer.parseInt(ecgText.getText())+Integer.par
seInt(testText.getText())+Integer.parseInt(injctnText.getText());

feesText.setText(String.valueOf(total));

} catch (NumberFormatException ex) {

//do something like telling the user the input is not a number

}

}

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

try {

int total=

Integer.parseInt(DText.getText())+Integer.parseInt(ecgText.getText())+Integer.par
seInt(testText.getText())+Integer.parseInt(injctnText.getText());

```

```
feesText.setText(String.valueOf(total));

} catch (NumberFormatException ex) {

//do something like telling the user the input is not a number

}

}
```

/*.....combo box searching.....*/

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

/*.....clear table data.....*/

int tbaleClear = patientTable.getRowCount();

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

for (int i = tbaleClear - 1; i >= 0; i--) {

model.removeRow(i);

}

int test = 1;

/*....show data by combo box searching.....*/

try {

String regNo = regComboBox.getSelectedItem().toString();

String str = "select * from patient_entry where reg_no = ?";

st = con.prepareStatement(str);

st.setString(1, regNo);

rs = st.executeQuery();
```

```
while (rs.next()) {  
  
    String regdate = rs.getString("reg_date");  
  
    regdateText.setText(regdate);  
  
    String name = rs.getString("name");  
  
    nameText.setText(name);  
  
    String addrs = rs.getString("address");  
  
    addrsText.setText(addrs);  
  
    String city = rs.getString("city");  
  
    cityText.setText(city);  
  
    String phone = rs.getString("contact");  
  
    numberText.setText(phone);  
  
    String gendr = rs.getString("gender");  
  
    genderText.setText(gendr);  
  
    String age = rs.getString("age");  
  
    ageText.setText(age);  
  
  
    test = 0;  
  
    break;  
}  
  
if (test == 1) {  
  
    JOptionPane.showMessageDialog(this, "Not Found");  
}
```

```
        } catch (SQLException e) {  
    }  
  
    try {  
        model = (DefaultTableModel) patientTable.getModel();  
  
        String sqr = "select * from patient_entry where reg_no = ?";  
        st = con.prepareStatement(sqr);  
        st.setString(1, regComboBox.getSelectedItem().toString());  
        rs = st.executeQuery();  
        while (rs.next()) {  
            String regno = rs.getString("reg_no");  
            String regdate = rs.getString("reg_date");  
            String name = rs.getString("name");  
            String addrs = rs.getString("address");  
            String city = rs.getString("city");  
            String phone = rs.getString("contact");  
            String gendr = rs.getString("gender");  
            String age = rs.getString("age");  
        }  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
}
```

```
    model.addRow(new Object[]{regno, regdate, name, addrs, city, phone,  
gendr, age});
```

```
}
```

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

```
}
```

```
}
```

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

```
//clear window
```

```
regNoText.setText("");
```

```
regdateText.setText("");
```

```
nameText.setText("");
```

```
addrsText.setText("");
```

```
cityText.setText("");
```

```
numberText.setText("");
```

```
genderText.setText("");
```

```
ageText.setText("");
```

```
//clear table
```

```
int tbaleClear = patientTable.getRowCount();

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

for (int i = tbaleClear - 1; i >= 0; i--) {

    model.removeRow(i);

}

}

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

    int tbaleClear = patientTable.getRowCount();

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

    for (int i = tbaleClear - 1; i >= 0; i--) {

        model.removeRow(i);

    }

    //search

    try {

        String regNo = regNoText.getText();

        String str = "select * from patient_entry where reg_no = ?";


```

```
st = con.prepareStatement(str);
st.setString(1, regNo);

rs = st.executeQuery();

while (rs.next()) {

    String regdate = rs.getString("reg_date");
    regdateText.setText(regdate);

    String name = rs.getString("name");
    nameText.setText(name);

    String addrs = rs.getString("address");
    addrsText.setText(addrs);

    String city = rs.getString("city");
    cityText.setText(city);

    String phone = rs.getString("contact");
    numberText.setText(phone);
```

```
String gendr = rs.getString("gender");
genderText.setText(gendr);

String age = rs.getString("age");
ageText.setText(age);

}

} catch (SQLException e) {

}

///patient show table data

try {
    model = (DefaultTableModel) patientTable.getModel();

    String sql = "select * from patient_entry where reg_no = ?";
    st = con.prepareStatement(sql);
    st.setString(1, regComboBox.getSelectedItem().toString());
    rs = st.executeQuery();
    while (rs.next()) {
        String regno = rs.getString("reg_no");
        String regdate = rs.getString("reg_date");
    }
}
```

```
String name = rs.getString("name");

String addrs = rs.getString("address");

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

String phone = rs.getString("contact");

String gendr = rs.getString("gender");

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

model.addRow(new Object[]{regno, regdate, name, addrs, city, phone,
gendr, age});

}

} catch (Exception e) {

}

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

//exit

this.dispose();

}

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

```
//clear table

int tbaleClear = patientTable.getRowCount();

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

for (int i = tbaleClear - 1; i >= 0; i--) {

    model.removeRow(i);

}

//save

try {

    String str = "insert into
patient_entry(reg_no,reg_date,name,address,city,contact,gender,age) values(?,?
?, ?, ?, ?, ?, ?)";

    st = con.prepareStatement(str);

    st.setString(1, regNoText.getText());
    st.setString(2, regdateText.getText());
    st.setString(3, nameText.getText());
    st.setString(4, addrsText.getText());
    st.setString(5, cityText.getText());
    st.setString(6, numberText.getText());
    st.setString(7, genderText.getText());
    st.setString(8, ageText.getText());
}
```

```
int i = st.executeUpdate();

if (i != 0) {
    JOptionPane.showMessageDialog(this, "Saved");
} else {
    JOptionPane.showMessageDialog(this, "Not Saved");
}

} catch (SQLException ex) {
    JOptionPane.showMessageDialog(null, ex.toString());
}

}

//save in table
try {
    model = (DefaultTableModel) patientTable.getModel();

    String sql = "select * from worker where Worker_ID = ?";
    st = con.prepareStatement(sql);
    st.setString(1, regNoText.getText());
}
```

```
rs = st.executeQuery();

while (rs.next()) {

    String regno = rs.getString("reg_no");

    String regdate = rs.getString("reg_date");

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

    String addrs = rs.getString("address");

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

    String phone = rs.getString("contact");

    String gendr = rs.getString("gender");

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



    model.addRow(new Object[]{regno, regdate, name, addrs, city, phone,
gendr, age});

}

} catch (Exception e) {

}

}

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

//delete
```

```
try {  
    String regNo = regNoText.getText();  
  
    String str = "DELETE FROM patient_entry where reg_no = ? ";  
  
    st = con.prepareStatement(str);  
    st.setString(1, regNo);  
  
    int i = st.executeUpdate();  
  
    if (i != 0) {  
        JOptionPane.showMessageDialog(this, "Deleted");  
    } else {  
        JOptionPane.showMessageDialog(this, "Not Deleted");  
    }  
  
} catch (SQLException ex) {  
    JOptionPane.showMessageDialog(null, ex.toString());  
}  
}
```

```
}
```

```
private void updateBtnActionPerformed(java.awt.event.ActionEvent evt) {  
    // update  
  
    try {  
        String reg_no = regNoText.getText();  
  
        String str = "UPDATE patient_entry set  
reg_no=?,reg_date=?,name=?,address=?,city=?,contact=?,gender=?,age=?  
WHERE reg_no=?";  
        st = (PreparedStatement) con.prepareStatement(str);  
  
        st.setString(1, reg_no);  
        st.setString(2, regdateText.getText());  
        st.setString(3, nameText.getText());  
        st.setString(4, addrsText.getText());  
        st.setString(5, cityText.getText());  
        st.setString(6, numberText.getText());  
        st.setString(7, genderText.getText());  
        st.setString(8, ageText.getText());  
        st.setString(9, reg_no);  
    }  
}
```

```

        st.executeUpdate();

        JOptionPane.showMessageDialog(null, "Data Successfully updated!!!!");

    } catch (SQLException ex) {

        JOptionPane.showMessageDialog(null, ex.toString());

    }

}

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

    malebtn.setSelected(false);

    genderText.setText("female");

}

/*.....select radio button.....*/

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

    femalebtn.setSelected(false);

    genderText.setText("male");

}

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

    /*.....searching data by key released.....*/

    try {

```

```
String regno = regNoText.getText();

String str = "select * from patient_entry where reg_no = ?";
st = con.prepareStatement(str);
st.setString(1, regno);

rs = st.executeQuery();

while (rs.next()) {

    String regdate = rs.getString("reg_date");
    regdateText.setText(regdate);

    String name = rs.getString("name");
    nameText.setText(name);

    String addrs = rs.getString("address");
    addrsText.setText(addrs);

    String city = rs.getString("city");
    cityText.setText(city);
```

```
String phone = rs.getString("contact");
numberText.setText(phone);

String gendr = rs.getString("gender");
genderText.setText(gendr);

String age = rs.getString("age");
ageText.setText(age);

}

} catch (SQLException e) {

}

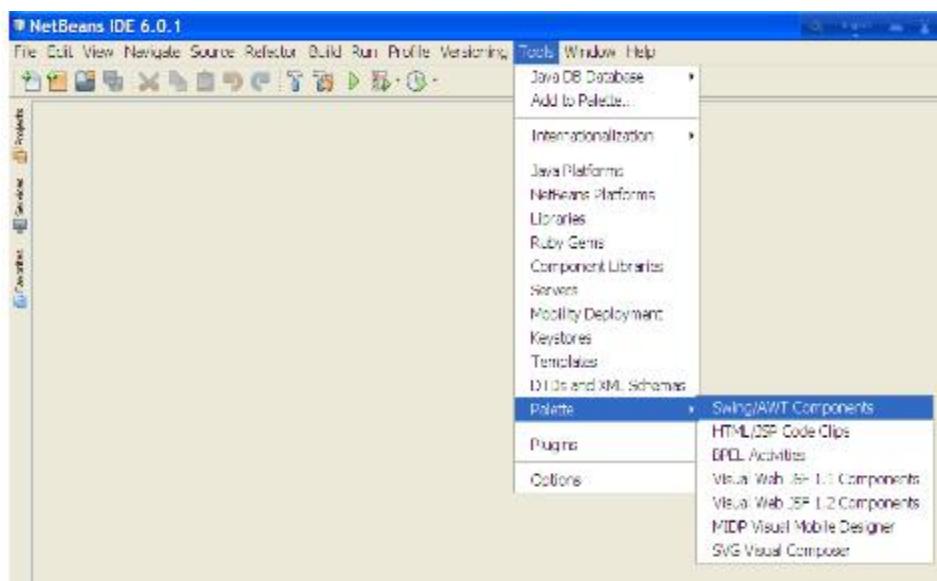
private void regNoTextActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
}
```

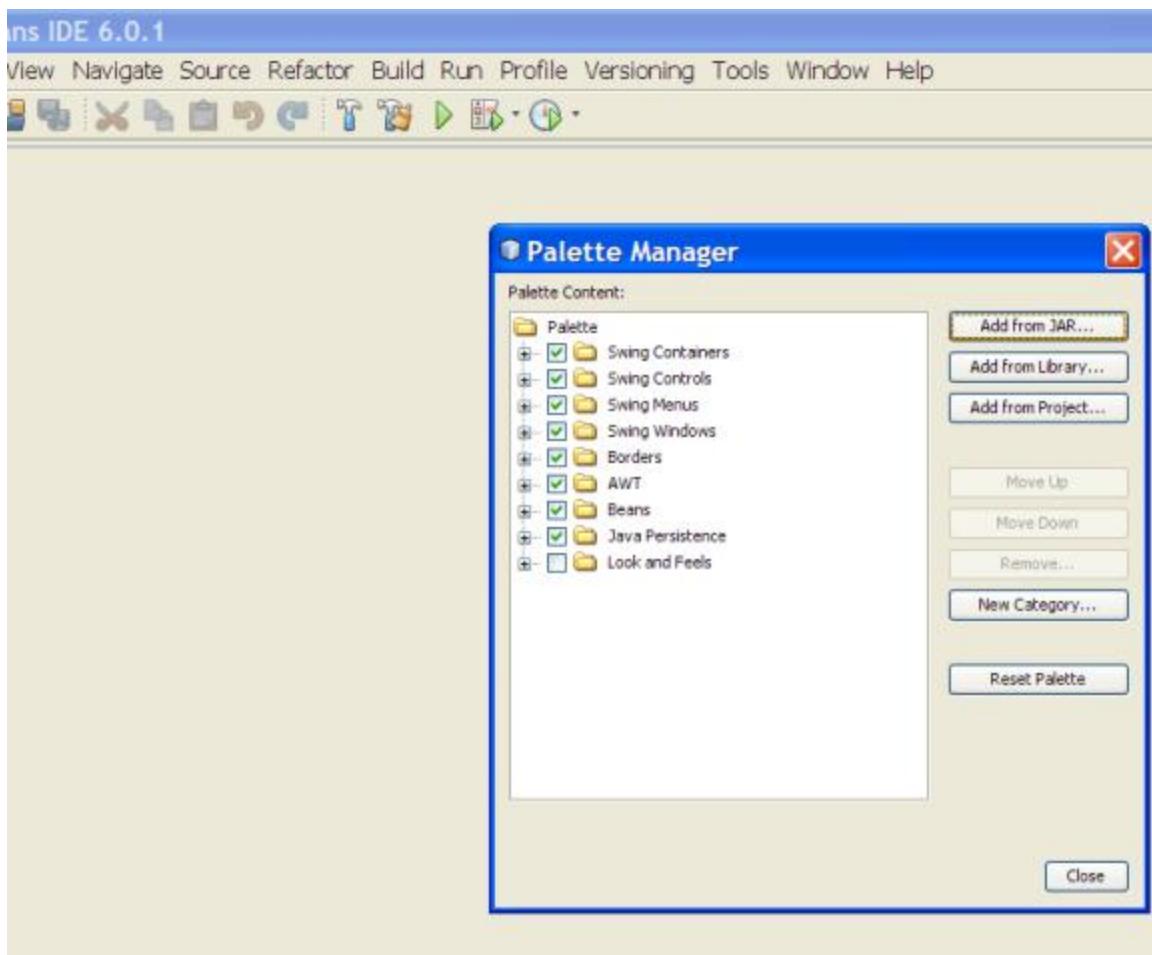
```
private void regTextActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}  
}
```

```
public static void main(String args[]) {  
    java.awt.EventQueue.invokeLater(new Runnable() {  
  
        public void run() {  
            new hospital().setVisible(true);  
        }  
    });  
}
```

jCalender jar file adding in netbeans

tools->palette->component->add jar(all file)





Showing current date and date format:

Following code are written globally below JFrame,

```
Public class university extends javax.swing.JFrame {
```

```
    Date now=new Date(); //....current date
```

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

Showing date in jtable:

```

190
191     //.....JTable For Student - show table data.....//
192
193     try{
194
195         DefaultTableModel model =(DefaultTableModel)StudentjTable.getModel();
196
197
198         String sqlstudent="select * from Student";
199         pst=con.prepareStatement(sqlstudent);
200         rs=pst.executeQuery();
201         while(rs.next()){
202             String sID=rs.getString("Student_ID");
203             String sName=rs.getString("Student_Name");
204             String sSalary=rs.getString("Salary");
205             String sBloodGroup=rs.getString("BloodGroup");
206             String sPhone_Number=rs.getString("Phone_Number");
207             String sAddress=rs.getString("Address");
208             java.sql.Date sJoinDate=rs.getDate("Join_Date");
209
210             model.addRow(new Object[]{sID,sName,sSalary,sBloodGroup,sPhone_Number,sAddress,sJoinDate});
211
212         }
213
214     }catch(Exception e){
215 }

```

Generating auto number:

#write the code inside constructor:

```

try{

String str="select COALESCE(MAX(id),0) from employee";

st = (PreparedStatement) con.prepareStatement(str);

rs=st.executeQuery();

while(rs.next())

{



String empid=rs.getString("COALESCE(MAX(id),0)");

int id=Integer.parseInt(empid)+1;

```

```
    idText.setText(Integer.toString(id));
}

}catch(SQLException e)

{
}
```

#write same code inside save button...

***Image:**

#add thumbnails jar file.

#write upto constructor:

```
String filename;
```

```
byte[] Person_image;
```

```
private ImageIcon imagedata;
```

#write the code inside upload button:

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

    JFileChooser chooser = new JFileChooser();

    chooser.showOpenDialog(null);

    File f = chooser.getSelectedFile();

    filename = f.getAbsolutePath();

    imgLocation.setText(filename);

    try {

        File image1 = new File(filename);
```

```
BufferedImage bufferedImage = ImageIO.read(image1);

BufferedImage thumbnail = Thumbnails.of(bufferedImage).size(150,
150).asBufferedImage();

ByteArrayOutputStream bos = new ByteArrayOutputStream();

ImageIO.write(thumbnail,"jpeg",bos);

ByteArrayInputStream is = new ByteArrayInputStream(bos.toByteArray());

byte[] buf = new byte[1024];

for(int readNum; (readNum=is.read(buf))!= -1;

{

    bos.write(buf, 0, readNum);

    System.out.println("Read "+readNum + "bytes");

}

Person_image = bos.toByteArray();

// Image image=bufferedImage.toImage(true); //this generates an image file

ImageIcon icon = new ImageIcon(thumbnail);
```

```
location.setIcon(icon);
```

```
}
```

```
catch(Exception e)
```

```
{
```

```
}
```

```
}
```

#save uploaded image:

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

```
    try {
```

```
        String str = "insert into  
employee(id,name,bdate,gender,address,contact,mail,dept,grade,jdate,pic)  
values(?,?,?,?,?,?,?,?,?,?)";
```

```
        st = (PreparedStatement) con.prepareStatement(str);
```

```
        java.sql.Date jDate = new java.sql.Date(dateJoin.getDate().getTime());
```

```
        java.sql.Date bDate = new java.sql.Date(birthDate.getDate().getTime());
```

```
        st.setString(1, idText.getText());
```

```
        st.setString(2, nameText.getText());
```

```
        st.setDate(3, bDate);
```

```
        st.setString(4, genderText.getText());
```

```
        st.setString(5, addrsText.getText());
```

```
        st.setString(6, contctText.getText());
```

```

        st.setString(7, mailText.getText());
        st.setString(8, deptCombo.getSelectedItem().toString());
        st.setString(9, gradeCombo.getSelectedItem().toString());
        st.setDate(10, jDate);
        st.setBytes(11, Person_image);
        st.executeUpdate();

        JOptionPane.showMessageDialog(null, "Data Successfully Inserted!!!!");
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, ex.toString());
    }
}

```

#search image:

```

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

    int test=1;

    try{
        String id=JOptionPane.showInputDialog("Enter the employee ID");

        String str = "select * from employee where id = ?";

        st = (PreparedStatement) con.prepareStatement(str);

        st.setString(1, id);

        rs = st.executeQuery();
    }
}

```

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

    idText.setText(id);

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

    nameText.setText(name);

    Date bDate = rs.getDate("bDate");

    birthDate.setDate(bDate);

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

    genderText.setText( gender);

    String addrs = rs.getString("address");

    addrsText.setText( addrs);

    String contct = rs.getString("contact");

    contctText.setText( contct);

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

    mailText.setText( mail);

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

    deptCombo.setSelectedItem( dept);

    String grd= rs.getString("grade");

    gradeCombo.setSelectedItem( grd);

    Date join = rs.getDate("jDate");

    dateJoin.setDate(join);
```

```
byte[]imaged=rs.getBytes("pic");

imagedata=new ImageIcon(imaged);

img_Label.setIcon(imagedata);

test=0;

break;

}

if(test==1){

JOptionPane.showMessageDialog(this, "Not Found");

}

}catch(SQLException e){

}

}
```

#delete image :

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

try {

String str = "delete from employee ";

st = (PreparedStatement) con.prepareStatement(str);

st.executeUpdate();

idText.setText("");

nameText.setText("");
}
```

```

birthDate.setDate(null);

genderText.setText("");

addrsText.setText("");

contctText.setText("");

mailText.setText("");

deptCombo.setSelectedItem("CSE");

gradeCombo.setSelectedItem("A");

dateJoin.setDate(null);

JOptionPane.showMessageDialog(null, "Data deleted!!!!");

} catch (SQLException ex) {

    JOptionPane.showMessageDialog(null, ex.toString());

}

}

```

#update image:

```

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

    try {

        String id = idText.getText();

        String str = "UPDATE employee set
name=?,bdate=?,gender=?,address=?,contact=?,mail=?,dept=?,grade=?,jDate=?,p
ic=? WHERE id=?";

        st = (PreparedStatement) con.prepareStatement(str);

```

```
java.sql.Date jDate = new java.sql.Date(dateJoin.getDate().getTime());  
java.sql.Date bDate = new java.sql.Date(birthDate.getDate().getTime());  
st.setString(11, id);  
st.setString(1, nameText.getText());  
st.setDate(2, bDate);  
st.setString(3, genderText.getText());  
st.setString(4, addrsText.getText());  
st.setString(5, contctText.getText());  
st.setString(6, mailText.getText());  
st.setString(7, deptCombo.getSelectedItem().toString());  
st.setString(8, gradeCombo.getSelectedItem().toString());  
st.setDate(9, jDate);  
st.setBytes(10, Person_image);  
st.executeUpdate();  
JOptionPane.showMessageDialog(null, "Data Successfully updated!!!!");  
} catch (SQLException ex) {  
    JOptionPane.showMessageDialog(null, ex.toString());  
}  
}
```

***Sample:**

