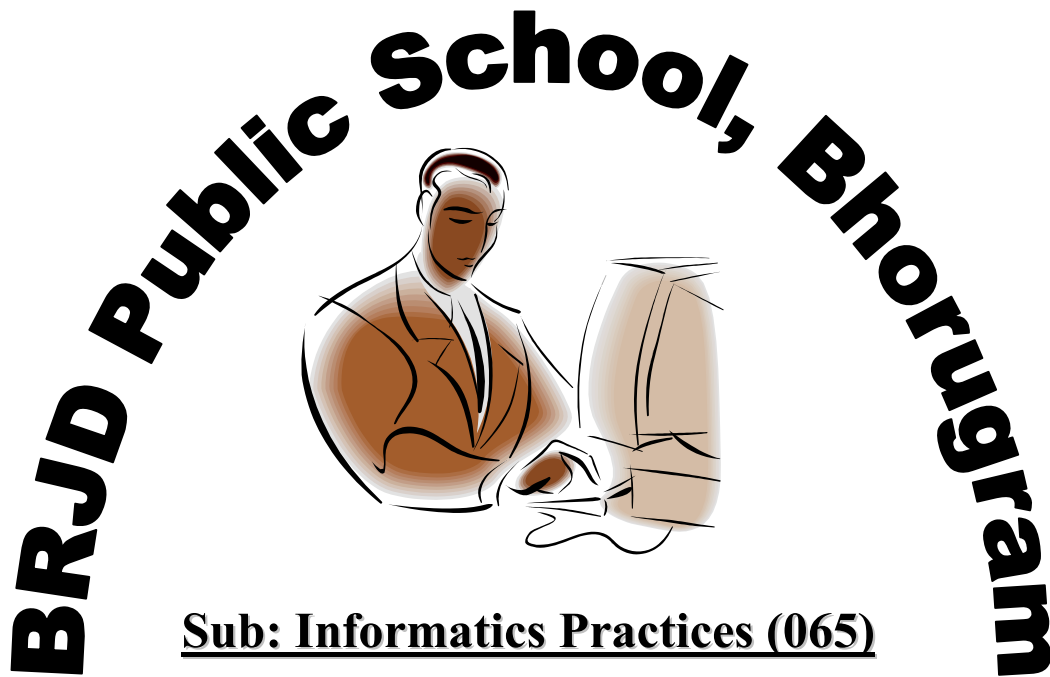


AISSCE 2017-18

PROJECT REPORT

School Management System



**Submitted By: -
Paramjeet
R.NO: 1631331
CLASS XII**

**Submitted To:-
MR. M. K. Gupta
HOD- Computer Dept.**

Acknowledgement

*I am very grateful to my guide and Teacher **Mr. Mukesh Kumar Gupta** under whose guidance I was able to perform my Project work. It was his devoted efforts and keen interest, which helped me in the completion of my Practical.*

I am also thankful to Mr. M. K. Gupta and Mr. L. S. Mehta for helping me in completion of my Project work.

***Name- PARAMJEET
R.NO. 1631331***

Class- XII

Certificate

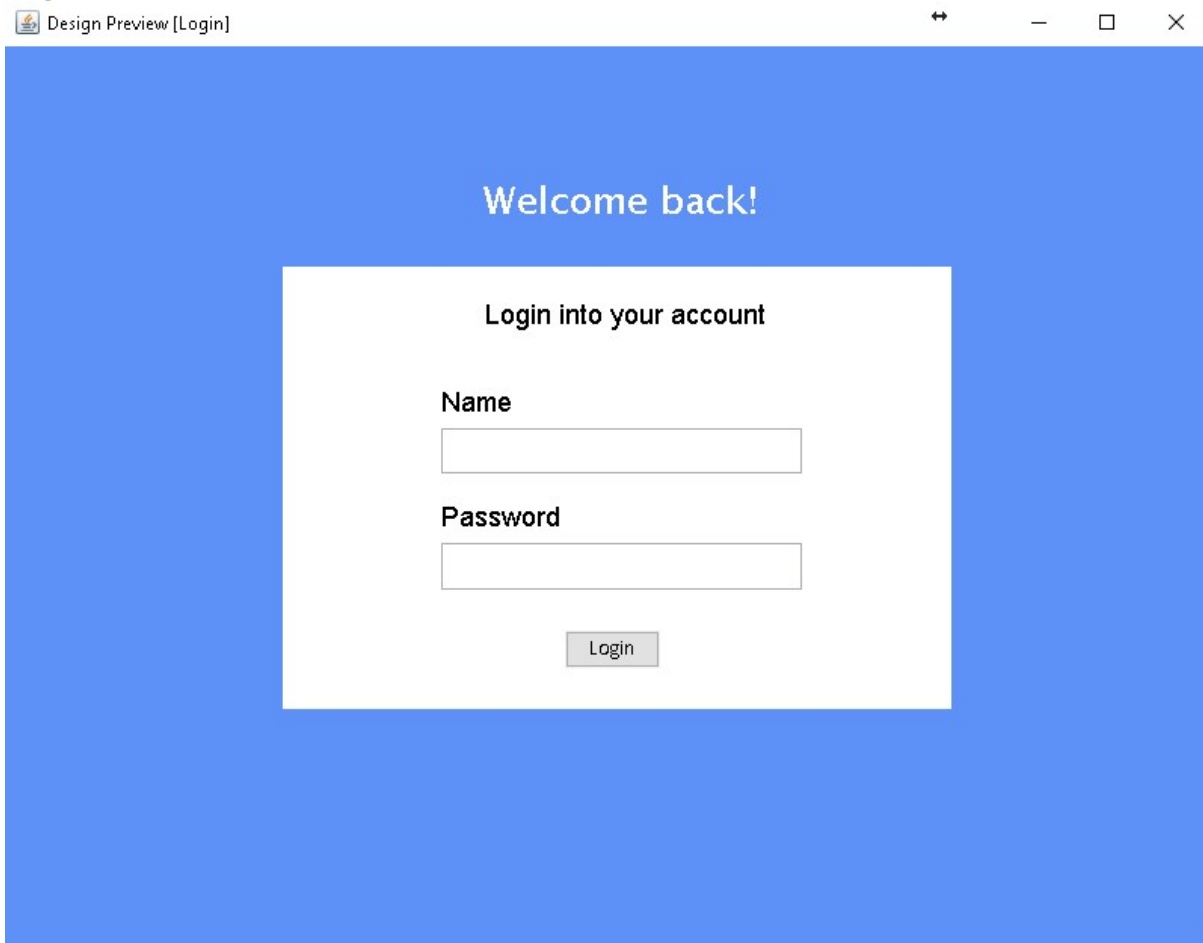
*This is to certify that the Project Work entitled **School Management System** is carried out by R.No.1631331.*

*He has completed his Project under my guidance.
His approach to work has been sincere and
satisfactory.*

*Mr. M. K. Gupta
HOD- Computer Dept.
BRJD Public School*

Login.java

Layout



Coding

```
private void logintbtnActionPerformed(java.awt.event.ActionEvent evt) {  
    String name = nametxt.getText();  
    String pswd = new String(pswdtxt.getPassword());  
    if(pswd.equals("1234")){  
        this.setVisible(false);  
        Welcome w = new Welcome();  
        w.setVisible(true);  
    }else{  
        JOptionPane.showMessageDialog(null,"Wrong Password");  
    }  
}
```

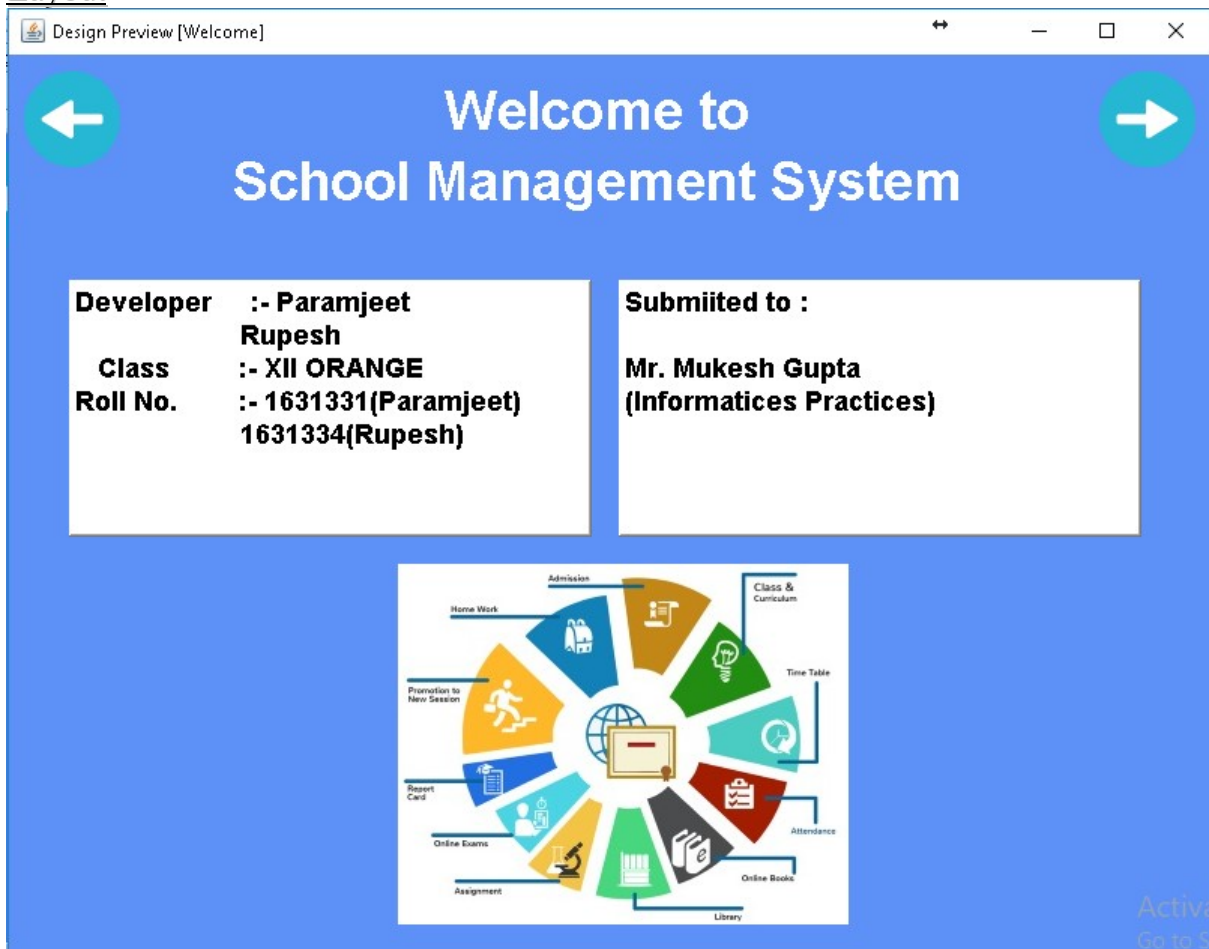
```

private void pswdtxtKeyPressed(java.awt.event.KeyEvent evt) {
    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
        String name = nametxt.getText();
        String pswd = new String(pswdtxt.getPassword());
        if(pswd.equals("1234")){
            this.setVisible(false);
            Welcome w = new Welcome();
            w.setVisible(true);
        }else{
            JOptionPane.showMessageDialog(null,"Wrong Password");
        }
    }
}

```

Welcome.java

Layout



Coding

```

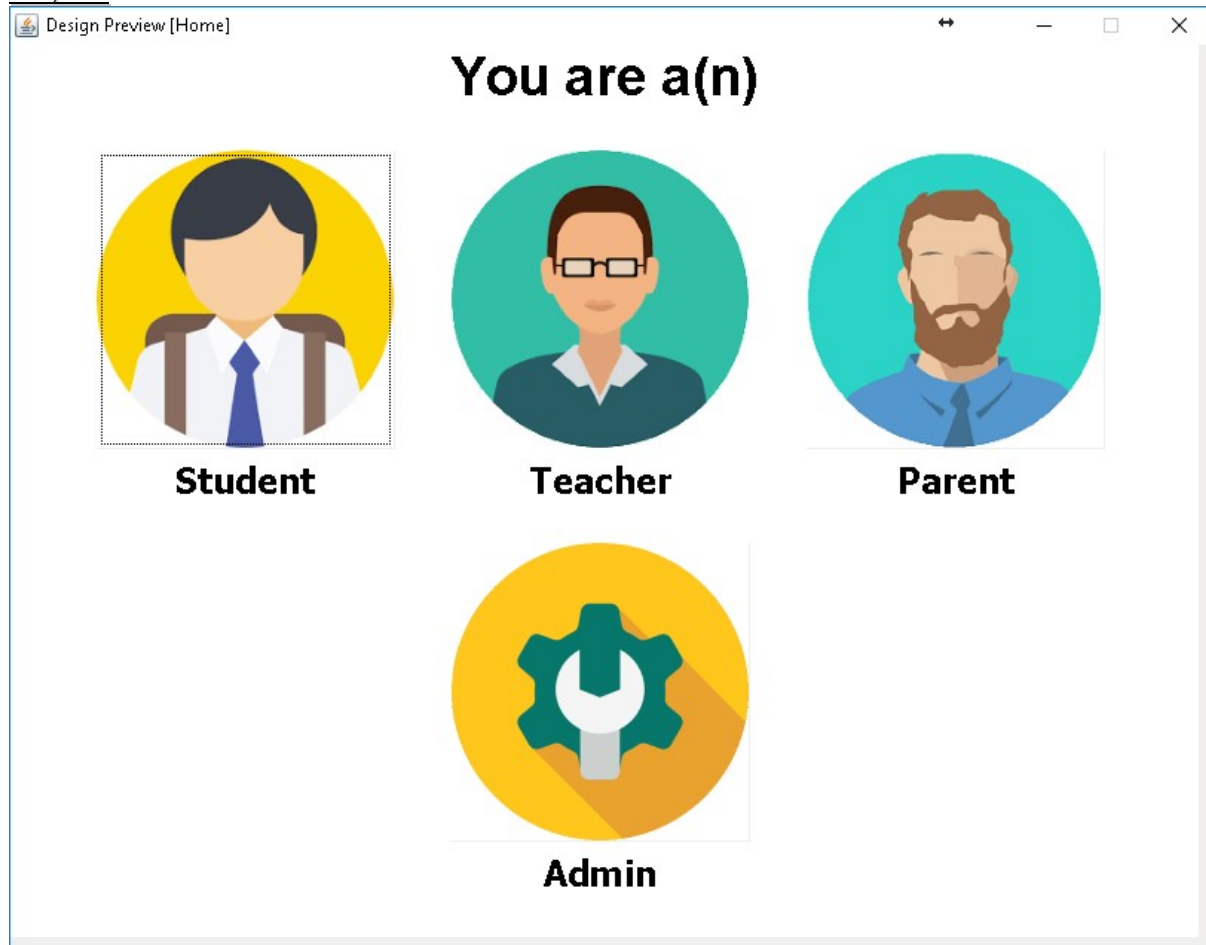
private void exitbtnMouseClicked(java.awt.event.MouseEvent evt) {
    System.exit(0);
}

```

```
private void nextbtnMouseClicked(java.awt.event.MouseEvent evt) {
    this.setVisible(false);
    new Home().setVisible(true);
}
```

Home.java

Layout



Coding

```
private void studentbtnActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    new Student_Dashboard().setVisible(true);
}
```

```
private void teacherbtnActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    StaffLogin tg = new StaffLogin();
    tg.opening = "teachdash";
    tg.setVisible(true);
}
```

```

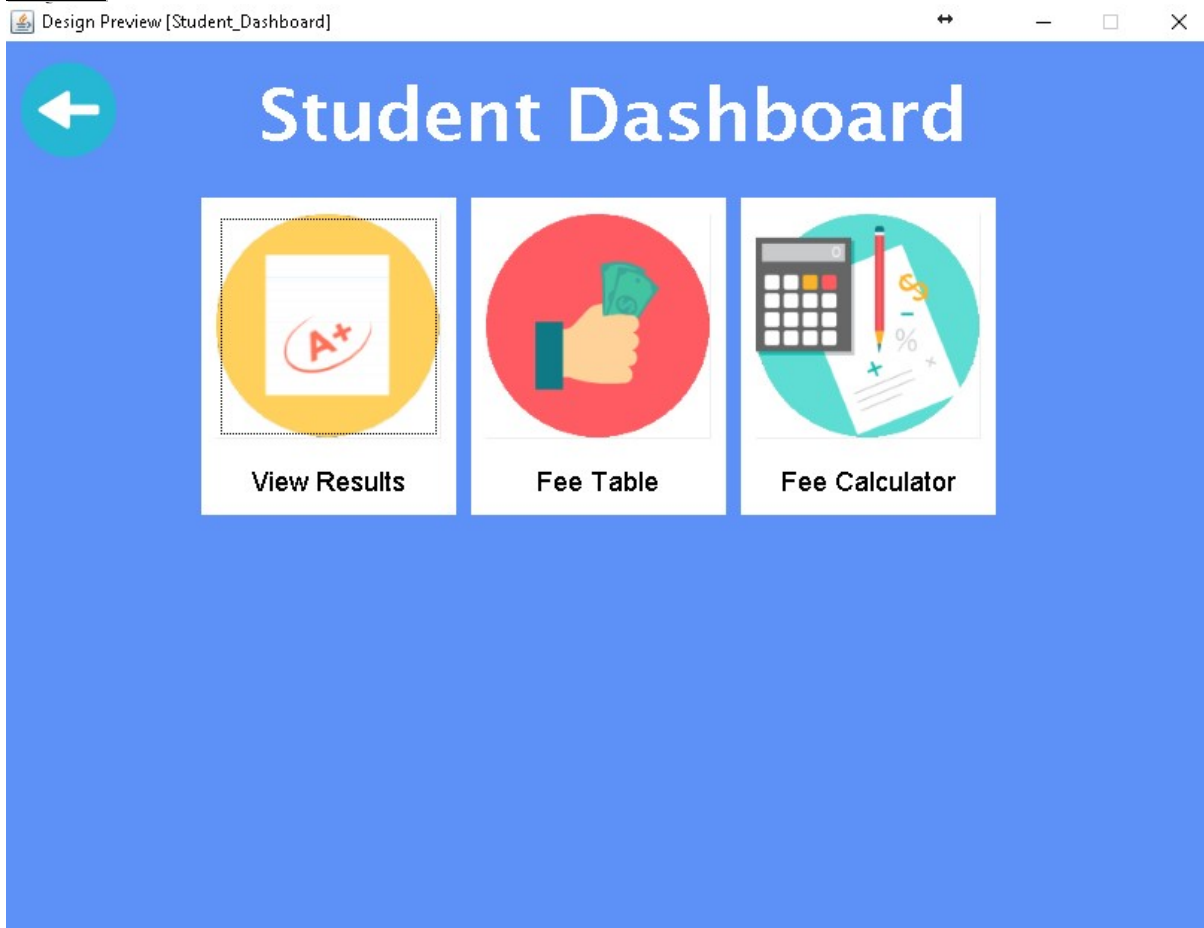
private void parentbtnActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    new Parent_Dashboard().setVisible(true);
}

private void adminbtnActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    StaffLogin tg = new StaffLogin();
    tg.opening = "admindash";
    tg.setVisible(true);
}

```

Student_Dashboard.java

Layout



Coding

```

private void resultsbtnActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    new View_Results().setVisible(true);
}

```

```

private void feecalcbtnActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    new Fee_Calculator().setVisible(true);
}

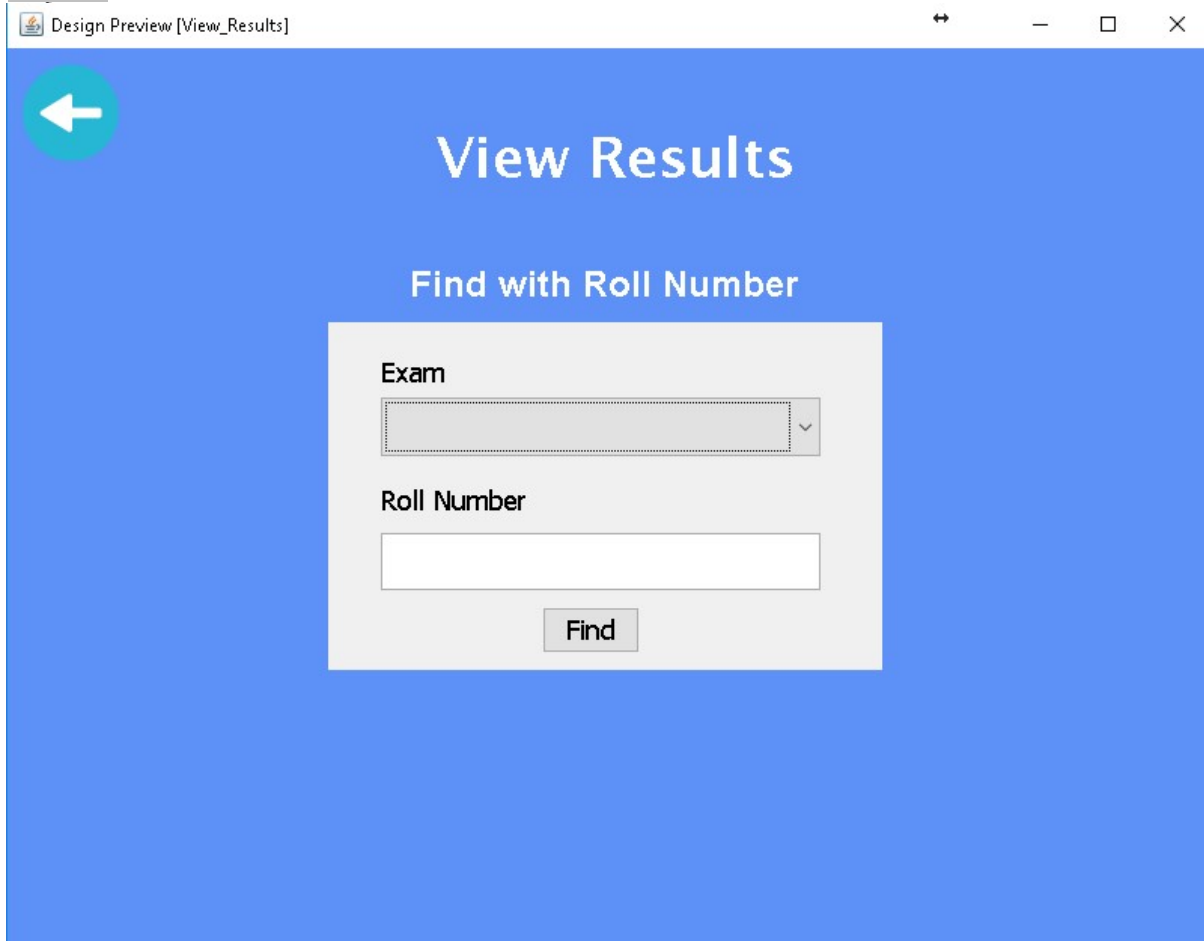
private void feestatusbtnActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    new Fee_Table().setVisible(true);
}

private void backbtnMouseClicked(java.awt.event.MouseEvent evt) {
    this.setVisible(false);
    new Home().setVisible(true);
}

```

View_Results.java

Layout



The image shows a design preview of a Java Swing window titled "View Results". The window has a blue background. In the top-left corner, there is a circular button with a white left-pointing arrow. The title "View Results" is centered at the top in a large, white, sans-serif font. Below the title, the text "Find with Roll Number" is centered in a smaller, white, sans-serif font. In the center of the window, there is a light gray rectangular panel. Inside this panel, the word "Exam" is followed by a dropdown menu. Below the dropdown menu, the text "Roll Number" is followed by a text input field. At the bottom of the panel, there is a button labeled "Find". The window's title bar includes standard OS controls (minimize, maximize, close) and a "Design Preview [View_Results]" label.

Coding

```
import java.sql.Connection;
```



```

import java.sql.DriverManager;
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.JComboBox;
import javax.swing.JOptionPane;
public static int parent = 0;

private void backbtnMouseClicked(java.awt.event.MouseEvent evt) {
    this.setVisible(false);
    if(parent==0){
        new Student_Dashboard().setVisible(true);
    }else if(parent==1){
        new Parent_Dashboard().setVisible(true);
    }
}

private void findbtnActionPerformed(java.awt.event.ActionEvent evt) {
    String exam = (String) examchk.getSelectedItem();
    String roll = rolltxt.getText();
    String ecode = exam.substring(exam.indexOf(':')+2);
    String ename = exam.substring(0,exam.indexOf(':')-1);
    MarkSheet ms = new MarkSheet();
    ms.ecode = ecode;
    ms.ename = ename;
    ms.eroll = roll;
    this.setVisible(false);
    ms.setVisible(true);
}

@Override
public void show() {
    super.show();
    try {
        Class.forName("java.sql.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/brjd", "root", "root");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM exams WHERE
Published=1;");
        if (!rs.isBeforeFirst() ) {
            JOptionPane.showMessageDialog(null,"No Results uploaded yet");

```

```


    }else{
        JComboBox exams = examchk;
        while(rs.next()){
            int examcode = rs.getInt("ExamCode");
            String examname = rs.getString("ExamName");
            exams.addItem(examname+" : "+examcode);
        }
    }
    rs.close();
    stmt.close();
    con.close();
} catch (ClassNotFoundException ex) {
    Logger.getLogger(Add_Staff.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {
    Logger.getLogger(Add_Staff.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

Marksheet.java

Layout

Design Preview [MarkSheet]



Mark Sheet

ExamName

Name

Father's Name

ExamRoll

Subject	Marks	Maximum Marks

Coding

```
import java.sql.Connection;
import java.sql.DriverManager;
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.JComboBox;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
```

```
public static String eroll = null;
public static String ecode = null;
public static String ename = null;
```

```
private void backbtn2MouseClicked(java.awt.event.MouseEvent evt) {
    this.setVisible(false);
    new View_Results().setVisible(true);
}
```

@Override

```
public void show() {
    super.show();
    try {
        Class.forName("java.sql.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/brjd", "root", "root");
        Statement stmt = con.createStatement();
        enametxt.setText(ename);
        ResultSet detail = stmt.executeQuery("SELECT
ExamRoll,Name,FathersName FROM Students WHERE ExamRoll="+eroll+";");
        if(!detail.isBeforeFirst()){
            JOptionPane.showMessageDialog(null,"Your details does not exist in
database please check you roll number");
        }else{
            detail.next();
            rolltxt.setText(eroll);
            nametxt.setText(detail.getString("Name"));
            fnametxt.setText(detail.getString("FathersName"));
            ResultSet rs = stmt.executeQuery("SELECT
Subject,Marks,MaximumMarks FROM results WHERE ExamCode="+ecode+
AND ExamRoll="+eroll+";");
```

```

if (!rs.isBeforeFirst() ) {
    JOptionPane.showMessageDialog(null,"No Results Found");
}else{
    DefaultTableModel model = (DefaultTableModel) examtbl.getModel();
    while(rs.next()){
        String subject = rs.getString("Subject");
        String marks = Integer.toString(rs.getInt("Marks"));
        String mm = Integer.toString(rs.getInt("MaximumMarks"));
        model.addRow(new Object[] {subject,marks,mm});
    }
}
rs.close();
}
stmt.close();
con.close();
} catch (ClassNotFoundException ex) {
    Logger.getLogger(Add_Staff.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {
    Logger.getLogger(Add_Staff.class.getName()).log(Level.SEVERE, null, ex);
}
}
}

```

Fee_Table.java

Layout

Design Preview [Fee_Table]

BRJD PUBLIC SCHOOL, BHORUGRAM DISTT. (CHURU)						
Particulars	LKG. To KG	Ist To III	IV to VIII	IX To X	XI Com - XII Com	XI Sci -XII Sci.
Registration Fee	1000	1000	1000	1000	1000	1000
Prospectus	200	200	200	200	200	200
Tuition Fee	9000	9000	18510	20930	32430	34850
Annual Fee	3000	5000	6050	6050	6050	6050
Computer Fee		800	3500	5000	7000	7000
Lab/Workshop Fee	-	-	3000	4500	4000	5000
Admission Fee	2000	4000	4000	4000	4000	5000
Caution Money(Refundable)	1000	1000	3000	3000	3000	3000
Total New Student (Day Sscholar)	16200	21000	39260	44680	57680	62100
Ist Instalment Old Student	4000	5000	11000	13000	17000	18000
Ist Instalment New Student	8200	11200	19200	21200	25200	27200
II Instalment (10 August)	4000	5000	11000	13000	17000	18000
III Instalment (10 November)	4000	4800	9060	10480	15480	16900
for Old Student (Day Scholar)	12000	14800	31060	36480	49480	52900
Hostel Fee	104500	104500	114400	125400	126500	127600
Pocket Money	6000	6000	6000	6000	7000	7000
Extra Caution money from Boarders					2000	2000
	110500	110500	120400	131400	135500	136600
For Old Student (Boarder)	122500	125300	151460	167880	182980	187500
For New Student (Boarder)	126700	131500	159660	176080	193180	198700

Coding

```
public static int parent = 0;
```

```
private void backbtnMouseClicked(java.awt.event.MouseEvent evt) {
    this.setVisible(false);
    if(parent==0){
        new Student_Dashboard().setVisible(true);
    }else if(parent==1){
        new Parent_Dashboard().setVisible(true);
    }
}
```

Fee_Calculator.java

Layout

The image shows a design preview of a Java Swing window titled "Design Preview [Fee_Calculator]". The window has a blue background and a white title bar. In the top-left corner, there is a circular button with a white left-pointing arrow. The main content area is a light gray rectangle containing the following elements:

- Class**: A dropdown menu with "Class 1st" selected.
- Stream**: A dropdown menu with "Science" selected.
- Eligible for following Scholarship**: A dropdown menu with "Ward of employee of BCT/Teacher of BRJD" selected.
- Radio Buttons**: Two radio buttons labeled "New Student" and "Old Student".
- Calculate**: A rectangular button.
- Annual Fee**: A text input field.
- Scholarship**: A text input field.
- Amount to pay**: A text input field.

Coding

```
public static int parent = 0;
```

```
private void backbtnMouseClicked(java.awt.event.MouseEvent evt) {  
    this.setVisible(false);  
    if(parent==0){  
        new Student_Dashboard().setVisible(true);  
    }else if(parent==1){  
        new Parent_Dashboard().setVisible(true);  
    }  
}
```

```
private void calculatebtnActionPerformed(java.awt.event.ActionEvent evt) {  
    int clas = classchk.getSelectedIndex()+1;  
    int stream = streamchk.getSelectedIndex();  
    int tutitionfee = 0;
```

```

int otherfee=0;
if(clas>=1 && clas<=3){
    tutionfee = 9000;
    if(newsturb.isSelected()){
        otherfee = 12000;
    }else if(oldsturb.isSelected()){
        otherfee = 6800;
    }
}
else if(clas>3 && clas<=8){
    tutionfee = 18510;
    if(newsturb.isSelected()){
        otherfee = 20750;
    }else if(oldsturb.isSelected()){
        otherfee = 12550;
    }
}
if(clas==9 || clas==10){
    tutionfee = 20930;
    if(newsturb.isSelected()){
        otherfee = 23750;
    }else if(oldsturb.isSelected()){
        otherfee = 15550;
    }
}
if(clas==11 || clas==12){
    if(stream==0){
        tutionfee = 34850;
        if(newsturb.isSelected()){
            otherfee = 27280;
        }else if(oldsturb.isSelected()){
            otherfee = 18080;
        }
    }else{
        tutionfee = 32430;
        if(newsturb.isSelected()){
            otherfee = 25250;
        }else if(oldsturb.isSelected()){
            otherfee = 17050;
        }
    }
}
int tutionfeedisc=0,otherfeedisc=0;
int ss = scholarshipchk.getSelectedIndex();

```

```

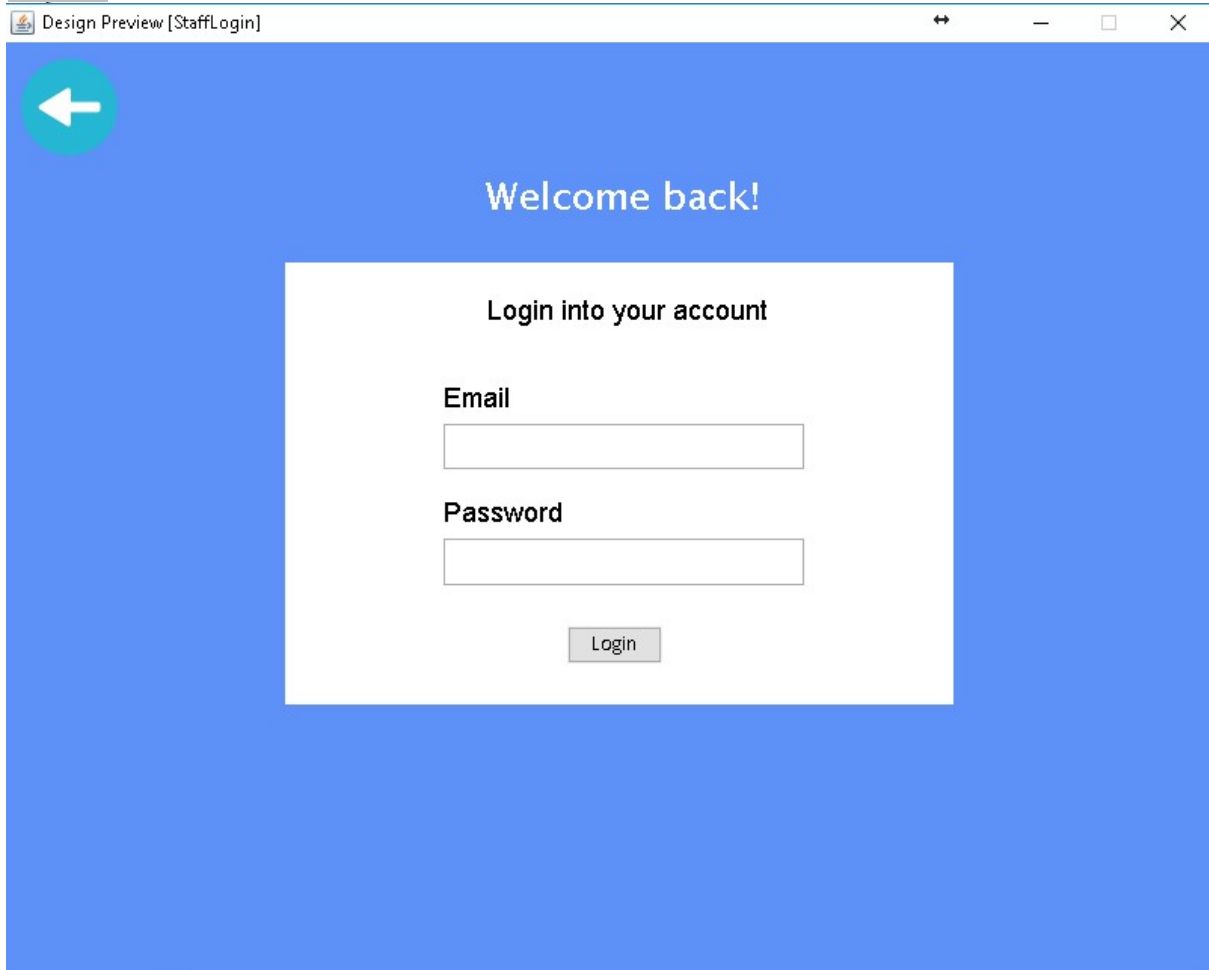
if(ss==0){
    tutionfeedisc =(int) (tutionfee*0.75);
    otherfeedisc =(int) (otherfee*0.25);
}else if(ss==1){
    tutionfeedisc =(int) (tutionfee*0.30);
}else if(ss==2){
    tutionfeedisc =(int) 10580;
}else if(ss==3){
    tutionfeedisc =(int) (tutionfee*0.20);
}
int fee = (int) tutionfee+otherfee;
int scholarship = tutionfeedisc+otherfeedisc;
int amt = fee-scholarship;
feetxt.setText(""+fee);
scholarshiptxt.setText(""+scholarship);
amounttxt.setText(""+amt);
}

private void classchkItemStateChanged(java.awt.event.ItemEvent evt) {
    int clas = classchk.getSelectedIndex()+1;
    if(clas==11 || clas==12){
        streamchk.setEnabled(true);
    }
}
}

```


Staff_Login.java

Layout



Coding

```
import java.awt.event.KeyEvent;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

public static String opening = null;

void login(){
    String email = emailtxt.getText();
    String pswd = pswdtxt.getText();
```

```

try {
    Class.forName("java.sql.Driver");
    Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/brjd","root","root");
    Statement stmt = con.createStatement();
    String query = "SELECT * FROM employee WHERE
email=\""+email+"\"";
    ResultSet rs = stmt.executeQuery(query);
    if(!rs.isBeforeFirst()){
        JOptionPane.showMessageDialog(null,"Sorry, your email is not registered
with us");
        rs.close();
        stmt.close();
        con.close();
    }else{
        rs.next();
        String uid = rs.getString("EmpID");
        String uemail = rs.getString("Email");
        String upswd = rs.getString("Password");
        String upost = rs.getString("Post");
        int uls = rs.getInt("ULS")+1;
        if(pswd.equals(P.decrypt(upswd))){
            stmt.execute("UPDATE employee SET ULS="+uls+" WHERE
Email=\""+uemail+"\"");
            this.setVisible(false);
            if(uls==1){
                First_Login fl = new First_Login();
                fl.userID = uid;
                fl.opening = opening;
                fl.setVisible(true);
                rs.close();
                stmt.close();
                con.close();
            }else
if((upost.equals("Teacher")||upost.equals("H.O.D.")||upost.equals("Developer")) &&
opening.equals("teachdash")){
                Teacher_Dashboard td = new Teacher_Dashboard();
                td.userID = uid;
                td.setVisible(true);
                rs.close();
                stmt.close();
                con.close();
            }else if((upost.equals("Admin")||upost.equals("Developer")) &&
opening.equals("admindash")){

```

```

        Admin_Dashboard ad = new Admin_Dashboard();
        ad.userID = uid;
        ad.setVisible(true);
        rs.close();
        stmt.close();
        con.close();
    }
    }else{
        rs.close();
        stmt.close();
        con.close();
        JOptionPane.showMessageDialog(null,"Wrong Password");
    }

}
} catch (ClassNotFoundException ex) {
    Logger.getLogger(StaffLogin.class.getName()).log(Level.SEVERE, null, ex);
} catch (SQLException ex) {
    Logger.getLogger(StaffLogin.class.getName()).log(Level.SEVERE, null, ex);
}
}
private void logintbtnActionPerformed(java.awt.event.ActionEvent evt) {
    login();
}

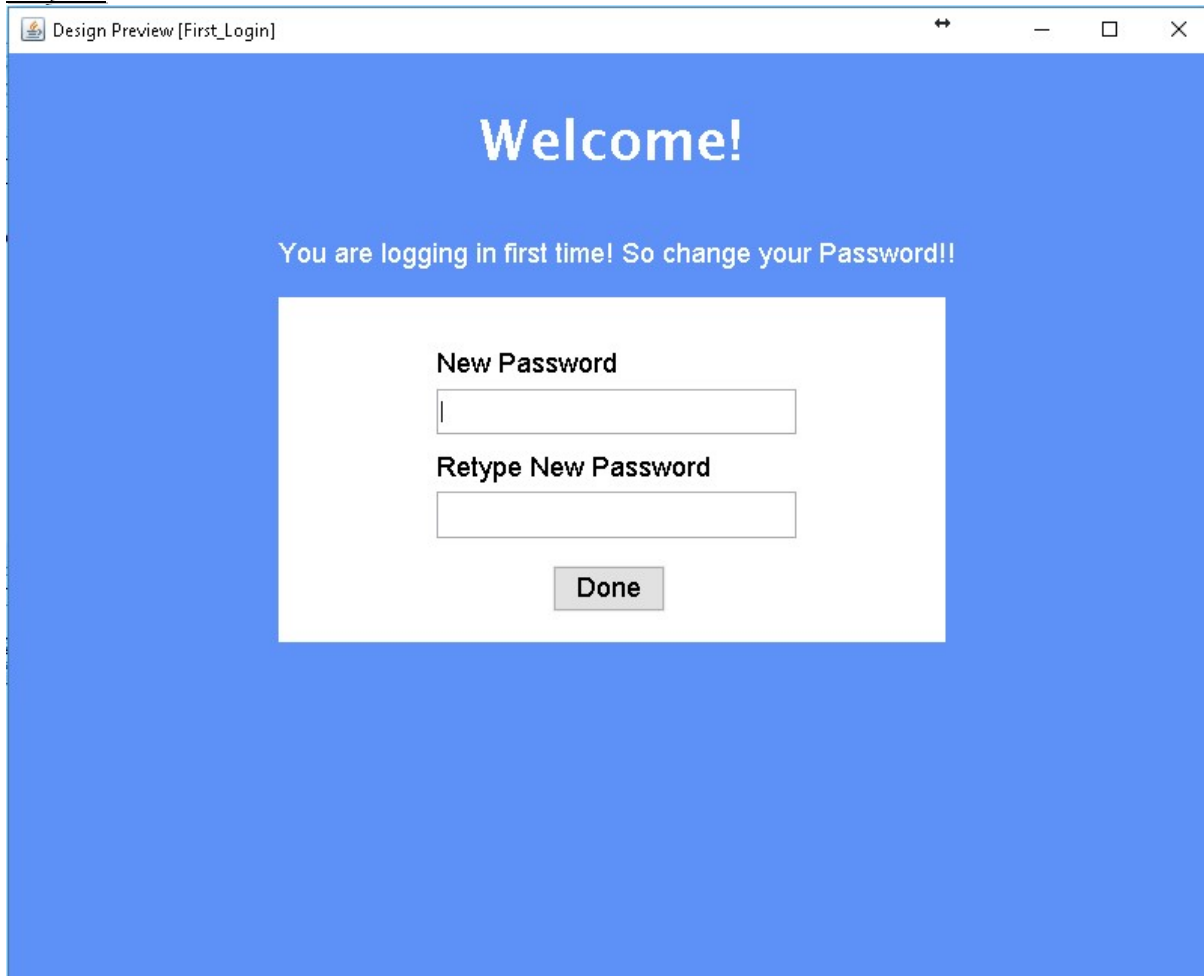
private void backbtnMouseClicked(java.awt.event.MouseEvent evt) {
    this.setVisible(false);
    new Home().setVisible(true);
}

private void pswdtxtKeyPressed(java.awt.event.KeyEvent evt) {
    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
        login();
    }
}
}

```

First_Login.java

Layout



Coding

```
import java.awt.event.KeyEvent;  
import java.sql.*;  
import java.sql.ResultSet;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.swing.JOptionPane;  
  
public static String userID = null;  
public static String opening = null;
```

```

void login(){
    String pswd = new String(pswd1txt.getPassword());
    String pswd2 = new String(pswd2txt.getPassword());
    if(!pswd.equals(pswd2)){
        JOptionPane.showMessageDialog(null,"Passwords do not match");
        return;
    }
    try {
        Class.forName("java.sql.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/brjd","root","root");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM employee WHERE
EmpID="+userID+";");
        rs.next();
        String email = rs.getString("Email");
        String query = "UPDATE employee SET Password='"+P.encrypt(pswd)+"'
WHERE email='"+email+"'";
        stmt.execute(query);
        this.setVisible(false);
        Teacher_Dashboard td = new Teacher_Dashboard();
        td.userID = userID;
        td.setVisible(true);
        rs.close();
        stmt.close();
        con.close();
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(StaffLogin.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(StaffLogin.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void donebtnActionPerformed(java.awt.event.ActionEvent evt) {
    login();
}

private void pswd2txtKeyPressed(java.awt.event.KeyEvent evt) {
    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
        login();
    }
}

private void pswd1txtKeyPressed(java.awt.event.KeyEvent evt) {
    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {

```

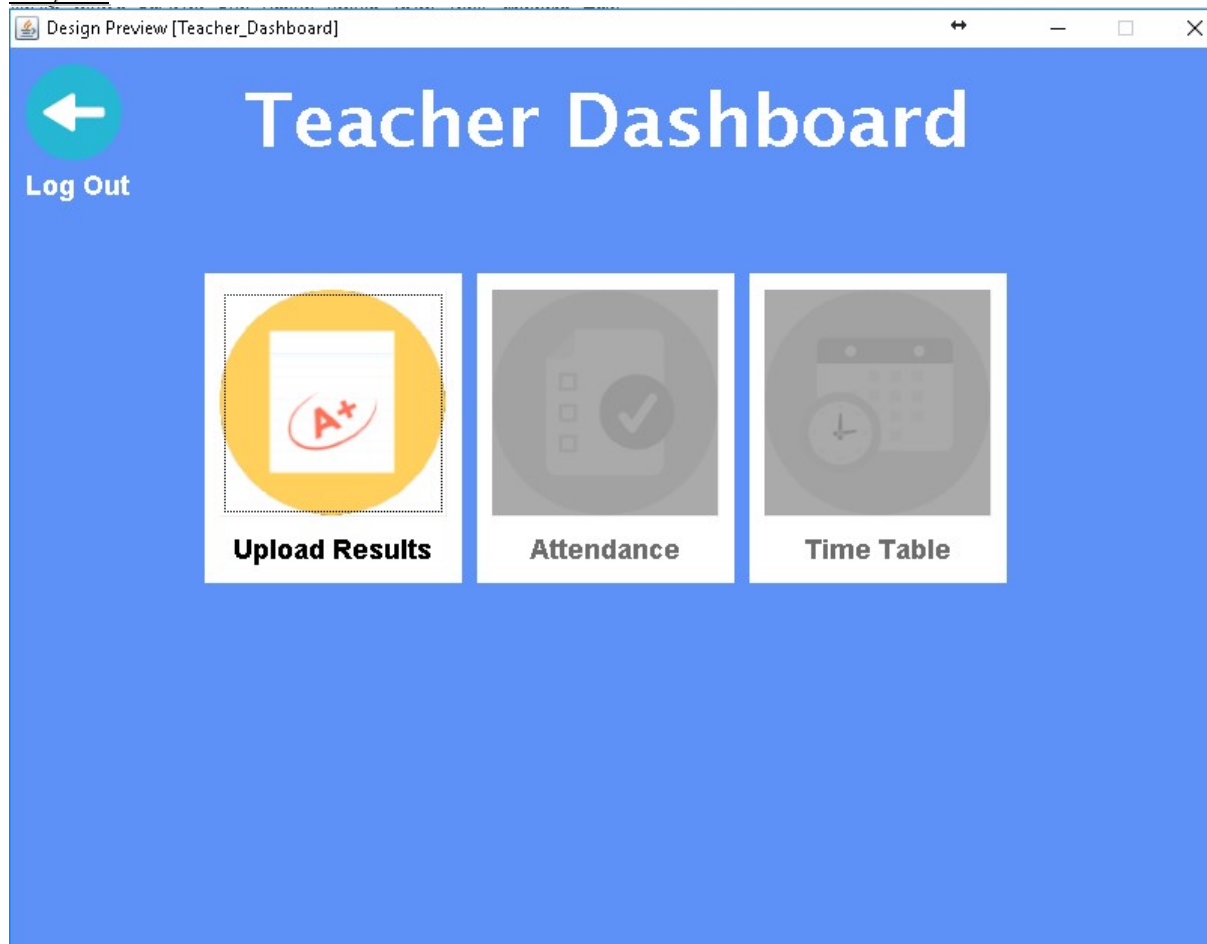
```

        pswd2txt.requestFocus();
    }
}

```

Teacher_Dashboard.java

Layout



Coding

```

public static String userID=null;

private void uploadresbtnActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    Upload_Results ur = new Upload_Results();
    ur.userID = userID;
    ur.setVisible(true);
}

```

Upload_Results.java

Layout

The image shows a design preview of a Java Swing window titled "Design Preview [Upload_Results]". The window has a blue background. In the top-left corner, there is a circular button with a white left-pointing arrow. The title "Upload Results" is displayed in large white font in the top-right area. Below this, there is a light gray panel containing a form. The form has four dropdown menus labeled "Class", "Section", "Subject", and "Exam". The "Class" dropdown is currently set to "Class 1st", "Section" to "Orange", and "Subject" to "Physics". Below these dropdowns is a table with three columns: "ExamRoll", "Name", and "Marks". The table is currently empty. At the bottom of the form, there is a label "Maximum Marks" followed by a text input field, and an "Upload" button.

Coding

```
import java.sql.Connection;  
import java.sql.DriverManager;  
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.JComboBox;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

public static String userID = null;

void createFields(){
    int clas = P.getClassCode(classchk,sectionchk);
    try {
        Class.forName("java.sql.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/brjd", "root", "root");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT ExamRoll,Name FROM
students WHERE Class="+clas+";");
        if (!rs.isBeforeFirst() ) {
            DefaultTableModel model = (DefaultTableModel) restbl.getModel();
            int r = model.getRowCount();
            for(int i=0;i<=model.getRowCount()-1;i++){
                model.removeRow(i);
            }
            JOptionPane.showMessageDialog(null,"Their is no student in this class as
per database");
        }else{
            DefaultTableModel model = (DefaultTableModel) restbl.getModel();
            int r = model.getRowCount();
            for(int i=0;i<=model.getRowCount()-1;i++){
                model.removeRow(i);
            }
            while(rs.next()){
                String examroll = Integer.toString(rs.getInt("ExamRoll"));
                String name = rs.getString("Name");
                model.addRow(new Object[]{examroll,name,""});
            }
        }
        rs.close();
        stmt.close();
        con.close();
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Add_Staff.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(Add_Staff.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```



```

    }
}
private void backbtnMouseClicked(java.awt.event.MouseEvent evt) {
    this.setVisible(false);
    Teacher_Dashboard td = new Teacher_Dashboard();
    td.userID = userID;
    td.setVisible(true);
}

private void classchkItemStateChanged(java.awt.event.ItemEvent evt) {
    createFields();
}

private void sectionchkItemStateChanged(java.awt.event.ItemEvent evt) {
    createFields();
}

private void uploadbtnActionPerformed(java.awt.event.ActionEvent evt) {
    int mm = Integer.parseInt(mmtxt.getText());
    String ecode="";
    String subject = (String) subjectchk.getSelectedItemAt();
    String exam = (String) examchk.getSelectedItemAt();
    ecode = exam.substring(exam.indexOf('.')+2);
    try {
        Class.forName("java.sql.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/brjd", "root", "root");
        Statement stmt = con.createStatement();
        DefaultTableModel model = (DefaultTableModel) restbl.getModel();
        int lastrow = model.getRowCount()-1;
        for(int i=0;i<=lastrow;i++){
            String roll = (String) model.getValueAt(i, 0);
            String marks = (String) model.getValueAt(i, 2);
            String query = "INSERT INTO results
VALUES("+roll+", "+ecode+", \""+subject+"\", "+marks+", "+mm+");";
            stmt.execute(query);
        }
        JOptionPane.showMessageDialog(null, "Result Uploaded Successfully!!!");
        int r = model.getRowCount();
        for(int i=0;i<=model.getRowCount()-1;i++){
            model.removeRow(i);
        }
        stmt.close();
        con.close();
    }
}

```

```

    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Add_Staff.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(Add_Staff.class.getName()).log(Level.SEVERE, null, ex);
    }
}

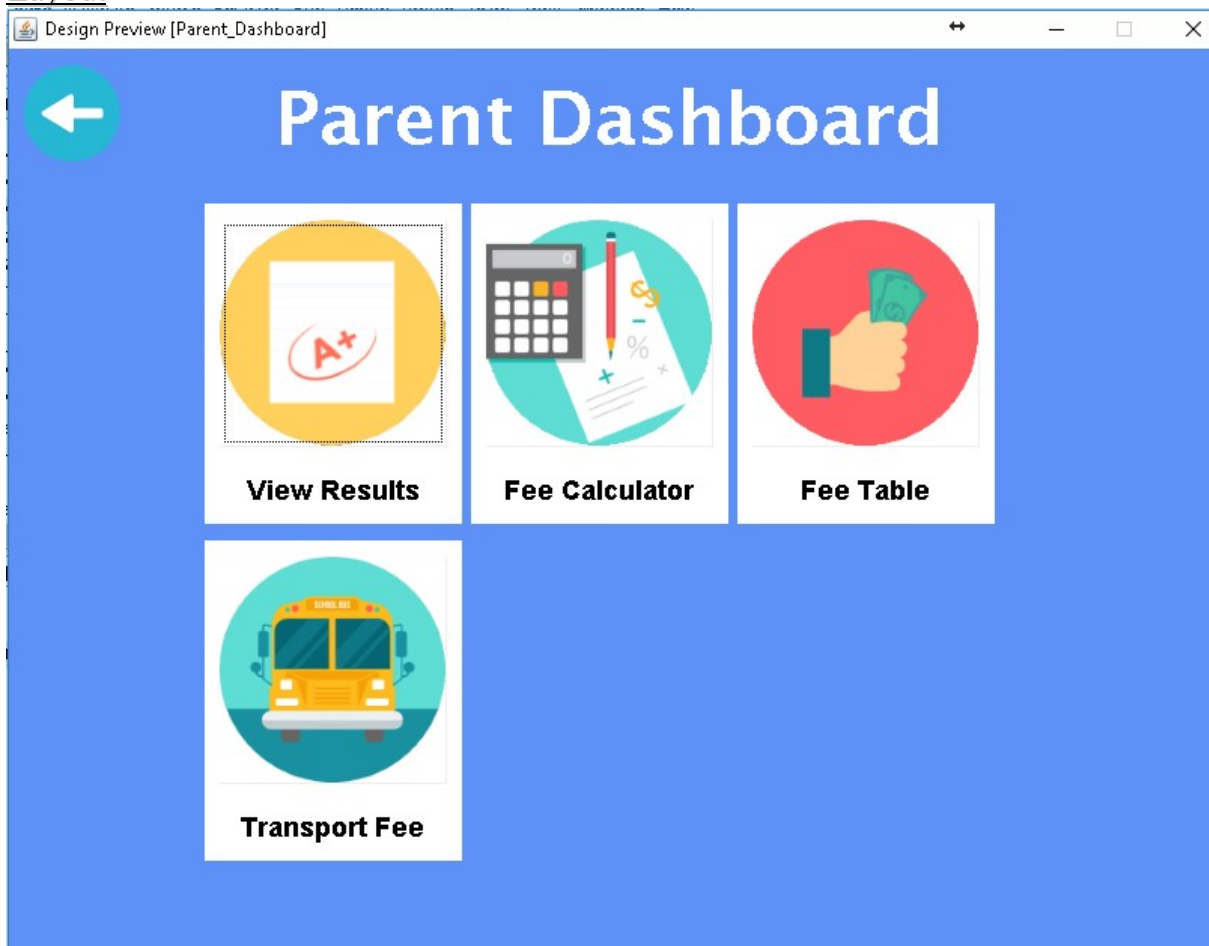
@Override
public void show() {
    super.show();
    try {
        Class.forName("java.sql.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/brjd", "root", "root");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM exams WHERE
Published=0;");
        if (!rs.isBeforeFirst() ) {
            JOptionPane.showMessageDialog(null,"No Results uploaded yet");
        }else{
            JComboBox exams = examchk;
            while(rs.next()){
                int examcode = rs.getInt("ExamCode");
                String examname = rs.getString("ExamName");
                exams.addItem(examname+" : "+examcode);
            }
        }
        ResultSet staffrs = stmt.executeQuery("SELECT * FROM employee
WHERE EmpID="+userID+";");
        staffrs.next();
        String post = staffrs.getString("Post");
        String dept = staffrs.getString("Department");
        subjectchk.setSelectedItem(dept);
        if(post.equals("Developer") || post.equals("admin"))
            subjectchk.setEnabled(true);
        rs.close();
        stmt.close();
        con.close();
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Add_Staff.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(Add_Staff.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

}

Parent_Dashboard.java

Layout



Coding

```
private void vresbtnActionPerformed(java.awt.event.ActionEvent evt) {  
    this.setVisible(false);  
    View_Results vr = new View_Results();  
    vr.parent = 1;  
    vr.setVisible(true);  
}
```

```

private void feecalcbtnActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    Fee_Calculator fc = new Fee_Calculator();
    fc.parent = 1;
    fc.setVisible(true);
}

private void feetablebtnActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    Fee_Table ft = new Fee_Table();
    ft.parent = 1;
    ft.setVisible(true);
}

private void backbtnMouseClicked(java.awt.event.MouseEvent evt) {
    this.setVisible(false);
    new Home().setVisible(true);
}

private void feetablebtn1ActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    Transport_Fee tf = new Transport_Fee();
    tf.parent = 1;
    tf.setVisible(true);
}

```

Transport_Fee.java

Layout



Transport Fee

TRANSPORTATION FEE 2017-18					
S.No.	Pickup Point	2014-15	2015-16	2016-17	2017-18
1	AJEETPURA	860	950	1045	1150
2	BADAWA	860	950	1045	1150
3	BALSAMAND	640	700	770	850
4	BHAGELA	570	630	690	760
5	BHANGAD	500	550	605	670
6	BHATHOD	610	670	735	810
7	BHIMSANA	720	790	870	960
8	BIDHWAN	640	700	770	850
9	BUDHSHAILI	720	790	870	960
10	CHENPURA BADA	570	630	820	900
11	CHENPURA CHOTA	570	630	850	940
12	CHUBKIATAL	720	790	800	880
13	DABARI	720	790	870	960
14	DEOSAR	720	790	870	960
15	DEVA	720	790	870	960
16	DHANI BHANKARAN	540	600	660	730
17	DHANI SOLAWALI	570	630	690	760
18	DHANOTI	430	470	515	570
19	DIGARALA	640	700	770	850
20	DUMKI	360	390	430	470
21	GADADDA	720	790	870	960

Page 1

Page 2

Coding

```
public static int parent = 0;
```

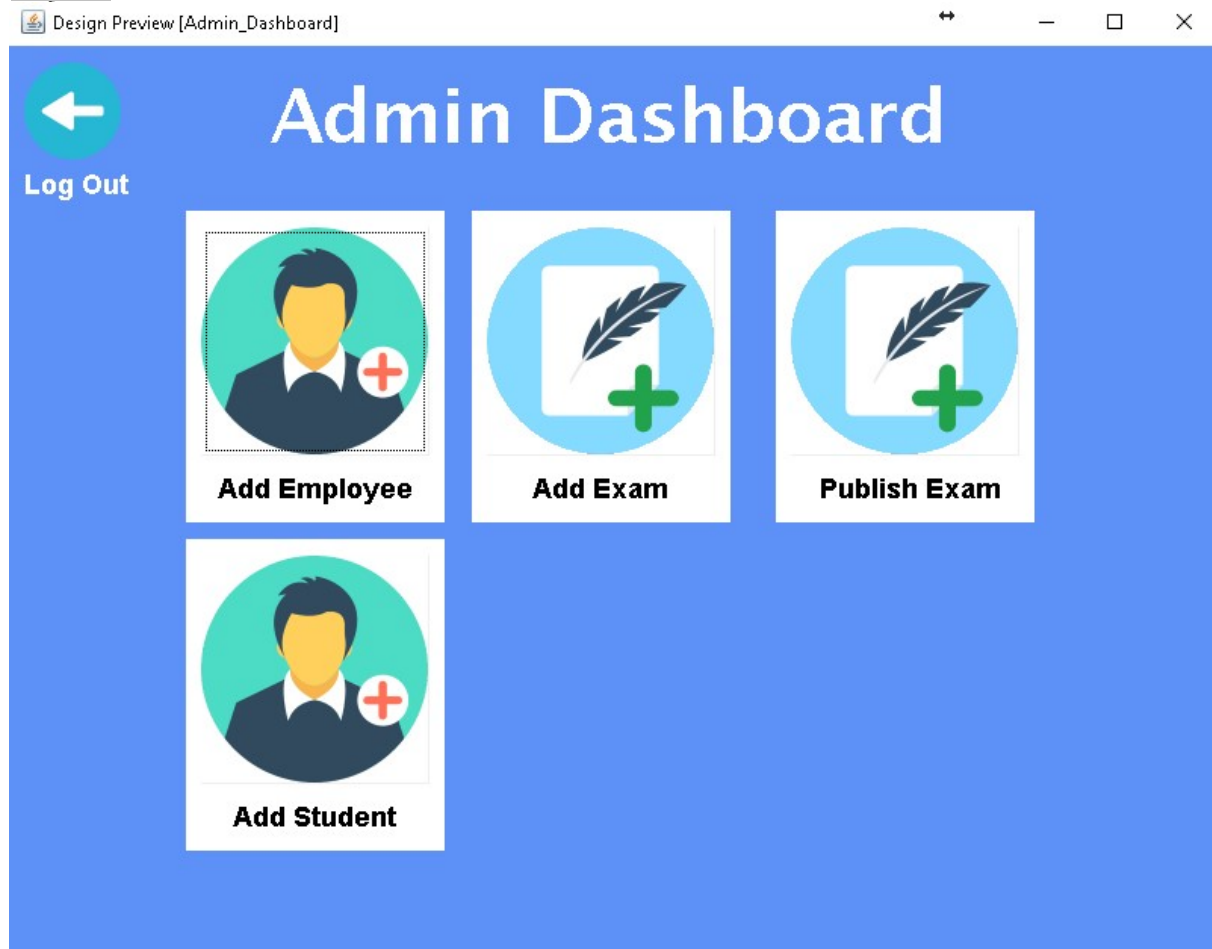
```
private void backbtn2MouseClicked(java.awt.event.MouseEvent evt) {
    this.setVisible(false);
    if(parent==0){
        new Student_Dashboard().setVisible(true);
    }else if(parent==1){
        new Parent_Dashboard().setVisible(true);
    }
}
```

```
private void p2btnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
```

```
private void p1btnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
```

Admin_Dashboard.java

Layout



Coding

```
public static String userID = null;

private void backbtnMouseClicked(java.awt.event.MouseEvent evt) {
    this.setVisible(false);
    new Home().setVisible(true);
}

private void addempbtnActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    Add_Staff as = new Add_Staff();
    as.userID = userID;
    as.setVisible(true);
}

private void addexambtnActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    Add_Exams ae = new Add_Exams();
```

```

        ae.userID = userID;
        ae.setVisible(true);
    }

    private void publishexambtnActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        Publish_Exam pe = new Publish_Exam();
        pe.userID = userID;
        pe.setVisible(true);
    }

    private void addstudentbtnActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        Add_Student as = new Add_Student();
        as.userID = userID;
        as.setVisible(true);
    }

```

Add_Staff.java

Layout

Design Preview [Add_Staff]

Add Staff

←

Employee No.		Name *	
<input type="text"/>		<input type="text"/>	
Email *		Mobile No. *	
<input type="text"/> @brjdschool.org ▾		<input type="text"/>	
Salary *	Gender *	Post *	Department *
<input type="text"/>	<input type="radio"/> M <input type="radio"/> F	Teacher ▾	Physics ▾
Aadhaar No. *		Date of Joining *	
<input type="text"/>	<input type="text"/>	<input type="text"/> / <input type="text"/> / <input type="text"/>	
Address		State	Pin Code
<input type="text"/>		<input type="text"/>	<input type="text"/>

Coding

```
import java.awt.event.KeyEvent;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
```

```
public static String userID = null;
boolean dojback = false, anback=false;
```

```
private void addbtnActionPerformed(java.awt.event.ActionEvent evt) {
    String name = nametxt.getText();
    String email = emailtxt.getText();
    String edomain = emailchk.getSelectedItem().toString();
    String mobile = mobiletxt.getText();
    String salary = salarytxt.getText();
    String gender = null;
    if(mrb.isSelected()) gender="M";
    else if(frb.isSelected()) gender="F";
```



```

String post = postchk.getSelectedItemAt().toString();
String department = deptchk.getSelectedItemAt().toString();
String empno = null;
String aadhaarno = anb1.getText()+"-"+anb2.getText()+"-"+anb3.getText();
String dateofjoin = yeardoj.getText()+"-"+monthdoj.getText()+"-
"+daydoj.getText();
String address = addresstxt.getText();
String state = statetxt.getText();
String pincode = pincodetxt.getText();
if(name.isEmpty() || email.isEmpty() || mobile.isEmpty() || salary.isEmpty() ||
aadhaarno.isEmpty() || dateofjoin.isEmpty()){
    JOptionPane.showMessageDialog(null,"Please fill the required * fields");
    return;
}
try {
    Class.forName("java.sql.Driver");
    Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/brjd", "root", "root");
    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery("SELECT EmpID FROM employee;");
    if (!rs.isBeforeFirst() ) {
        empno = "101";
    }else{
        rs.last();
        empno = ""+(rs.getInt("EmpID")+1);
    }
    String password = P.encrypt(name+"."+empno);
    String addemployee = "INSERT INTO employee
VALUES("+empno+",\""+name+"\", \""+email+edomain+"\", \""+mobile+"\", \""+salary+"\",
\""+gender+"\", \""+post
+"\", \""+department+"\", \""+aadhaarno+"\", \""+dateofjoin+"\", \""+address+"\", \""+st
ate+"\", \""+pincode+"\", \""+password+"\", 0);";
    boolean done = stmt.execute(addemployee);
    JOptionPane.showMessageDialog(null,"<HTML>Employee Successfull
Added!!"+"<BR>"+
        "EmpID : "+empno+"<BR>"+
        "Name : "+name+"<BR>"+
        "Email : "+email+edomain+"<BR>"+
        "Password : "+name+"."+empno+"<BR>"+
        "Post : "+post+"<BR>"+
        "Department : "+department
    );
    rs.close();

```

```

        stmt.close();
        con.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null,"An Error
Occured!!\n"+e.getLocalizedMessage());
    }
}

private void daydojKeyReleased(java.awt.event.KeyEvent evt) {
    int len = daydoj.getText().length();
    if(len==2 && !dojback){
        dojback = false;
        monthdoj.requestFocus();
    }
    if(len<2){
        dojback = false;
    }
}

private void monthdojKeyReleased(java.awt.event.KeyEvent evt) {
    int len = monthdoj.getText().length();
    if(len==2 && !dojback){
        dojback = false;
        yeardoj.requestFocus();
    }else if (len==0 && evt.getKeyCode() == KeyEvent.VK_BACK_SPACE) {
        daydoj.requestFocus();
        dojback = true;
    }
    if(len<2){
        dojback = false;
    }
}

private void yeardojKeyReleased(java.awt.event.KeyEvent evt) {
    int len = yeardoj.getText().length();
    if (len==0 && evt.getKeyCode() == KeyEvent.VK_BACK_SPACE) {
        monthdoj.requestFocus();
        dojback = true;
    }
}

private void anb1KeyReleased(java.awt.event.KeyEvent evt) {
    int len = anb1.getText().length();

```

```

        if(len==4 && !anback){
            anback = false;
            anb2.requestFocus();
        }
        if(len<4){
            anback = false;
        }
    }

private void anb2KeyReleased(java.awt.event.KeyEvent evt) {
    int len = anb2.getText().length();
    if(len==4 && !anback){
        anback = false;
        anb3.requestFocus();
    }else if (len==0 && evt.getKeyCode() == KeyEvent.VK_BACK_SPACE) {
        anb1.requestFocus();
        anback = true;
    }
    if(len<4){
        anback = false;
    }
}

private void anb3KeyReleased(java.awt.event.KeyEvent evt) {
    int len = anb3.getText().length();
    if (len==0 && evt.getKeyCode() == KeyEvent.VK_BACK_SPACE) {
        anb2.requestFocus();
        anback = true;
    }
}

private void resetbtnActionPerformed(java.awt.event.ActionEvent evt) {
    Add_Staff p = this;
    Add_Staff as = new Add_Staff();
    as.setVisible(true);
    p.dispose();
}

private void backbtnMouseClicked(java.awt.event.MouseEvent evt) {
    this.setVisible(false);
    Admin_Dashboard ad = new Admin_Dashboard();
    ad.userID = userID;
    ad.setVisible(true);
}

```

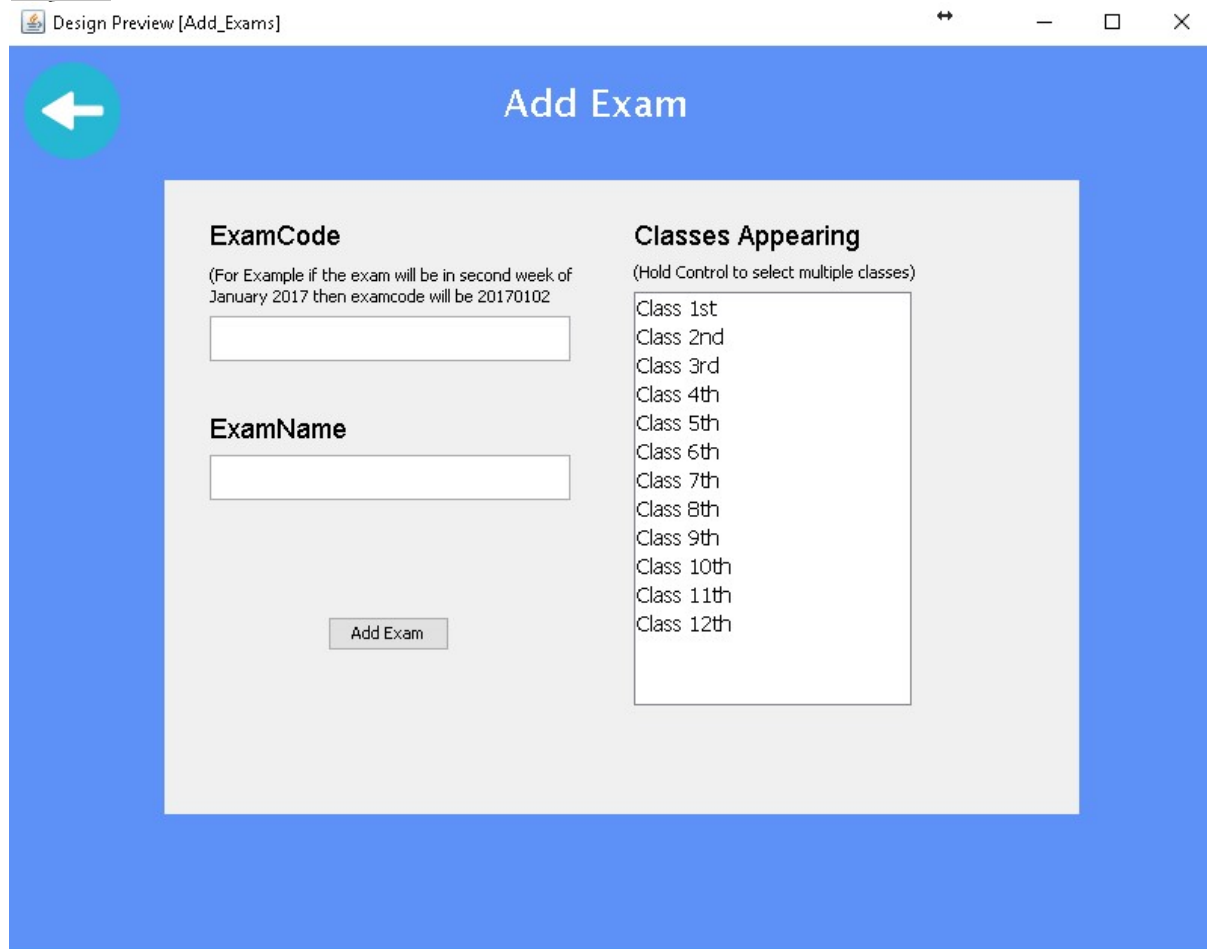
```

@Override
public void show() {
    super.show();
    String empno=null;
    try {
        Class.forName("java.sql.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/brjd", "root", "root");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT EmpID FROM employee;");
        if (!rs.isBeforeFirst() ) {
            empno = "101";
        }else{
            rs.last();
            empno = ""+(rs.getInt("EmpID")+1);
        }
        empnotxt.setText(empno);
        rs.close();
        stmt.close();
        con.close();
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Add_Staff.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(Add_Staff.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

```

Add_Exams.java

Layout



Coding

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

```
public static String userID = null;
```

```
private void backbtn2MouseClicked(java.awt.event.MouseEvent evt) {
    this.setVisible(false);
    Admin_Dashboard ad = new Admin_Dashboard();
    ad.userID = userID;
    ad.setVisible(true);
}
```

```

    }

    private void addexambtnActionPerformed(java.awt.event.ActionEvent evt) {
        String ecode = ecodetxt.getText();
        String ename = enametxt.getText();
        String ca = "";
        int[] sc = calst.getSelectedIndices();
        for(int i=0;i<=(sc.length-1);i++){
            if(!ca.isEmpty()){
                ca = ca + "," + (sc[i]+1);
            }else{
                ca = ca + "" + (sc[i]+1);
            }
        }
        try {
            Class.forName("java.sql.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/brjd", "root", "root");
            Statement stmt = con.createStatement();
            String query = "INSERT INTO exams
VALUES("+ecode+",\""+ename+"\",0,\""+ca+"\"),";
            stmt.execute(query);
            JOptionPane.showMessageDialog(null,"Exam Successfully Added!!!");
            stmt.close();
            con.close();
            ecodetxt.setText("");
            enametxt.setText("");
            calst.clearSelection();
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(Add_Exams.class.getName()).log(Level.SEVERE, null,
ex);
        } catch (SQLException ex) {
            Logger.getLogger(Add_Exams.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }
}

```

P.java

Coding

```
import javax.swing.JComboBox;
```

```
public class P {  
    public static String encrypt(String Password){  
        Password =  
Password.replace("a","^").replace("b","~").replace("c","/").replace("d","a").replace("e","b")  
  
.replace("f","c").replace("g","d").replace("h","e").replace("i","f").replace("j","g").replace("k","h")  
  
.replace("l","i").replace("m","j").replace("n","k").replace("o","l").replace("p","m").replace("q","n")  
  
.replace("r","o").replace("s","p").replace("t","q").replace("u","r").replace("v","s").replace("w","t")  
  
.replace("x","u").replace("y","v").replace("z","w").replace("^","x").replace("~","y").replace("/","z")  
  
.replace("A","^").replace("B","~").replace("C","/").replace("D","A").replace("E","B").replace("F","C")  
  
.replace("G","D").replace("H","E").replace("I","F").replace("J","G").replace("K","H").replace("L","I")  
  
.replace("M","J").replace("N","K").replace("O","L").replace("P","M").replace("Q","N").replace("R","O")  
  
.replace("S","P").replace("T","Q").replace("U","R").replace("V","S").replace("W","T").replace("X","U")  
  
.replace("Y","V").replace("Z","W").replace("^","X").replace("~","Y").replace("/","Z").replace("1","^")  
  
.replace("2","~").replace("3","/").replace("4","1").replace("5","2").replace("6","3").replace("7","4")  
    }  
}
```

```

.replace("8","5").replace("9","6").replace("0","7").replace("^","8").replace("~","9").r
eplace("/","0");
    return Password;
}
public static String decrypt(String Password){
    Password =
Password.replace("z","^").replace("y","~").replace("x","/").replace("w","z").replace(
"v","y")

.replace("u","x").replace("t","w").replace("s","v").replace("r","u").replace("q","t").re
place("p","s")

.replace("o","r").replace("n","q").replace("m","p").replace("l","o").replace("k","n").r
eplace("j","m")

.replace("i","l").replace("h","k").replace("g","j").replace("f","i").replace("e","h").repl
ace("d","g")

.replace("c","f").replace("b","e").replace("a","d").replace("^","c").replace("~","b").re
place("/","a")

.replace("Z","^").replace("Y","~").replace("X","/").replace("W","Z").replace("V","Y
").replace("U","X")

.replace("T","W").replace("S","V").replace("R","U").replace("Q","T").replace("P","
S").replace("O","R")

.replace("N","Q").replace("M","P").replace("L","O").replace("K","N").replace("J","
M").replace("I","L")

.replace("H","K").replace("G","J").replace("F","I").replace("E","H").replace("D","G
").replace("C","F")

.replace("B","E").replace("A","D").replace("^","C").replace("~","B").replace("/","A"
).replace("0","^")

.replace("9","~").replace("8","/").replace("7","0").replace("6","9").replace("5","8").re
place("4","7")

.replace("3","6").replace("2","5").replace("1","4").replace("^","3").replace("~","2").r
eplace("/","1");
    return Password;
}

```



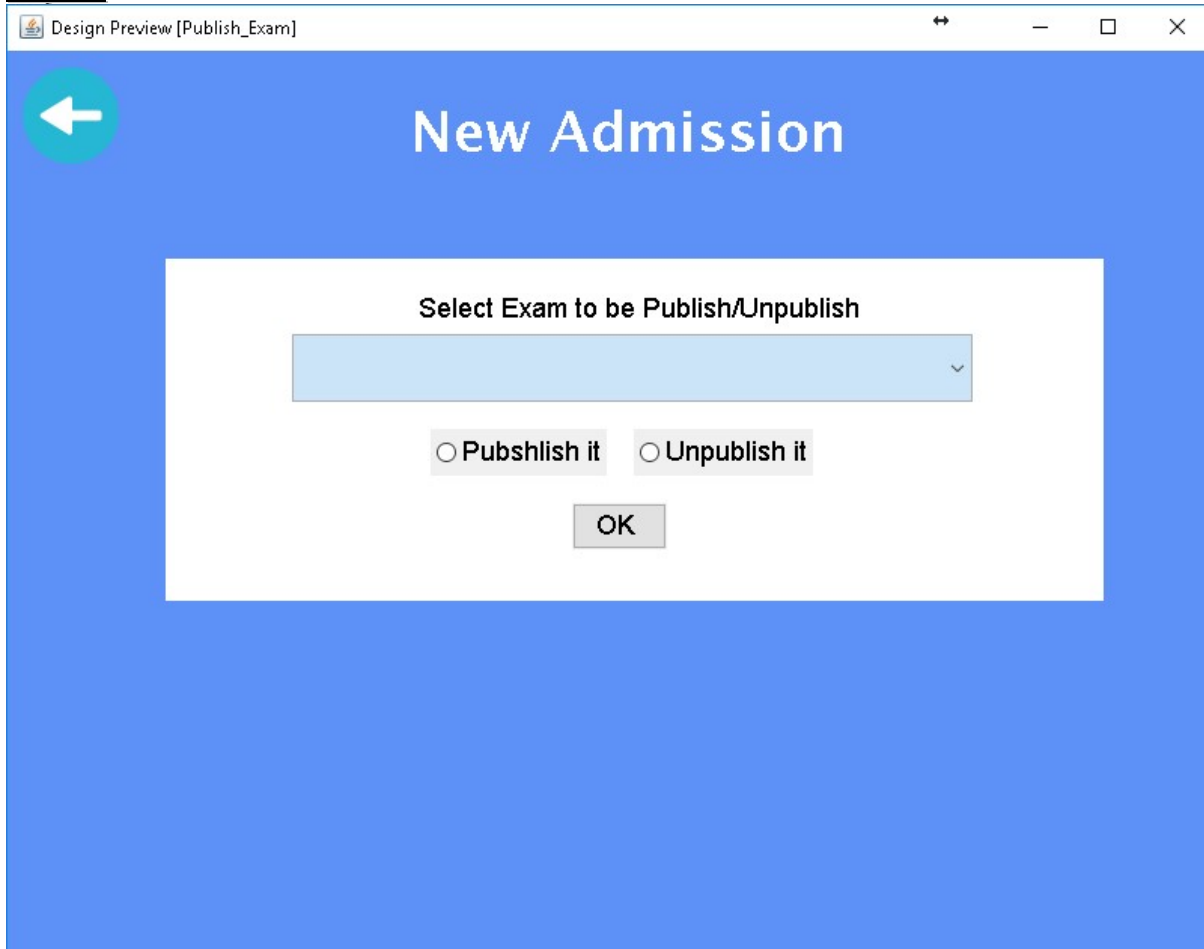
```

public static int getClassCode(JComboBox clas, JComboBox section){
    String cls = "";
    int clsindex = clas.getSelectedIndex();
    int sectionindex = section.getSelectedIndex();
    if(clsindex==0){
        cls = cls+"1";
    }else if(clsindex==1){
        cls = cls+"2";
    }else if(clsindex==2){
        cls = cls+"3";
    }else if(clsindex==3){
        cls = cls+"4";
    }else if(clsindex==4){
        cls = cls+"5";
    }else if(clsindex==5){
        cls = cls+"6";
    }else if(clsindex==6){
        cls = cls+"7";
    }else if(clsindex==7){
        cls = cls+"8";
    }else if(clsindex==8){
        cls = cls+"9";
    }else if(clsindex==9){
        cls = cls+"10";
    }else if(clsindex==10){
        cls = cls+"11";
    }else if(clsindex==11){
        cls = cls+"12";
    }
    if(sectionindex==0){
        cls = cls+"1";
    }else if(sectionindex==1){
        cls = cls+"2";
    }else if(sectionindex==2){
        cls = cls+"3";
    }
    }
    return Integer.parseInt(cls);
}
}

```

Publish_Result.java

Layout



Coding

```
import java.sql.Connection;  
import java.sql.DriverManager;  
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.JComboBox;
```

```

import javax.swing.JOptionPane;

public static String userID=null;

private void okbtnActionPerformed(java.awt.event.ActionEvent evt) {
    String exam = (String) examchk.getSelectedItem();
    int start = exam.indexOf('.')+2;
    int end = exam.indexOf('')-1;
    String ecode = ""+exam.substring(start,end)+"";
    try {
        Class.forName("java.sql.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/brjd", "root", "root");
        Statement stmt = con.createStatement();
        if(pubrb.isSelected()){
            stmt.execute("UPDATE exams SET Published=1 WHERE
ExamCode="+ecode+";");
        }else if(upubrb.isSelected()){
            stmt.execute("UPDATE exams SET Published=0 WHERE
ExamCode="+ecode+";");
        }
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Publish_Exam.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (SQLException ex) {
        Logger.getLogger(Publish_Exam.class.getName()).log(Level.SEVERE, null,
ex);
    }
}

private void backbtnMouseClicked(java.awt.event.MouseEvent evt) {
    this.setVisible(false);
    Admin_Dashboard ad = new Admin_Dashboard();
    ad.userID = userID;
    ad.setVisible(true);
}

@Override
public void show() {
    super.show();
    try {
        Class.forName("java.sql.Driver");

```

```

        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/brjd", "root", "root");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM exams;");
        if (!rs.isBeforeFirst() ) {
            JOptionPane.showMessageDialog(null,"No Results uploaded yet");
        }else{
            JComboBox exams = examchk;
            while(rs.next()){
                int examcode = rs.getInt("ExamCode");
                String examname = rs.getString("ExamName");
                int pub = rs.getInt("Published");
                if(pub==0){
                    exams.addItem(examname+" : "+examcode+" (Not Published)");}
                else if(pub==1){
                    exams.addItem(examname+" : "+examcode+" (Published)");}
            }
        }
        rs.close();
        stmt.close();
        con.close();
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Add_Staff.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(Add_Staff.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

```

Add_Student.java

Layout

Design Preview [Add_Student]

Admission Number	ExamRoll	Class Code
<input type="text"/>	<input type="text"/>	<input type="text"/>
Class Roll	Name	Father 's Name
<input type="text"/>	<input type="text"/>	<input type="text"/>
Mother's Name	Mobile Number	Mobile Number 2
<input type="text"/>	<input type="text"/>	<input type="text"/>
Email	Address	
<input type="text"/>	<input type="text"/>	
Village	State	Pin Code
<input type="text"/>	<input type="text"/>	<input type="text"/>
DOB(yyyy-MM-dd)	Aadhaar Number	Gender
<input type="text"/>	<input type="text"/>	<input type="radio"/> M <input type="radio"/> F
		<input type="button" value="Add"/>

Coding

```
import java.awt.event.KeyEvent;  
import java.sql.*;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.swing.JOptionPane;  
  
public static String userID = null;  
  
private void backbtnMouseClicked(java.awt.event.MouseEvent evt) {  
    this.setVisible(false);  
    Admin_Dashboard ad = new Admin_Dashboard();
```

```

        ad.userID = userID;
        ad.setVisible(true);
    }

    private void addbtnActionPerformed(java.awt.event.ActionEvent evt) {
        String ano = anotxt.getText();
        String eroll = erolltxt.getText();
        String ccode = ccode.txt.getText();
        String croll = crolltxt.getText();
        String name = nametxt.getText();
        String fname = fnametxt.getText();
        String mname = mnametxt.getText();
        String dob = dobtxt.getText();
        String mobile = mobiletxt.getText();
        String email = emailtxt.getText();
        String mobile2 = mobile2txt.getText();
        String address = addresstxt.getText();
        String village = villagetxt.getText();
        String state = statetxt.getText();
        String pin = pintxt.getText();
        String aadhaar = aadhaartxt.getText();
        String gender = null;
        if(mrb.isSelected())
            gender="M";
        else if(frb.isSelected())
            gender="F";
        if(ano.isEmpty() || eroll.isEmpty() || ccode.isEmpty() || croll.isEmpty() ||
name.isEmpty()
        || fname.isEmpty() || mname.isEmpty() ||dob.isEmpty() || mobile.isEmpty()
|| email.isEmpty() ||
        mobile2.isEmpty() || address.isEmpty() || village.isEmpty() ||
state.isEmpty() || pin.isEmpty() ||
        aadhaar.isEmpty() || gender.isEmpty()){
            JOptionPane.showMessageDialog(null,"All the fields are compulsory,Please
fill all the fields");
        }
        String updatequery = "INSERT INTO students
VALUES("+ano+", "+eroll+", "+ccode+", "+croll+", "\"+name+"\", \""+fname+"\", \""+
mname+"\", \""
        +dob+"\", "+mobile+", "\"+email+"\", "+mobile2+", "\"+address+"\", \""+village+"\", \""
+state+"\", "+pin+", "\"+aadhaar+"\", \""+gender+"\"");";
        try {
            Class.forName("java.sql.Driver");

```

```

        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/brjd", "root", "root");
        Statement stmt = con.createStatement();
        stmt.execute(updatequery);
        JOptionPane.showMessageDialog(null,"Student has been added to
database!!!");
        stmt.close();
        con.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null,"An Error
Occured!!\n"+e.getLocalizedMessage());
    }
}

```