Competitive Exams Related Information Management System

A

Report

Submitted in partial fulfilment of the

BE IV SEMESTER DATABASE MANAGEMENT SYSTEMS LAB

INFORMATION TECHNOLOGY

By

V.A.V.V OMKAR <1602-18-737-086>

Under the Guidance of

B. Leelavathy

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2019-2020

BONAFIDE CERTIFICATE

This is to certify that the project report titled "Competitive Exams Related Information Management System" is the project work of Mr. V.A.V.V OMKAR bearing roll number :- 1602-18-737-086 who had completed this project work under my supervision in the IV semester for the academic year 2019-2020.

Signature

External Examiner

Signature

Internal Examiner

Abstract

This project primarily deals with the management of the information related to the competitive exams. With the increasing competitive spirit among the students who resort to attempt numerous such tests, it has become more important to provide and secure the information about such exams at one place so that it's easier to get the gist of what those exams require and test as it can provide everything from how to register, checking the eligibility criteria and how the score is generated. The project is implemented by SQL at the back -end and by JAVA at the front-end. Through this project, an efficient information management system is achieved which helps the students to be updated about various examinations they take and also helps them to be aware of the grading and the minimum qualifying score needed to pass the test.

Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

Introduction

General Information: This project basically gives the essence of an examination help centre because it gives complete information about various examinations in order to ensure that the right information is passed to the students at the right time.

Specific information about my project: The main aim of this project is to create Java's GUI based forms which accept multiple values and finally updating those values in the database using JDBC connectivity.

Requirement Analysis:

List of tables along with their attributes and domain types:

1. *Students:-*
    a) *Name: A multi-valued attribute comprising the name of the student. (Char)*
    b) *D.O.B: An attribute containing the date of birth of the student. (Date)*
    c) *Id: A primary key attribute which contains unique number which distinguishes a student. (Varchar)*
    d) *Educational Qualification: An attribute that comprises of the academic qualifications of a student. (Varchar)*
    e) *Work Experience: An attribute that contains information about industry exposure and internship details, if any. (Number)*
    f) *Grades: A multi-valued attribute that contains details about the grades secured by the student in class 10, class 12, Bachelor's degree. (Varchar)*
2. *Examinations:-*
    a) *Ename: The attribute which has the name of the examination. (Char)*

Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

b) *Pattern:* The attribute containing the details of the exam i.e. whether it's an objective type or subjective type or both. (Char)
c) *Marks:* An attribute used for holding the total marks of the examination. (Number)
d) *Score Validity:* It contains the time duration from the date of results until the final score of the exam is valid i.e. accepted by organizations or universities. (Number)
e) *Grading:* It contains details about the how the exam is graded. (Varchar)
f) *Duration:* It contains the duration of the exam. (Varchar)
g) *Syllabus:* It stores the syllabus of the exam. (Char)
h) *Pre-requisites:* It holds down a basic/essential necessities for a student to appear for the exam. (Varchar)

3. *Eligibility Criteria:*
   a) *Eligibility:* It checks whether a student is eligible for the exam or not. (Varchar)
   b) *Ename:* It contains the name of the exam. (Char)
4. *Performance Report:(weak entity set)*
   a) *Date:* It stores the date of publication of the result. (Date)
   b) *Score:* It consists of the score obtained by the student. (Number)
   c) *Qualifying Marks:* It contains the minimum marks required to qualify the exam. (Number)
   d) *Result:* It stores the final result (pass/fail) of the student. (Char)
5. *Take:* It's a relationship set between Students set and Examinations set.
6. *Check:* It's a relationship set between Students set and Eligibility Criteria set.
7. *Generate:* It's a relationship set between Examinations set and Performance Report set.

Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

Obtain: It's a weak-entity relationship set between Students set and Performance Report set.

Architecture and Technology used:

SOFTWARES USED:

Java Eclipse, Oracle 10g Database, Java SE version 7, SQL Plus.

Java Swing:

Swing is a GUI widget toolkit for Java. It is a part of Oracle's Java Foundation Classes(JFC) - an API for providing a graphical user interface(GUI) for Java programs. Swing was developed to provide a more sophisticated set of GUI components than the earlier AWT. Swing provides a look and feel that emulates the look and feel of several platforms and it also supports a pluggable look and feel that allows applications to possess a look and feel unrelated to the underlying platform. It has more powerful and flexible components when compared to AWT. 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.

SQL:

Structured Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's Relational model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use SQL as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

Design:-

Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

E-R Diagrams:-

Database design:-

The basic requirements for this project are 8 tables and 20 attributes in all. Out of the 8 tables, there are 4 entity sets and 4 relationship sets. The information given below is the list of all attributes with respect to their tables:-

SQL>desc Students;

Name	***Null?***	***Type***
----------	***--------***	***----------***

Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

NAME NOT NULL CHAR(50) SQL>

DOB DATE

ID NOT NULL VARCHAR2(20)

EDUCATIONAL_QUALIFICATIONS VARCHAR2(50)

WORK_EXPERIENCE VARCHAR2(10)

GRADES VARCHAR2(50)

SQL> desc Examinations;

Name Null? Type

--- -------- ----------------------------

ENAME NOT NULL CHAR(50)

PATTERN CHAR(50)

MARKS NUMBER(5)

SCORE_VALIDITY VARCHAR2(20)

GRADING VARCHAR2(50)

DURATION VARCHAR2(30)

SYLLABUS VARCHAR2(50)

PRE_REQUISITES VARCHAR2(50)

SQL> desc Performance_Report;

Name Null? Type

--- -------- ----------------------------

DATE_OF_PUBLISHING DATE

SCORE NUMBER(10)

QUALIFYING_MARKS NUMBER(10)

Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

RESULT NOT NULL VARCHAR2(20)

ID VARCHAR2(20)

SQL> desc Generate;

Name Null? Type

--- -------- ----------------------------

ENAME CHAR(50)

RESULT VARCHAR2(20)

SQL> desc Obtain;

Name Null? Type

--- -------- ----------------------------

ID VARCHAR2(20)

RESULT VARCHAR2(20)

SQL> desc Eligibility_Criteria;

Name Null? Type

--- -------- ----------------------------

ENAME NOT NULL CHAR(50)

ELIGIBILITY NOT NULL CHAR(20)

SQL> desc Take;

Name Null? Type

--- -------- ----------------------------

Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

ID VARCHAR2(20)

ENAME CHAR(50)

SQL> desc Checks;

Name Null? Type

--- -------- ----------------------------

ID VARCHAR2(20)

ELIGIBILITY CHAR(20)

ENAME CHAR(50)

Implementation:-

Front-end program and its connectivity:-

Java-SQL Connectivity using JDBC:

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

The connection to the database can be performed using Java programming (JDBC API) as:

```
public void connectToDB()
    {
        try
        {
```

```
            Class.forName("oracle.jdbc.driver.OracleDriver");
            connection =
DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
B825TEI:1521:xe","system","10Omkar");
            statement = connection.createStatement();

          }
          catch (SQLException connectException)
          {

System.out.println(connectException.getMessage());

System.out.println(connectException.getSQLState());

System.out.println(connectException.getErrorCode());
            System.exit(1);
          }
          catch (Exception e)
          {
               System.err.println("Unable to find and load
driver");
               System.exit(1);
          }
    }
```

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

Table Created in SQL for above mentioned purpose is as:

Create table Performance_Report(date_of _publishing date, score number(10), qualifying_marks number(10), result varchar2(20), id varchar2(20));

Program:

Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.awt.List;
import java.sql.*;
public class PerformanceReport {
	private JPanel p1;
	private JFrame frame;

	private JMenuItem miInsert1,miUpdate1,midelete1,miView1;
	private JLabel lbldate,lblscore,lblresult,lblqualifyingmarks;
	private JTextField txtdate,txtscore,txtresult,txtqualifyingmarks;
	private JButton btn;
	private JTextArea txtmsg;
	//private static Statement stmt;
	private Connection connection;
	private Statement statement;
	public  PerformanceReport(JPanel p1,JFrame frame,JMenuItem miInsert1,JMenuItem
miUpdate1,JMenuItem midelete1,JMenuItem miView1) {
		try
		{
			Class.forName("oracle.jdbc.driver.OracleDriver");
		}
		catch (Exception e)
		{
			System.err.println("Unable to find and load driver");
			System.exit(1);
		}
		connectToDB();
		this.frame=frame;
		this.p1=p1;
		this.miInsert1=miInsert1;
		this.midelete1=midelete1;
		this.miUpdate1=miUpdate1;
		this.miView1=miView1;
		txtdate=new JTextField(20);
		txtscore=new JTextField(20);
		txtresult=new JTextField(20);
		txtqualifyingmarks=new JTextField(20);
		txtmsg=new JTextArea(8,50);
		lbldate=new JLabel("Date:");
		lblscore=new JLabel("Score:");
		lblresult=new JLabel("Result:");
		lblqualifyingmarks=new JLabel("Qualifying marks:");
		//queryHandler();
	}
	public void connectToDB()
    {
		try
		{
```


Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

```
                //connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1522:ORCL","mydbms","mydbms"
);
                    connection =
DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-
B825TEI:1521:xe","system","10Omkar");
                    statement = connection.createStatement();

            }
            catch (SQLException connectException)
            {
              System.out.println(connectException.getMessage());
              System.out.println(connectException.getSQLState());
              System.out.println(connectException.getErrorCode());
              System.exit(1);
            }
    }
      public void registerListenerInterfaces() {
            miInsert1.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent ae) {
                          p1.removeAll();
                          frame.invalidate();
                          frame.validate();
                          frame.repaint();

                          JPanel p=new JPanel();
                          txtdate=new JTextField(20);
                          txtscore=new JTextField(20);
                          txtresult=new JTextField(20);
                          txtqualifyingmarks=new JTextField(20);
                          txtmsg=new JTextArea(8,50);
                          btn=new JButton();
                          //a grid of lbl and txtfield
                          p.add(lbldate);
                          p.add(txtdate);
                          p.add(lblscore);
                          p.add(txtscore);
                          p.add(lblresult);
                          p.add(txtresult);
                          p.add(lblqualifyingmarks);
                          p.add(txtqualifyingmarks);
                          p.setLayout(new GridLayout(5,2));

                          p1.add(p);
                          p1.add(btn);
                          p1.add(txtmsg);//msg text area added to panel
                          btn.setText("INSERT");
                          p1.setLayout(new FlowLayout());
                          frame.add(p1,BorderLayout.CENTER);
                          frame.validate();

                          //register listener
```


Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

```
                        btn.addActionListener(new ActionListener() {

                              @Override
                              public void actionPerformed(ActionEvent e) {
                                    // TODO Auto-generated method stub
                                    try {
                                          Statement statement =
connection.createStatement();
                                          String query= "INSERT INTO performance
report VALUES(" + txtdate.getText() + ", " + "'" + txtscore.getText() + "'," + "'" +
txtresult.getText() + ","+ txtqualifyingmarks.getText()+")";
                                          int i = statement.executeUpdate(query);
                                                txtmsg.append("\nInserted " + i
+ " rows successfully");
                                    } catch (SQLException e1) {
                                          // TODO Auto-generated catch block
                                    //     e1.printStackTrace();
                        JOptionPane.showMessageDialog(frame,"Please Enter Valid Input");
                                          txtmsg.append(e1.getMessage());
                                    }
                              }
                        });

                  }
            });

            miUpdate1.addActionListener(new ActionListener() {
                  public void actionPerformed(ActionEvent ae) {

                        p1.removeAll();
                        frame.invalidate();
                        frame.validate();
                        frame.repaint();
                        JPanel p=new JPanel();
                        txtdate=new JTextField(20);
                        txtscore=new JTextField(20);
                        txtresult=new JTextField(20);
                        txtqualifyingmarks=new JTextField(20);
                        txtmsg=new JTextArea(8,50);
                        txtmsg.setEditable(false);
                        btn=new JButton();
                        List idlist =new List(10);
                        try
                        {
                         ResultSet rs = statement.executeQuery("SELECT date FROM
product");
                          while (rs.next())
                          {
                                idlist.add(rs.getString("date"));
                          }
                        }
                        catch (SQLException e)
```


Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

```
                        {
                JOptionPane.showMessageDialog(frame,"Please Enter Valid Input");
                          txtmsg.append(e.getMessage());
                        }
                        p1.add(idlist);
                        p.add(lbldate);
                        p.add(txtdate);
                        p.add(lblscore);
                        p.add(txtscore);
                        p.add(lblresult);
                        p.add(txtresult);
                        p.add(lblqualifyingmarks);
                        p.add(txtqualifyingmarks);
                        p.setLayout(new GridLayout(5,2));
                        p1.add(p);
                        p1.add(btn);
                        p1.add(txtmsg);
                        btn.setText("Update");
                        p1.setLayout(new FlowLayout());

                        frame.add(p1,BorderLayout.CENTER);
                        frame.validate();

                        idlist.addItemListener(new ItemListener()
                        {
                              @Override
                              public void itemStateChanged(ItemEvent arg0) {
                                    // TODO Auto-generated method stub

                                    try
                                    {
                                       ResultSet rs =
statement.executeQuery("SELECT * FROM performance report where date
="+idlist.getSelectedItem());
                                          rs.next();
                                          txtdate.setText(rs.getString("DATE"));

      txtscore.setText(rs.getString("SCORE"));

      txtresult.setText(rs.getString("RESULT"));

      txtqualifyingmarks.setText(rs.getString("QUALIFYING MARKS"));
                                    }
                                    catch (SQLException selectException)
                                    {

      txtmsg.append(selectException.getMessage());
                                    }
                              }
                        });
```


Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

```
                        btn.addActionListener(new ActionListener() {

                                @Override
                                public void actionPerformed(ActionEvent e) {
                                        // TODO Auto-generated method stub
                                        try
                                        {
                                                Statement statement =
connection.createStatement();
                                                txtmsg.append("UPDATE performance
report "
                                                        + "SET score='" +
txtscore.getText() + "', "
                                                        + "result='" +
txtresult.getText() + "', "
                                                        + ",qualifying marks ="+
txtqualifyingmarks.getText() + " WHERE date = "
                                                        +
idlist.getSelectedItem()+"\n");
                                                int i = statement.executeUpdate("UPDATE
performance report "
                                                + "SET score='" + txtscore.getText() +
"', "
                                               + "result='" + txtresult.getText() + "',
"
                                                +",qualifying marks ="+
txtqualifyingmarks.getText() + " WHERE date = "
                                                + idlist.getSelectedItem());
                                                System.out.println("successful");

                                                txtmsg.append("\nUpdated " + i + " rows
successfully");
                                                //idlist.removeAll();
                                                //loadSailors();
                                        }
                                        catch (SQLException insertException)
                                        {
        JOptionPane.showMessageDialog(frame,"Please Enter Valid Input");
        txtmsg.append(insertException.getMessage());
                                        }

                                }
                        });
                }
        });


        midelete1.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent ae) {

                        p1.removeAll();
                        frame.invalidate();
```


Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

```
                        frame.validate();
                        frame.repaint();
                        JPanel p=new JPanel();
                        txtdate=new JTextField(20);
                        txtscore=new JTextField(20);
                        txtresult=new JTextField(20);
                        txtqualifyingmarks=new JTextField(20);
                        txtmsg=new JTextArea(8,50);
                        btn=new JButton();
                        List idlist =new List(10);
                        try
                        {
                         ResultSet rs = statement.executeQuery("SELECT date FROM
performance report");
                          while (rs.next())
                          {
                                 idlist.add(rs.getString("date"));
                          }
                        }
                        catch (SQLException e)
                        {
                          txtmsg.append(e.getMessage());
                        }
                        p1.add(idlist);
                        p.add(lbldate);
                        p.add(txtdate);
                        p.add(lblscore);
                        p.add(txtscore);
                        p.add(lblresult);
                        p.add(txtresult);
                        p.add(lblqualifyingmarks);
                        p.add(txtqualifyingmarks);
                        p.setLayout(new GridLayout(5,2));
                        p1.add(p);
                        p1.add(btn);
                        p1.add(txtmsg);
                        btn.setText("Delete");
                        p1.setLayout(new FlowLayout());

                        frame.add(p1,BorderLayout.CENTER);
                        frame.validate();

                        idlist.addItemListener(new ItemListener()
                        {
                               @Override
                               public void itemStateChanged(ItemEvent arg0) {
                                      // TODO Auto-generated method stub

                                      try
                                      {
```


Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

```
                                        ResultSet rs =
statement.executeQuery("SELECT * FROM performance report where DATE
="+idlist.getSelectedItem());
                                                rs.next();
                                                txtdate.setText(rs.getString("DATE"));

        txtscore.setText(rs.getString("SCORE"));

        txtresult.setText(rs.getString("RESULT"));

        txtqualifyingmarks.setText(rs.getString("QUALIFYING MARKS"));
                                        }
                                        catch (SQLException selectException)
                                        {

        txtmsg.append(selectException.getMessage());
                                        }
                                }
                        });


                        btn.addActionListener(new ActionListener() {

                                @Override
                                public void actionPerformed(ActionEvent e) {
                                        // TODO Auto-generated method stub
                                        try
                                        {
                                                Statement statement =
connection.createStatement();
                                                int i = statement.executeUpdate("delete
from performance report where date=" + idlist.getSelectedItem());
                                                txtmsg.append("\nDeleted " + i + " rows
successfully");
                                                //idlist.removeAll();
                                                //loadSailors();
                                                txtdate.setText(null);
                                                txtscore.setText(null);
                                                txtresult.setText(null);
                                                txtqualifyingmarks.setText(null);
                                                idlist.removeAll();
                                        }
                                        catch (SQLException insertException)
                                        {
        JOptionPane.showMessageDialog(frame,"Please Enter Valid Input");

        txtmsg.append(insertException.getMessage());
                                        }

                                }
                        });
                }
```


Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

```
		});

		miView1.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				// TODO Auto-generated method stub
			//	Statement statement=connection.createStatement();
				p1.removeAll();
				frame.invalidate();
				frame.validate();
				frame.repaint();
				p1.add(txtmsg);
				frame.add(p1,BorderLayout.CENTER);
				frame.validate();

				try {
					ResultSet rs=statement.executeQuery("select * from
performance report");
					String s="";
					while(rs.next())
						s=s+(rs.getString(1)+"  "+rs.getInt(2)+"
"+rs.getString(3)+"  "+rs.getInt(4)+"\n");
						txtmsg.setText(s);
				} catch (SQLException e1) {
					// TODO Auto-generated catch block
					e1.printStackTrace();
				}

			}
		});


	}


}
```

User Interface:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class ExaminationUI extends JFrame {
	private static final long serialVersionUID = 1L;
	private JMenu
mnustudents,mnuexaminations,mnuperformance_report,mnueligibility_criteria,mnucheck,mn
utake,mnugenerate,mnuobtain;
	private JMenuBar mnuBar;
	private JMenuItem miInsert1,miUpdate1,miDelete1,miView1;
	private JMenuItem miInsert2,miUpdate2,miDelete2,miView2;
	private JMenuItem miInsert3,miUpdate3,miDelete3,miView3;
	private JMenuItem miInsert4,miUpdate4,miDelete4,miView4;
```


Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

```
	private JMenuItem miInsert5,miUpdate5,miDelete5,miView5;
	private JMenuItem miInsert6,miUpdate6,miDelete6,miView6;
	private JMenuItem miInsert7,miUpdate7,miDelete7,miView7;
	private JMenuItem miInsert8,miUpdate8,miDelete8,miView8;
	private JTextField txtField;

	static JPanel p1;

	void initialize()
	{
		//po=new JPanel();
		p1=new JPanel();
		mnustudents=new JMenu("Students");
		mnuexaminations=new JMenu("Examinations");
		mnueligibility_criteria=new JMenu("Eligibility Criteria");
		mnuperformance_report=new JMenu("Performance Report");
		mnucheck=new JMenu("Checks");
		mnutake=new JMenu("Take");
		mnugenerate=new JMenu("Generate");
		mnuobtain=new JMenu("Obtain");
		mnuBar=new JMenuBar();

		miInsert1=new JMenuItem("Insert");
		miUpdate1=new JMenuItem("Update");
		miDelete1=new JMenuItem("Delete");
		miView1=new JMenuItem("View");

		miInsert2=new JMenuItem("Insert");
		miUpdate2=new JMenuItem("Update");
		miDelete2=new JMenuItem("Delete");
		miView2=new JMenuItem("View");

		miInsert3=new JMenuItem("Insert");
		miUpdate3=new JMenuItem("Update");
		miDelete3=new JMenuItem("Delete");
		miView3=new JMenuItem("View");

		miInsert4=new JMenuItem("Insert");
		miUpdate4=new JMenuItem("Update");
		miDelete4=new JMenuItem("Delete");
		miView4=new JMenuItem("View");

		miInsert5=new JMenuItem("Insert");
		miUpdate5=new JMenuItem("Update");
		miDelete5=new JMenuItem("Delete");
		miView5=new JMenuItem("View");

		miInsert6=new JMenuItem("Insert");
		miUpdate6=new JMenuItem("Update");
		miDelete6=new JMenuItem("Delete");
		miView6=new JMenuItem("View");
```


Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

```
		miInsert7=new JMenuItem("Insert");
		miUpdate7=new JMenuItem("Update");
		miDelete7=new JMenuItem("Delete");
		miView7=new JMenuItem("View");

		miInsert8=new JMenuItem("Insert");
		miUpdate8=new JMenuItem("Update");
		miDelete8=new JMenuItem("Delete");
		miView8=new JMenuItem("View");



		txtField = new JTextField("Competitive exams related information
management system");
		txtField.setFont(new Font("Serif", Font.PLAIN, 25));
		txtField.setEditable(false);
	//	po.setBackground(Color.MAGENTA);
	}
	void addComponentsToFrame()
	{
		mnustudents.add(miInsert1);
		mnustudents.add(miUpdate1);
		mnustudents.add(miDelete1);
		mnustudents.add(miView1);

		mnuexaminations.add(miInsert2);
		mnuexaminations.add(miUpdate2);
		mnuexaminations.add(miDelete2);
		mnuexaminations.add(miView2);

		mnueligibility_criteria.add(miInsert3);
		mnueligibility_criteria.add(miUpdate3);
		mnueligibility_criteria.add(miDelete3);
		mnueligibility_criteria.add(miView3);

		mnuperformance_report.add(miInsert4);
		mnuperformance_report.add(miUpdate4);
		mnuperformance_report.add(miDelete4);
		mnuperformance_report.add(miView4);

		mnucheck.add(miInsert5);
		mnucheck.add(miUpdate5);
		mnucheck.add(miDelete5);
		mnucheck.add(miView5);

		mnutake.add(miInsert6);
		mnutake.add(miUpdate6);
		mnutake.add(miDelete6);
		mnutake.add(miView6);

		mnugenerate.add(miInsert7);
		mnugenerate.add(miUpdate7);
```


Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

```
                mnugenerate.add(miDelete7);
                mnugenerate.add(miView7);

                mnuobtain.add(miInsert8);
                mnuobtain.add(miUpdate8);
                mnuobtain.add(miDelete8);
                mnuobtain.add(miView8);
                mnuBar.add(mnustudents);
                mnuBar.add(mnuexaminations);
                mnuBar.add(mnueligibility_criteria);
                mnuBar.add(mnuperformance_report);
                mnuBar.add(mnucheck);
                mnuBar.add(mnutake);
                mnuBar.add(mnugenerate);
                mnuBar.add(mnuobtain);
                setJMenuBar(mnuBar);


                p1.setLayout(new BorderLayout());
                p1.add(txtField,BorderLayout.CENTER);

                this.setLayout(new BorderLayout());
                add(p1,BorderLayout.CENTER);

        }
        void register()
        {
                Students t1=new
Students(p1,ExaminationUI.this,miInsert1,miUpdate1,miDelete1,miView1);
                t1.registerListenerInterfaces();
                ExaminationsTable t2=new
ExaminationsTable(p1,ExaminationUI.this,miInsert2,miUpdate2,miDelete2,miView2);
                t2.registerListenerInterfaces();
                EligibilityCriteria t3=new
EligibilityCriteria(p1,ExaminationUI.this,miInsert3,miUpdate3,miDelete3,miView3);
                t3.registerListenerInterfaces();
                PerformanceReport t4=new PerformanceReport(p1, ExaminationUI.this,
miInsert4, miUpdate4, miDelete4, miView4);
                t4.registerListenerInterfaces();
                Checks t5=new
Checks(p1,ExaminationUI.this,miInsert5,miUpdate5,miDelete5,miView5);
                t5.registerListenerInterfaces();
                Take t6=new Take(p1, ExaminationUI.this, miInsert6,miUpdate6, miDelete6,
miView6);
                t6.registerListenerInterfaces();
                Generate t7=new
Generate(p1,ExaminationUI.this,miInsert7,miUpdate7,miDelete7,miView7);
                t7.registerListenerInterfaces();
                Obtain t8=new
Obtain(p1,ExaminationUI.this,miInsert8,miUpdate8,miDelete8,miView8);
                t8.registerListenerInterfaces();
                 addWindowListener(new WindowAdapter(){
```


Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

```
                        public void windowClosing(WindowEvent evt)
                        {
                                Int a
=JOptionPane.showConfirmDialog(ExaminationUI.this,"Are you sure?", "This will close
the UI", JOptionPane.OK_CANCEL_OPTION, JOptionPane.INFORMATION_MESSAGE);
                                if(a==JOptionPane.YES_OPTION)

ExaminationUI.this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

                        }
                    });


        }


        public ExaminationUI()
        {
                initialize();
                addComponentsToFrame();
                register();
                pack();
                setTitle("Competitive exams related information management system");
                setSize(600,500);
                setVisible(true);
                //setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        }

}
```

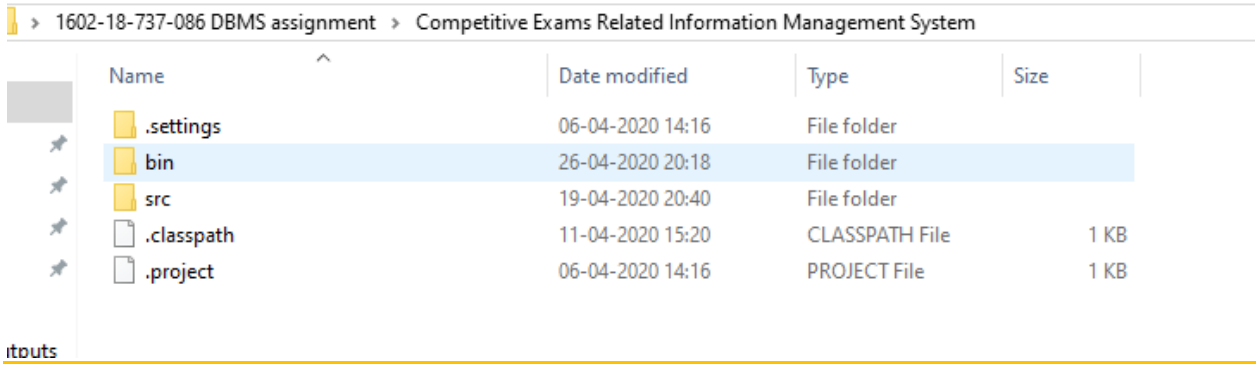
Main:

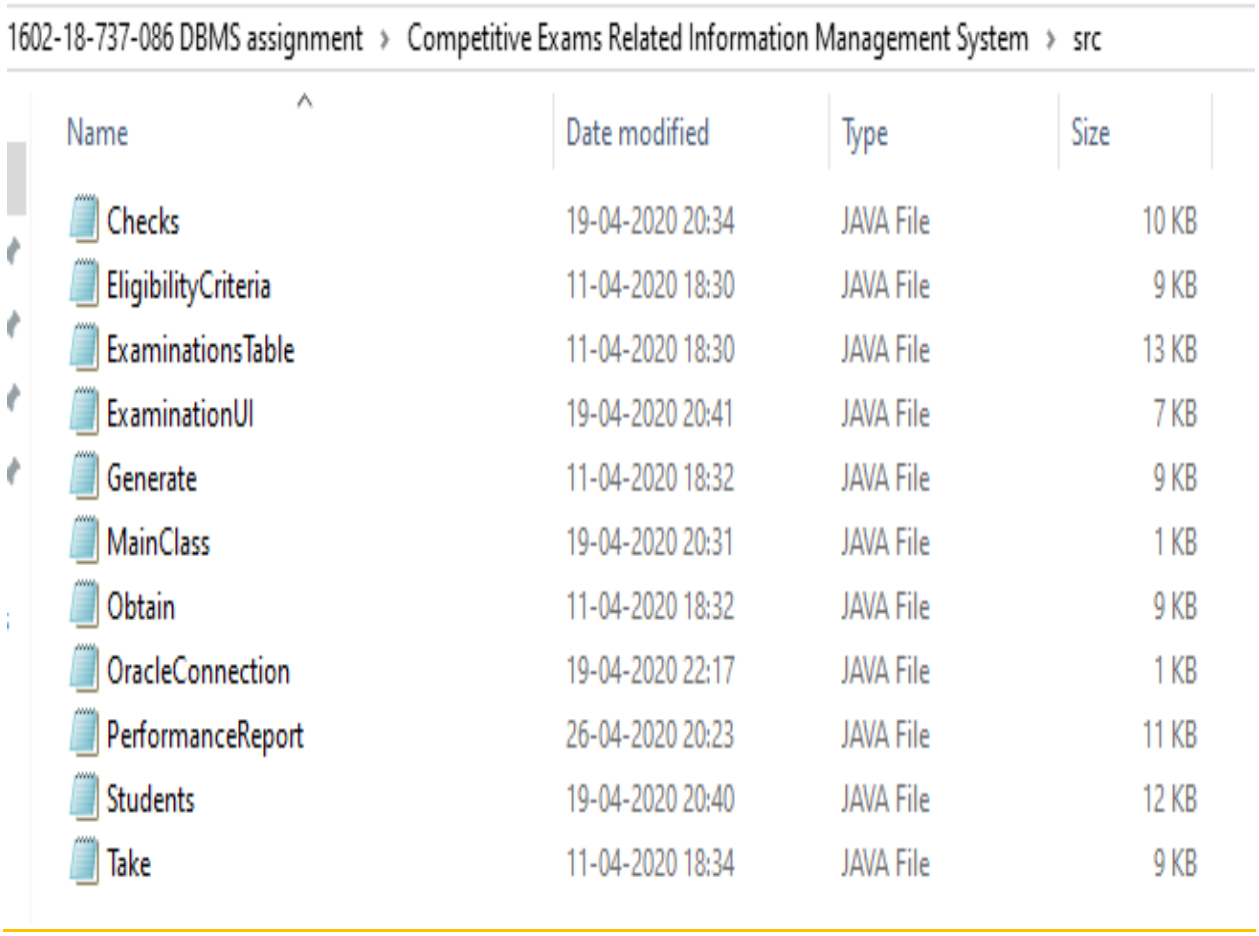
```
public class MainClass {
        public static void main(String[] args)
        {
                ExaminationUI eui=new ExaminationUI();
        }
}
```

GitHub links and folder structure:

https://github.com/Omkar-perfectionist/Myfirstproject

Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR





Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

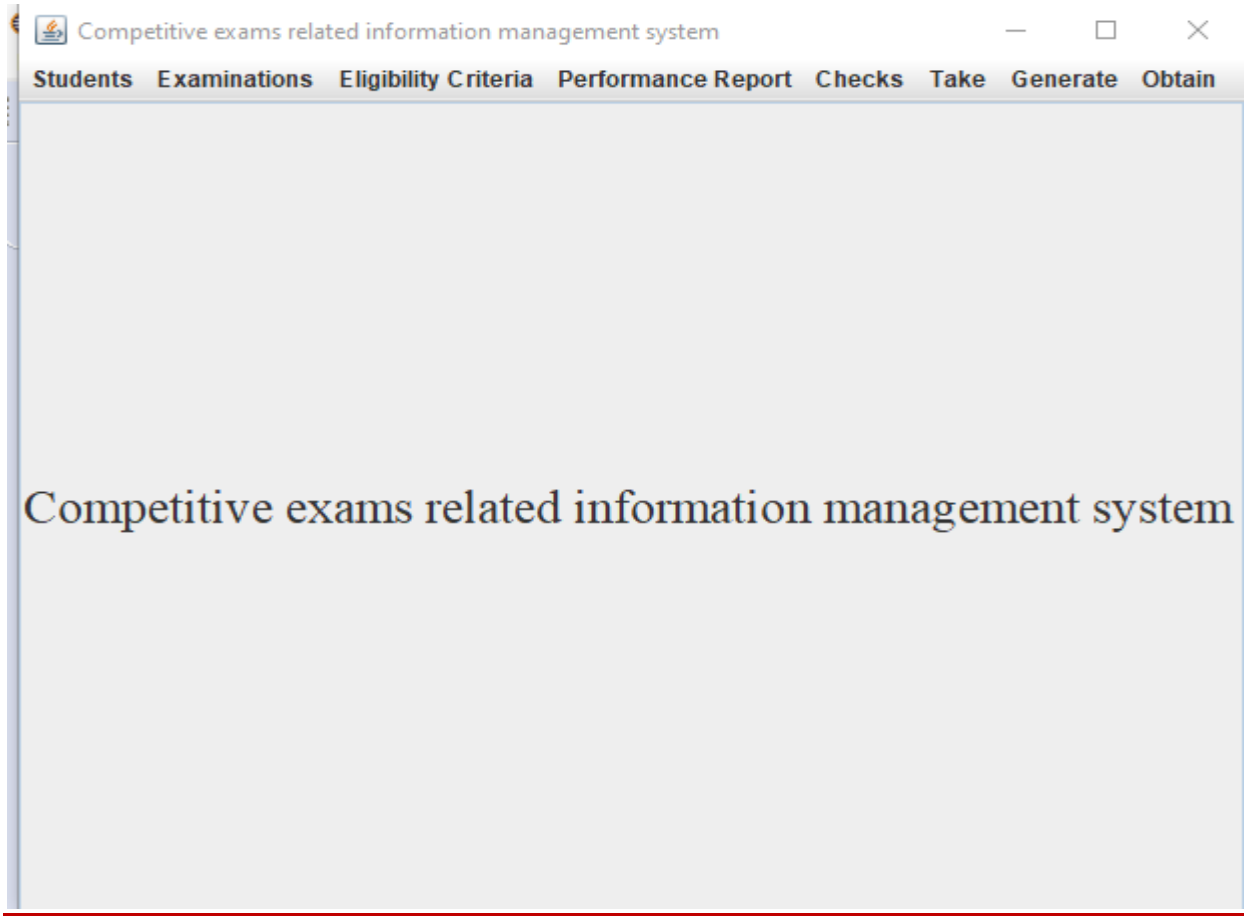
1602-18-737-086 DBMS assignment › Competitive Exams Related Information Management System › bin

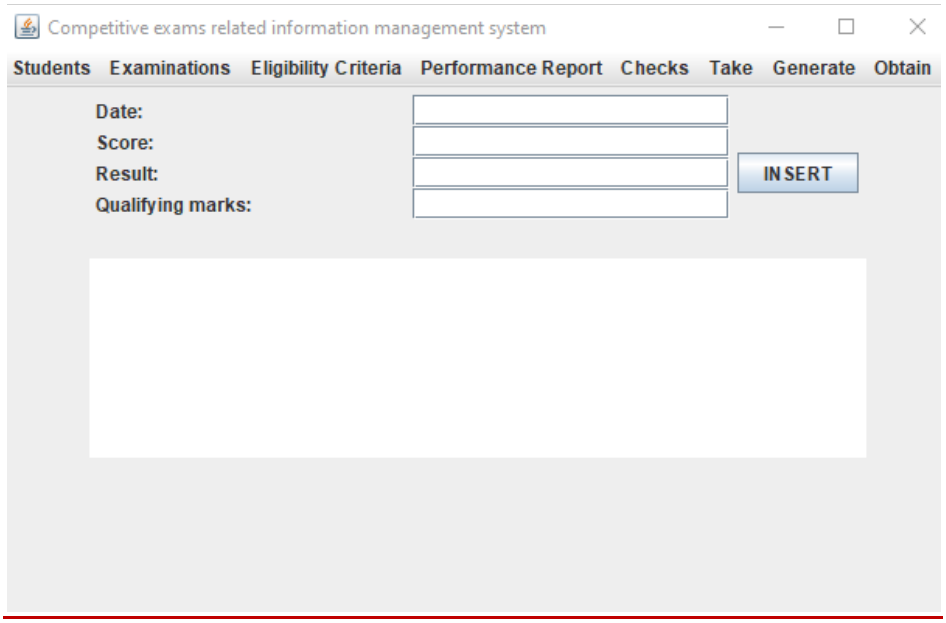
Name	Date modified	Type	Size
ExaminationsTable$4.class	26-04-2020 20:18	CLASS File	3 KB
ExaminationsTable.class	26-04-2020 20:18	CLASS File	4 KB
ExaminationUI$1.class	26-04-2020 20:18	CLASS File	1 KB
ExaminationUI.class	26-04-2020 20:18	CLASS File	6 KB
Generate$1$1.class	26-04-2020 20:18	CLASS File	2 KB
Generate$1.class	26-04-2020 20:18	CLASS File	3 KB
Generate$2$1.class	26-04-2020 20:18	CLASS File	3 KB
Generate$2$2.class	26-04-2020 20:18	CLASS File	2 KB
Generate$2.class	26-04-2020 20:18	CLASS File	4 KB
Generate$3$1.class	26-04-2020 20:18	CLASS File	3 KB
Generate$3$2.class	26-04-2020 20:18	CLASS File	2 KB
Generate$3.class	26-04-2020 20:18	CLASS File	4 KB
Generate$4.class	26-04-2020 20:18	CLASS File	3 KB
Generate.class	26-04-2020 20:18	CLASS File	4 KB
MainClass.class	26-04-2020 20:18	CLASS File	1 KB
Obtain$1$1.class	26-04-2020 20:18	CLASS File	2 KB
Obtain$1.class	26-04-2020 20:18	CLASS File	3 KB
Obtain$2$1.class	26-04-2020 20:18	CLASS File	3 KB
Obtain$2$2.class	26-04-2020 20:18	CLASS File	2 KB
Obtain$2.class	26-04-2020 20:18	CLASS File	4 KB
Obtain$3$1.class	26-04-2020 20:18	CLASS File	3 KB
Obtain$3$2.class	26-04-2020 20:18	CLASS File	2 KB
Obtain$3.class	26-04-2020 20:18	CLASS File	4 KB
Obtain$4.class	26-04-2020 20:18	CLASS File	3 KB
Obtain.class	26-04-2020 20:18	CLASS File	4 KB
OracleConnection.class	26-04-2020 20:18	CLASS File	2 KB
PerformanceReport$1$1.class	26-04-2020 20:23	CLASS File	3 KB
PerformanceReport$1.class	26-04-2020 20:23	CLASS File	3 KB
PerformanceReport$2$1.class	26-04-2020 20:23	CLASS File	3 KB

Testing:

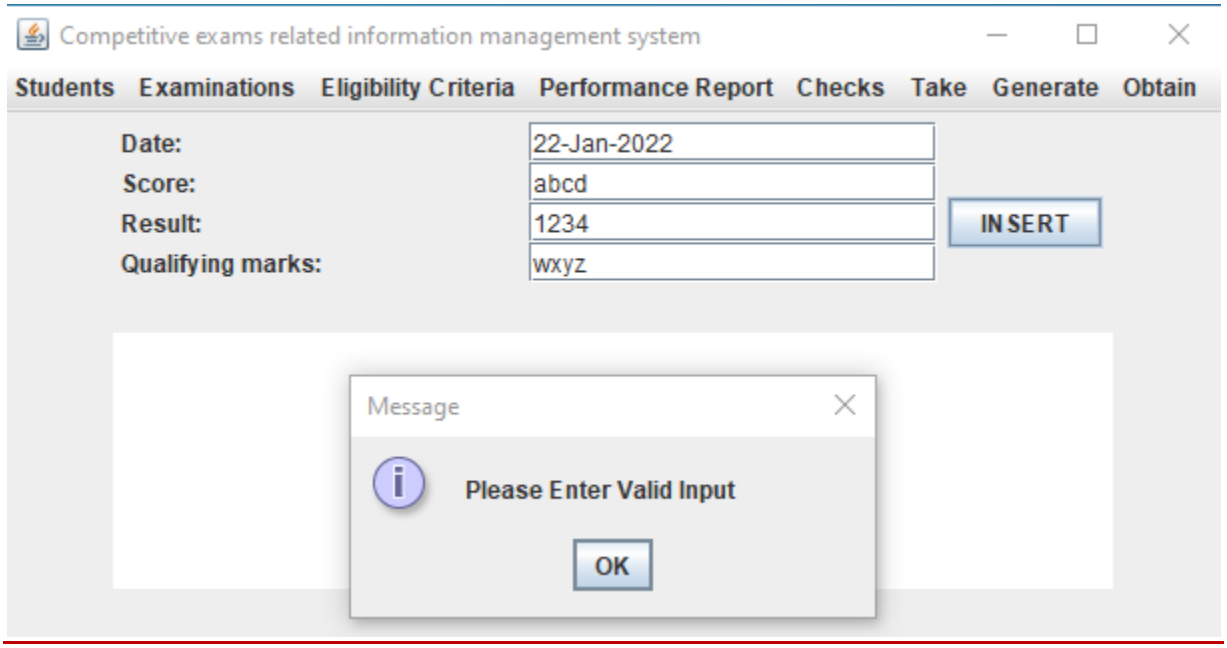
Java GUI testing:

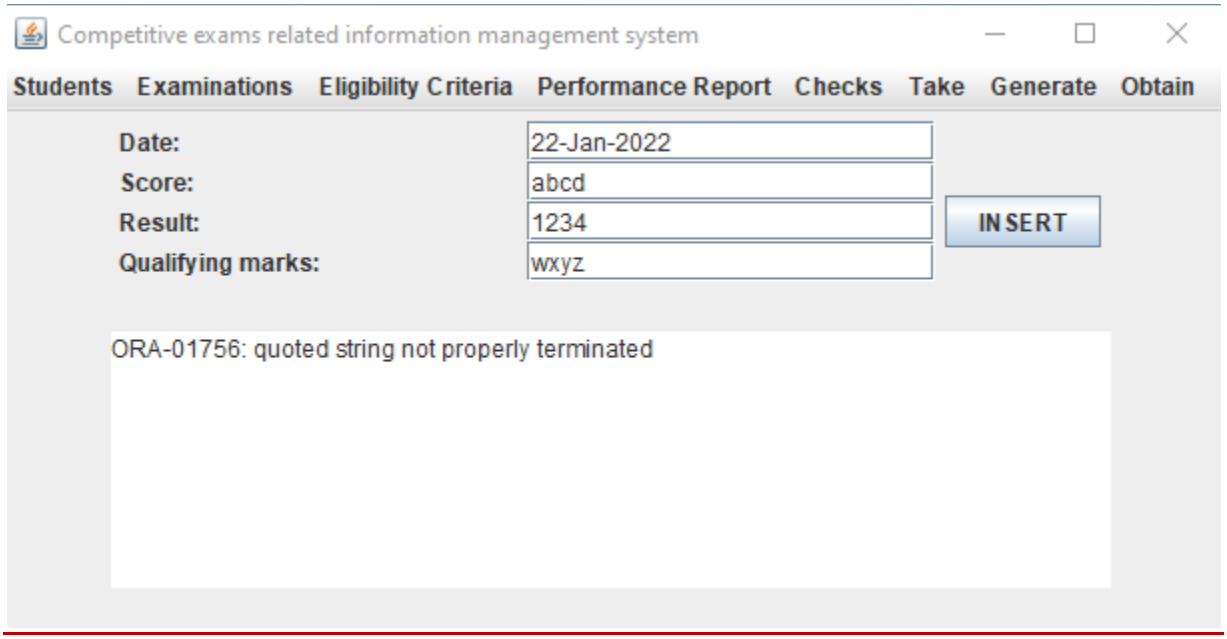
Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR



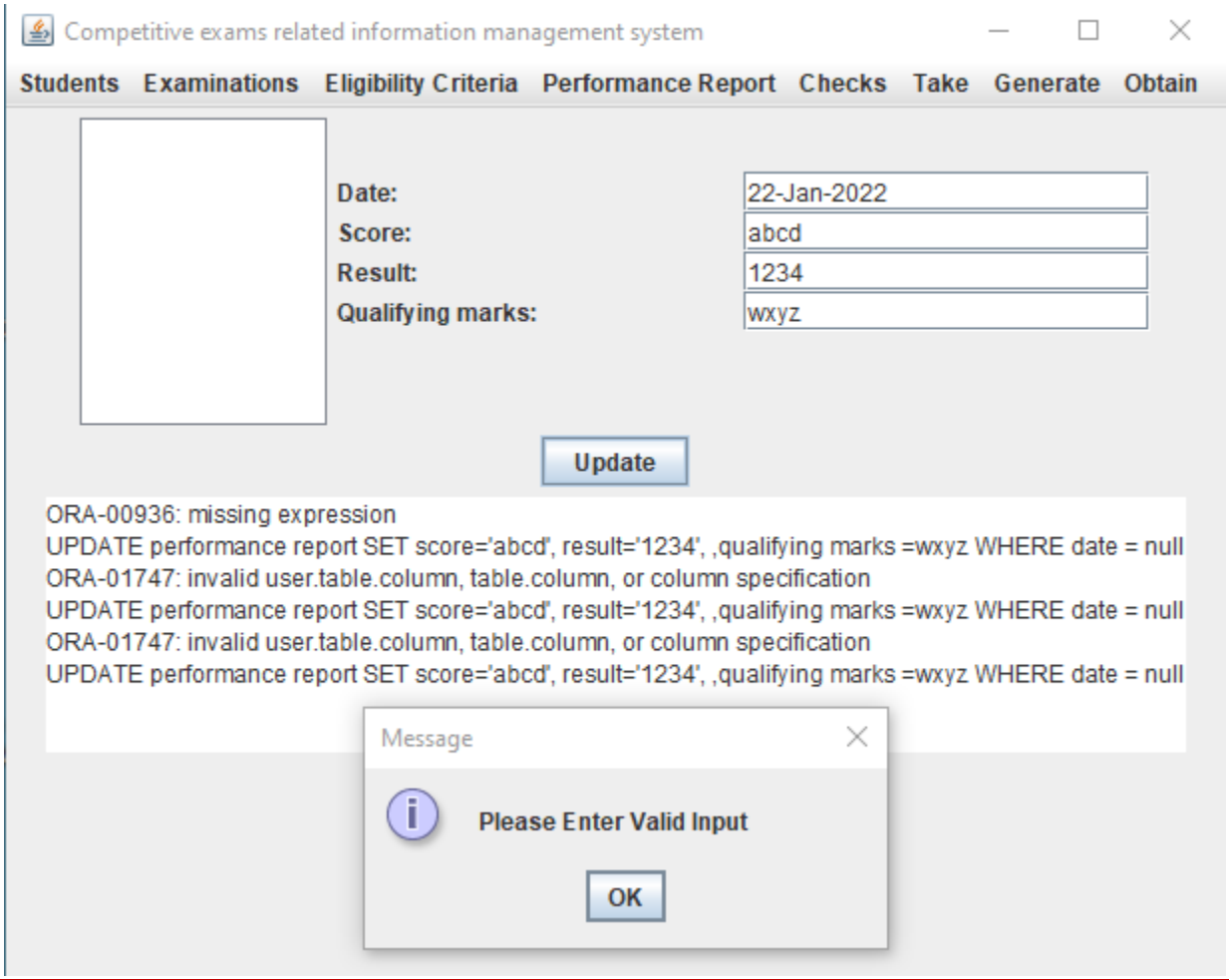


Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR





Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

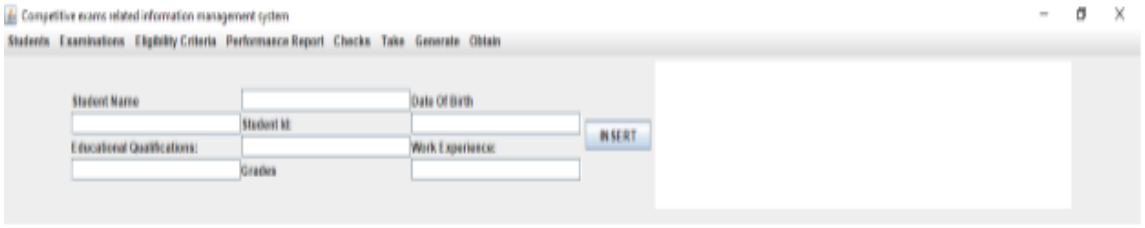


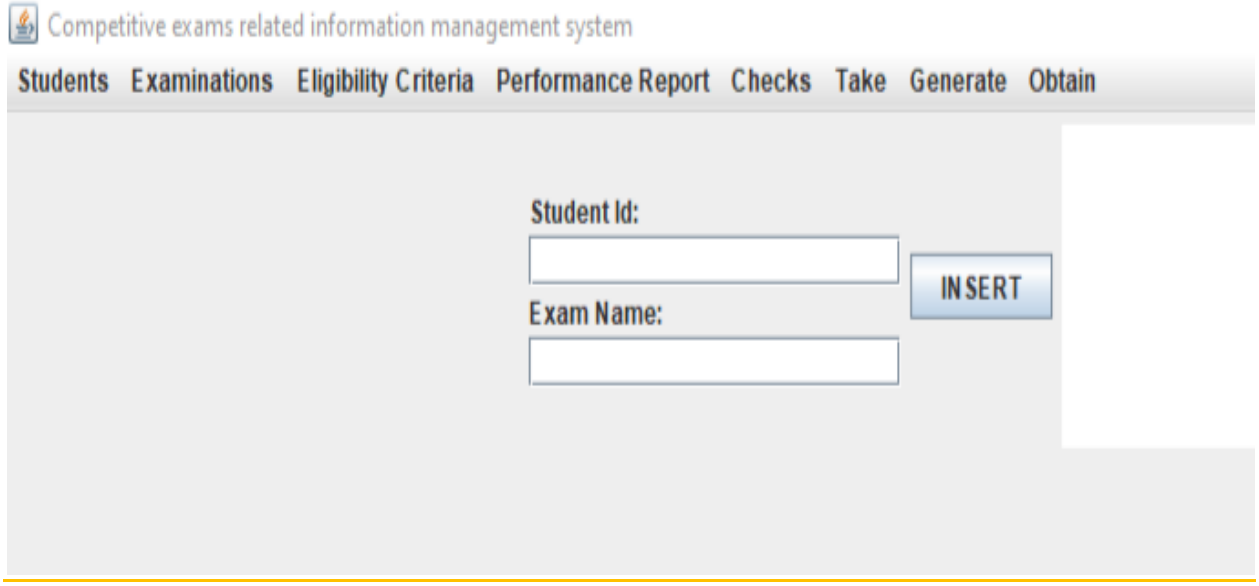
Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

Results:-

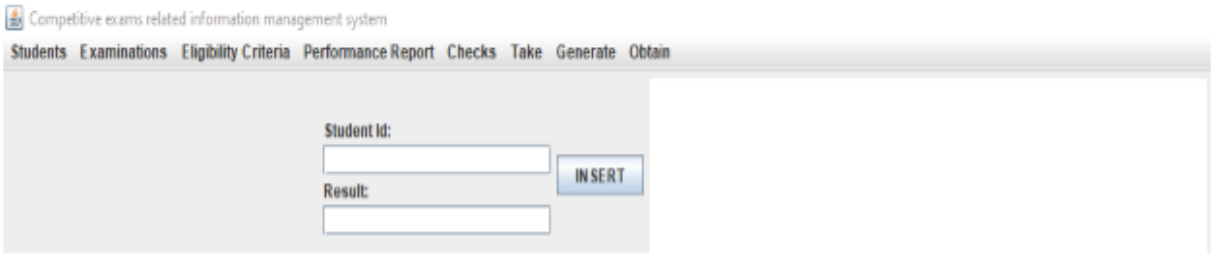
Given below are the following screenshots belong to the tables represented by the front-end forms using Graphical User Interface(GUI):

Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR





Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

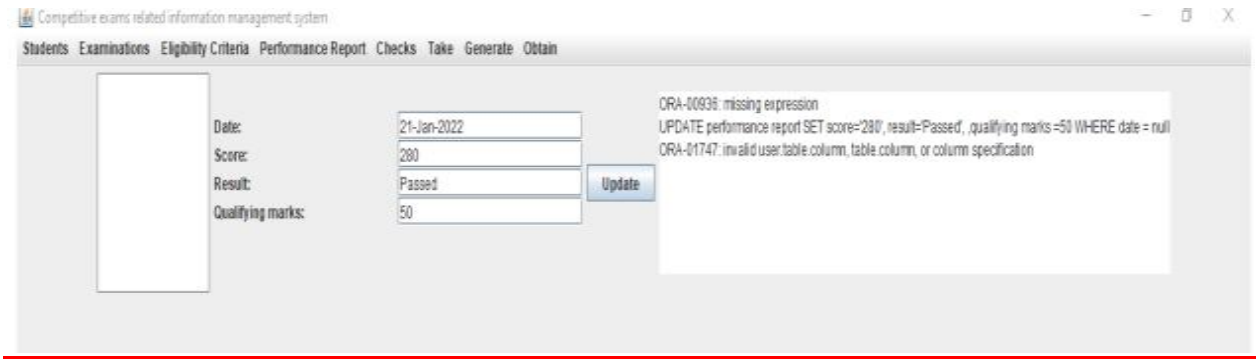


DML Operations on the Performance_Report table:-

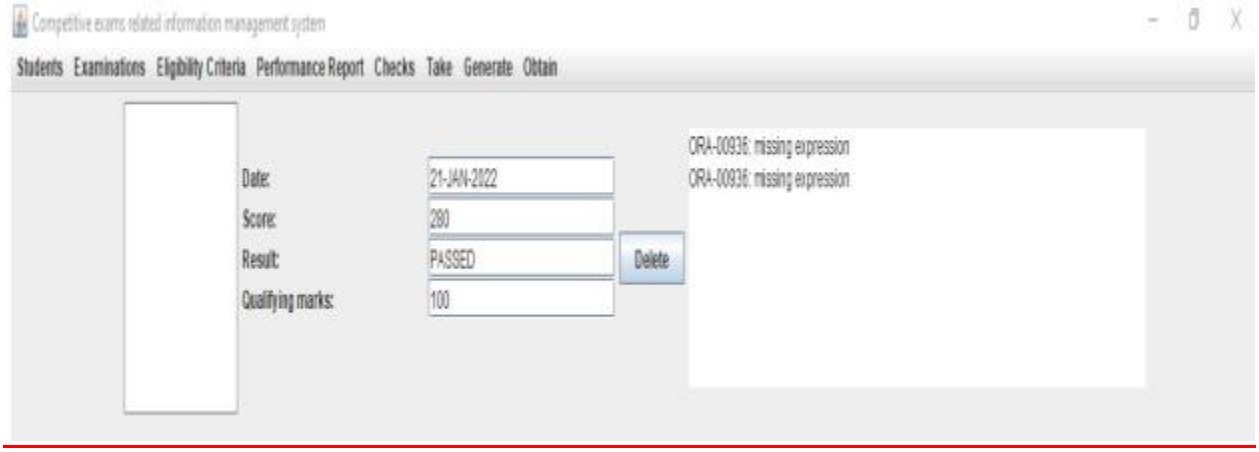
Insert:-

Update:-

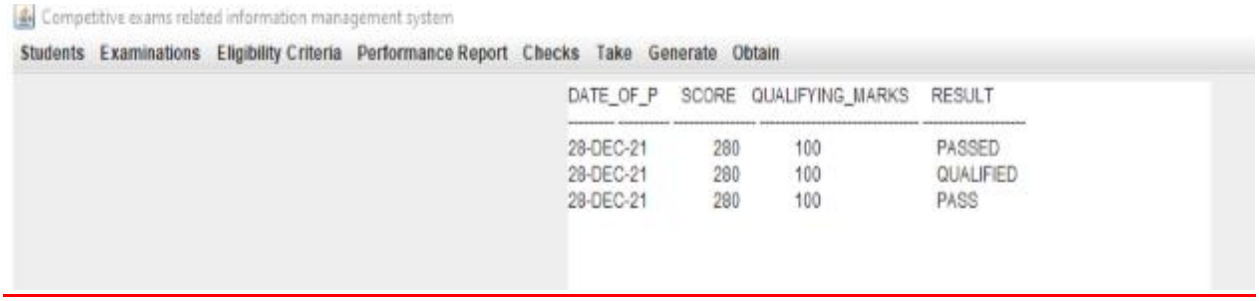
Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR



Delete:-

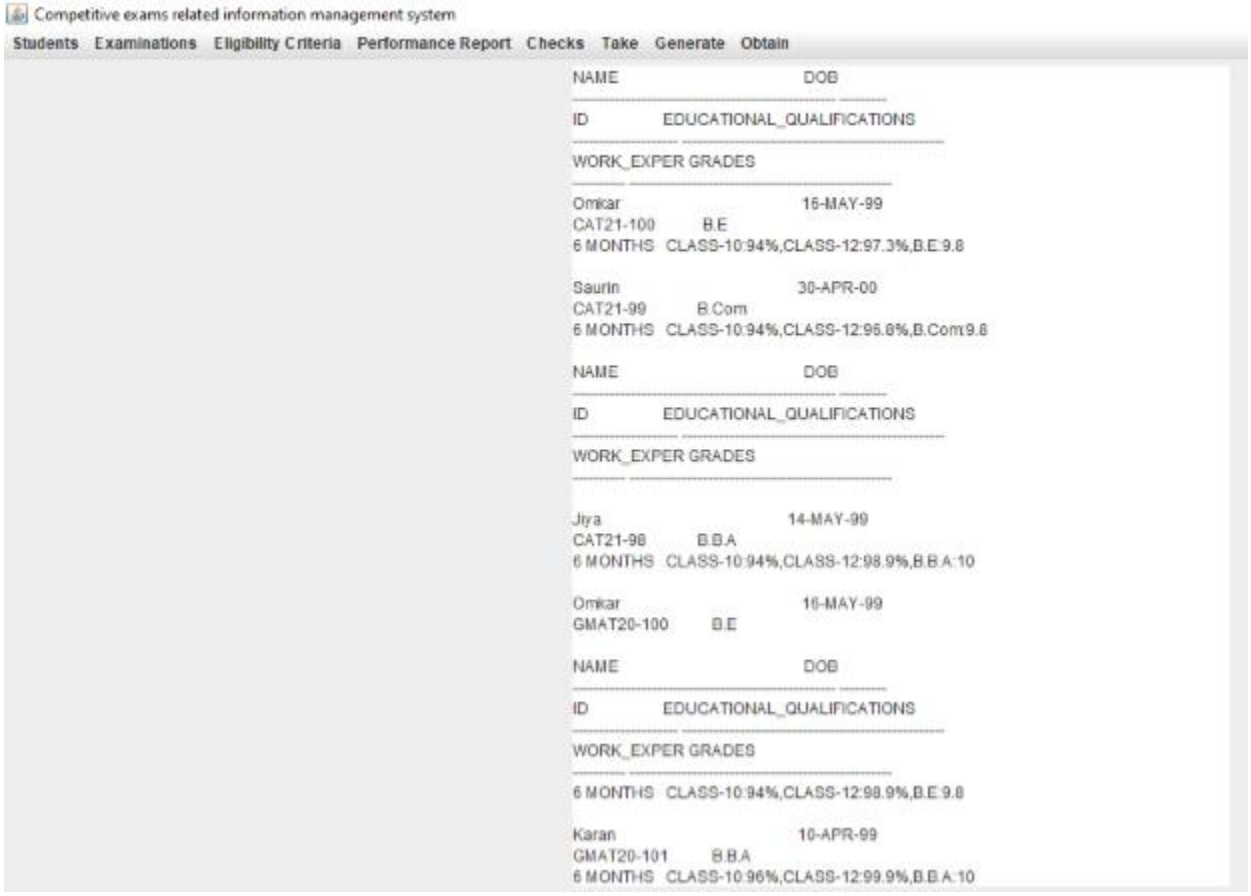


View:-



Views of other tables:-

Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR



Discussion and future work:-

Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR

While executing the project, I've acquired the knowledge of creating a front-end application through Java and linking it to the database at the back-end. This project efficiently stores the data in the tables and the data can be manipulated with ease through a user-friendly and visually appealing graphical interface. Coming to future work, this project can be implemented in the form of a web-application. Since this project deals with the examinations, it might be an ideal web-application.

References:

- *https://docs.oracle.com/javase/8/docs/index.html*
- *https://www.javatpoint.com/dbms-tutorial*
- *https://www.sqlines.com/articles/java/sql*
- *https://www.studytonight.com/dbms/*

Conclusion:

The above mini project titled "Competitive Exams Related Information Management System" was completed successfully.

Roll number: 1602-18-737-086
Name: V.A.V.V OMKAR