

JAVA SWING BASED- E MEDICINE SALES SYSTEM- SQL CONNECTIVITY USING JDBC

A

Report

*Submitted in partial fulfilment of the
Requirements for the award of the Degree of*

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

DORNALA PRAJWAL REDDY <1602-19-737-086>

Under the Guidance of

MS B. Leelavathy



**Department of Information Technology
Vasavi College of Engineering (Autonomous)
(Affiliated to Osmania University)
Ibrahimbagh, Hyderabad-31**

2020-2021

BONAFIDE CERTIFICATE

This is to certify that this project report titled '**Student Notes Management System**' is a project work of Mr. Dornala Prajwal Reddy bearing roll no. 1602-19-737-086 who carried out the project under my supervision in the IV semester for the academic year 2020- 2021.

Signature

External examiner

Signature

Internal examiner

ABSTRACT

Student Notes Management system is an application designed especially for studying pursuing different colleges, courses and subjects. It is an easy interactive application used to write notes for daily classes and store them in such a way that they have all the details of subject, topic, professor, college and class date are easily searched back based on date of class and topic and can be continued to study.

` The Main Objective of this is to virtually write notes during the on-going online classes. The User can enter course and subject details and write notes for each class, topic, professor, time and date.

For a same course and subject, a student can write notes for any number of topics and classes.

REQUIREMENT ANALYSIS

List of tables:

- COLLEGE
- STUDENT
- COURSE
- SUBJECT
- NOTES

List of attributes with their domain types:

COLLEGE :

- College_code: varchar2(5)
- College_name : varchar2(30)
- College_address : varchar2(100)

STUDENT:

- Student_id : number(5)
- Student_name : varchar2(20)
- Student_contact : varchar2(10)
- Student_address : varchar2(50)
- College_code : varchar2(5)

SUBJECT:

- Subject_id : number(5)
- Subject_name : varchar2(20)
- Teacher_contact : varchar2(10)
- Teacher_name : varchar2(20)

COURSE:

- Course_id: number(5)
- Course_name : varchar2(20)
- Subject_id : number(5)
- Course_credits : number(5)
- Enrollment_year : number(5)
- Course_start_date : date
- Course_end_date : date
- College_code : varchar2(5)
- Student_id : number(5)

TITLE: STUDENT NOTES MANAGEMENT SYSTEM

NOTES:

- Notes_id : number(5)
- Subject_id : number(5)
- Course_id : number(5)
- Notes_write : varchar2(50)
- Dateofclass : date

AIM AND PRIORITY OF THE PROJECT

To create a Java GUI based Student Notes Management System which takes the values like college code, student id, course id etc. from the student who uses the application. It also stores academic information of the user like contact and teacher details and here only admin can insert or update or delete the information from tables. The user can only update his notes, course details, subject details and can view other tables. These values are to be updated or deleted in the database using JDBC connectivity.

ARCHITECTURE AND TECHNOLOGY USED:

SOFTWARE USED:

Java Eclipse, Oracle 11g Database, Java SE version 8, SQL Plus.

Java SWING:

SWING is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.

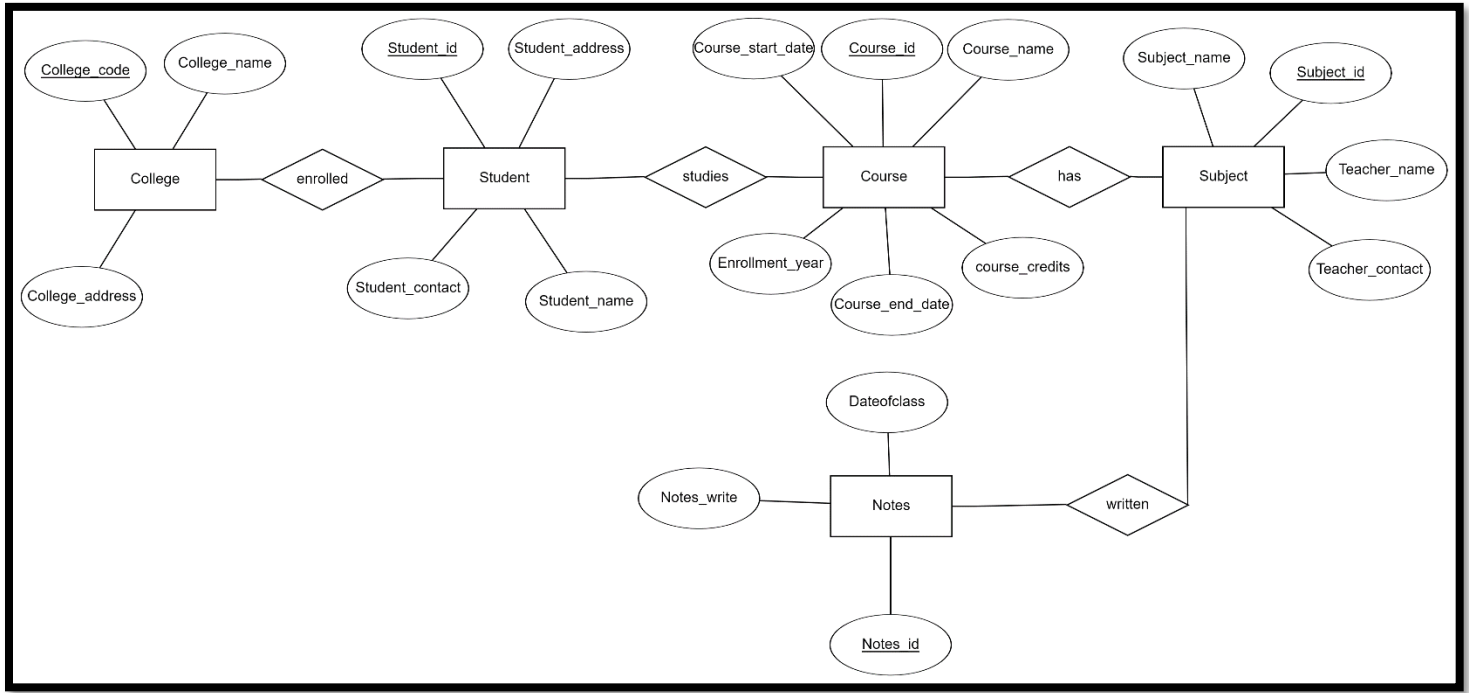
Swing was developed to provide a more sophisticated set of GUI components than the earlier AWT. Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than 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:

Structure 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:

ER DIAGRAM:



DATA DESIGN:

Mapping Cardinalities and Participation Constraints:

Every student registered need to have one (only one) profile mandatorily. Therefore, this is a total participation and one-to-one relationship.

Every student has to enter the details of the college he/she enrolled in it. It is a total participation and one-to-one relationship.

Also, every student should be only enrolled in one course, so that he/she would be able to write notes for its subjects

Therefore, it is a total participation from the courses.

A user can have any number of subjects at the same time. Notes can be written for any subjects, hence this is a many-to-many relationship.

DDL COMMANDS:

TABLE: COLLEGE-

```
1.create table college(  
  college_code varchar2(5),  
  college_name varchar2(30) NOT NULL,  
  college_address varchar2(100) NOT NULL,  
  CONSTRAINT pk_college_code PRIMARY KEY (college_code));
```

```
SQL> create table college(  
  2  college_code varchar2(5),  
  3  college_name varchar2(30) NOT NULL,  
  4  college_address varchar2(100) NOT NULL,  
  5  CONSTRAINT pk_college_code PRIMARY KEY(college_code));  
Table created.
```

TABLE: STUDENT-

```
2.create table student(  
  student_id number(5),  
  student_name varchar2(20) NOT NULL,  
  student_contact varchar2(10) NOT NULL,  
  student_address varchar2(50) NOT NULL,  
  college_code varchar2(5) NOT NULL,  
  CONSTRAINT fk_employee_cc FOREIGN KEY (college_code) REFERENCES  
college(college_code) ON DELETE CASCADE,  
  CONSTRAINT pk_student_id PRIMARY KEY (student_id)) ;
```

```
SQL> create table student(  
  2  student_id number(5),  
  3  student_name varchar2(20) NOT NULL,  
  4  student_contact varchar2(10) NOT NULL,  
  5  student_address varchar2(50) NOT NULL,  
  6  college_code varchar2(5) NOT NULL,  
  7  CONSTRAINT fk_student_cc FOREIGN KEY(college_code)  
  8  REFERENCES college(college_code) ON DELETE CASCADE,  
  9  CONSTRAINT pk_student_id PRIMARY KEY(student_id));  
Table created.
```

TABLE: SUBJECT-

3. create table subject(
subject_id number(5),
subject_name varchar2(20) NOT NULL,
teacher_contact varchar2(10) NOT NULL,
teacher_name varchar2(20) NOT NULL,
CONSTRAINT pk_subject_id PRIMARY KEY (subject_id));

```
SQL> create table subject(  
2  subject_id number(5),  
3  subject_name varchar2(20) NOT NULL,  
4  teacher_contact varchar2(10) NOT NULL,  
5  teacher_name varchar2(20) NOT NULL,  
6  CONSTRAINT pk_subject_id PRIMARY KEY(subject_id));  
  
Table created.
```

TABLE: COURSE-

4. create table course(
course_id number(5),
course_name varchar2(20) NOT NULL,
subject_id number(5) NOT NULL,
course_credits number(5) NOT NULL,
enrollment_year number(5) NOT NULL,
course_start_date DATE NOT NULL,
course_end_date DATE NOT NULL,
college_code varchar2(5) NOT NULL,
student_id number(5) NOT NULL,
CONSTRAINT fk_course_subi FOREIGN KEY (subject_id) REFERENCES client(subject_id)
ON DELETE CASCADE,
CONSTRAINT fk_course_cc FOREIGN KEY (college_code) REFERENCES
college(college_code) ON DELETE CASCADE,
CONSTRAINT fk_course_si FOREIGN KEY (student_id) REFERENCES student(student_id)
ON DELETE CASCADE,
CONSTRAINT pk_course_id PRIMARY KEY (course_id));

```
SQL> create table course(  
 2  course_id number(5),  
 3  course_name varchar2(20) NOT NULL,  
 4  subject_id number(5) NOT NULL,  
 5  course_credits number(5) NOT NULL,  
 6  enrollment_year number(5) NOT NULL,  
 7  course_start_date DATE NOT NULL,  
 8  course_end_date DATE NOT NULL,  
 9  college_code varchar2(5) NOT NULL,  
10  student_id number(5) NOT NULL,  
11  CONSTRAINT fk_course_subi FOREIGN KEY(subject_id) REFERENCES subject(subject_id) ON DELETE CASCADE,  
12  CONSTRAINT fk_course_cc FOREIGN KEY(college_code) REFERENCES college(college_code) ON DELETE CASCADE,  
13  CONSTRAINT fk_course_si FOREIGN KEY(student_id) REFERENCES student(student_id) ON DELETE CASCADE,  
14  CONSTRAINT pk_course_id PRIMARY KEY(course_id));
```

Table created.

TABLE: NOTES-

```
5. create table notes(  
  notes_id number(5) NOT NULL  
  subject_id number(5) NOT NULL,  
  course_id number(5) NOT NULL,  
  notes_write number(5) NOT NULL,  
  dateofclass DATE NOT NULL,  
  CONSTRAINT fk_notes_subi FOREIGN KEY (subject_id) REFERENCES  
subject(subject_id) ON DELETE CASCADE,  
  CONSTRAINT fk_notes_ci FOREIGN KEY (course_id) REFERENCES course(course_id)  
ON DELETE CASCADE,  
  CONSTRAINT pk_notes_ni PRIMARY KEY (notes_id));
```

```
SQL> create table notes(  
 2  notes_id number(5) NOT NULL,  
 3  subject_id number(5) NOT NULL,  
 4  course_id number(5) NOT NULL,  
 5  notes_write varchar2(50) NOT NULL,  
 6  dateofclass DATE NOT NULL,  
 7  CONSTRAINT fk_notes_si FOREIGN KEY(subject_id) REFERENCES subject(subject_id) ON DELETE CASCADE,  
 8  CONSTRAINT fk_notes_ci FOREIGN KEY(course_id) REFERENCES course(course_id) ON DELETE CASCADE,  
 9  CONSTRAINT pk_notes_ni PRIMARY KEY(notes_id));
```

Table created.

ALL TABLES:

```
SQL> desc college;
```

Name	Null?	Type
COLLEGE_CODE	NOT NULL	VARCHAR2(5)
COLLEGE_NAME	NOT NULL	VARCHAR2(30)
COLLEGE_ADDRESS	NOT NULL	VARCHAR2(100)

```
SQL> desc student;
```

Name	Null?	Type
STUDENT_ID	NOT NULL	NUMBER(5)
STUDENT_NAME	NOT NULL	VARCHAR2(20)
STUDENT_CONTACT	NOT NULL	VARCHAR2(10)
STUDENT_ADDRESS	NOT NULL	VARCHAR2(50)
COLLEGE_CODE	NOT NULL	VARCHAR2(5)

```
SQL> desc subject;
```

Name	Null?	Type
SUBJECT_ID	NOT NULL	NUMBER(5)
SUBJECT_NAME	NOT NULL	VARCHAR2(20)
TEACHER_CONTACT	NOT NULL	VARCHAR2(10)
TEACHER_NAME	NOT NULL	VARCHAR2(20)

```
SQL> desc course;
```

Name	Null?	Type
COURSE_ID	NOT NULL	NUMBER(5)
COURSE_NAME	NOT NULL	VARCHAR2(20)
SUBJECT_ID	NOT NULL	NUMBER(5)
COURSE_CREDITS	NOT NULL	NUMBER(5)
ENROLLMENT_YEAR	NOT NULL	NUMBER(5)
COURSE_START_DATE	NOT NULL	DATE
COURSE_END_DATE	NOT NULL	DATE
COLLEGE_CODE	NOT NULL	VARCHAR2(5)
STUDENT_ID	NOT NULL	NUMBER(5)

```
SQL> desc notes;
```

Name	Null?	Type
NOTES_ID	NOT NULL	NUMBER(5)
SUBJECT_ID	NOT NULL	NUMBER(5)
COURSE_ID	NOT NULL	NUMBER(5)
NOTES_WRITE	NOT NULL	VARCHAR2(50)
DATEOFCCLASS	NOT NULL	DATE

TITLE: STUDENT NOTES MANAGEMENT SYSTEM

DML COMMANDS:

1. COLLEGE:

```
SQL> insert into college values('&college_code','&college_name','&college_address');
Enter value for college_code: 101
Enter value for college_name: vasavi
Enter value for college_address: ibrahimbagh
old 1: insert into college values('&college_code','&college_name','&college_address')
new 1: insert into college values('101','vasavi','ibrahimbagh')

1 row created.

SQL> /
Enter value for college_code: 102
Enter value for college_name: vjit
Enter value for college_address: miyapur
old 1: insert into college values('&college_code','&college_name','&college_address')
new 1: insert into college values('102','vjit','miyapur')

1 row created.

SQL> /
Enter value for college_code: 103
Enter value for college_name: snit
Enter value for college_address: himampur
old 1: insert into college values('&college_code','&college_name','&college_address')
new 1: insert into college values('103','snit','himampur')

1 row created.

SQL> /
Enter value for college_code: 104
Enter value for college_name: gsit
Enter value for college_address: uppall
old 1: insert into college values('&college_code','&college_name','&college_address')
new 1: insert into college values('104','gsit','uppall')

1 row created.

SQL> /
Enter value for college_code: 105
Enter value for college_name: cbit
Enter value for college_address: sherling
old 1: insert into college values('&college_code','&college_name','&college_address')
new 1: insert into college values('105','cbit','sherling')

1 row created.
```

2. STUDENT:

```
SQL> insert into student values(&student_id,&student_name,&student_contact,&student_address,&college_code);
Enter value for student_id: 086
Enter value for student_name: Amar
Enter value for student_contact: 8272838811
Enter value for student_address: Guntur
Enter value for college_code: 101
old 1: insert into student values(&student_id,&student_name,&student_contact,&student_address,&college_code')
new 1: insert into student values(086,'Amar','8272838811','Guntur','101')

1 row created.

SQL> /
Enter value for student_id: 088
Enter value for student_name: Deepu
Enter value for student_contact: 9273782111
Enter value for student_address: Karimnagar
Enter value for college_code: 102
old 1: insert into student values(&student_id,&student_name,&student_contact,&student_address,&college_code')
new 1: insert into student values(088,'Deepu','9273782111','Karimnagar','102')

1 row created.

SQL> /
Enter value for student_id: 082
Enter value for student_name: Weesa
Enter value for student_contact: 8428192467
Enter value for student_address: Siddipet
Enter value for college_code: 103
old 1: insert into student values(&student_id,&student_name,&student_contact,&student_address,&college_code')
new 1: insert into student values(082,'Weesa','8428192467','Siddipet','103')

1 row created.

SQL> /
Enter value for student_id: 075
Enter value for student_name: Joshua
Enter value for student_contact: 9639628367
Enter value for student_address: Hyderabad
Enter value for college_code: 104
old 1: insert into student values(&student_id,&student_name,&student_contact,&student_address,&college_code')
new 1: insert into student values(075,'Joshua','9639628367','Hyderabad','104')

1 row created.
```

3. SUBJECT:


```
SQL> insert into subject values(&subject_id,&subject_name,&teacher_contact,&teacher_name');
Enter value for subject_id: 401
Enter value for subject_name: Physics
Enter value for teacher_contact: 7292887654
Enter value for teacher_name: Iqbal
old 1: insert into subject values(&subject_id,&subject_name,&teacher_contact,&teacher_name')
new 1: insert into subject values(401,'Physics','7292887654','Iqbal')

1 row created.

SQL> /
Enter value for subject_id: 402
Enter value for subject_name: Maths
Enter value for teacher_contact: 9428341782
Enter value for teacher_name: Imam
old 1: insert into subject values(&subject_id,&subject_name,&teacher_contact,&teacher_name')
new 1: insert into subject values(402,'Maths','9428341782','Imam')

1 row created.

SQL> /
Enter value for subject_id: 403
Enter value for subject_name: DBMS
Enter value for teacher_contact: 7258279812
Enter value for teacher_name: Indra
old 1: insert into subject values(&subject_id,&subject_name,&teacher_contact,&teacher_name')
new 1: insert into subject values(403,'DBMS','7258279812','Indra')

1 row created.

SQL> /
Enter value for subject_id: 404
Enter value for subject_name: PS
Enter value for teacher_contact: 7367215267
Enter value for teacher_name: Esha
old 1: insert into subject values(&subject_id,&subject_name,&teacher_contact,&teacher_name')
new 1: insert into subject values(404,'PS','7367215267','Esha')

1 row created.

SQL> /
Enter value for subject_id: 405
Enter value for subject_name: HumanValues
Enter value for teacher_contact: 9364262987
Enter value for teacher_name: Tarun
old 1: insert into subject values(&subject_id,&subject_name,&teacher_contact,&teacher_name')
new 1: insert into subject values(405,'HumanValues','9364262987','Tarun')

1 row created.
```

4. COURSE:

TITLE: STUDENT NOTES MANAGEMENT SYSTEM

```
SQL> insert into course values(&course_id,&course_name,&subject_id,&course_credits,&enrollment_year,&course_start_date,&course_end_date,&college_code,&student_id);
Enter value for course_id: 732
Enter value for course_name: ME
Enter value for subject_id: 402
Enter value for course_credits: 5
Enter value for enrollment_year: 2017
Enter value for course_start_date: TO_DATE('20/06/2017','DD/MM/YY')
Enter value for course_end_date: TO_DATE('20/12/2017','DD/MM/YY')
Enter value for college_code: 102
Enter value for student_id: 88
old 1: insert into course values(&course_id,&course_name,&subject_id,&course_credits,&enrollment_year,&course_start_date,&course_end_date,&college_code,&student_id)
new 1: insert into course values(732,'ME',402,5,2017,TO_DATE('20/06/2017','DD/MM/YY'),TO_DATE('20/12/2017','DD/MM/YY'),'102',88)

1 row created.

SQL> /
Enter value for course_id: 733
Enter value for course_name: BCA
Enter value for subject_id: 403
Enter value for course_credits: 7
Enter value for enrollment_year: 2019
Enter value for course_start_date: TO_DATE('12/01/2019','DD/MM/YY')
Enter value for course_end_date: TO_DATE('11/11/2019','DD/MM/YY')
Enter value for college_code: 103
Enter value for student_id: 82
old 1: insert into course values(&course_id,&course_name,&subject_id,&course_credits,&enrollment_year,&course_start_date,&course_end_date,&college_code,&student_id)
new 1: insert into course values(733,'BCA',403,7,2019,TO_DATE('12/01/2019','DD/MM/YY'),TO_DATE('11/11/2019','DD/MM/YY'),'103',82)

1 row created.

SQL> /
Enter value for course_id: 734
Enter value for course_name: MCA
Enter value for subject_id: 404
Enter value for course_credits: 4
Enter value for enrollment_year: 2020
Enter value for course_start_date: TO_DATE('11/02/2020','DD/MM/YY')
Enter value for course_end_date: TO_DATE('21/08/2020','DD/MM/YY')
Enter value for college_code: 104
Enter value for student_id: 75
old 1: insert into course values(&course_id,&course_name,&subject_id,&course_credits,&enrollment_year,&course_start_date,&course_end_date,&college_code,&student_id)
new 1: insert into course values(734,'MCA',404,4,2020,TO_DATE('11/02/2020','DD/MM/YY'),TO_DATE('21/08/2020','DD/MM/YY'),'104',75)

1 row created.
```

5. NOTES:

TITLE: STUDENT NOTES MANAGEMENT SYSTEM

```
SQL> insert into notes values(&notes_id,&subject_id,&course_id,&'&notes_write',&dateofclass);
Enter value for notes_id: 501
Enter value for subject_id: 401
Enter value for course_id: 731
Enter value for notes_write: Today there has been discussion about
Enter value for dateofclass: TO_DATE('22-04-2019','DD/MM/YY')
old 1: insert into notes values(&notes_id,&subject_id,&course_id,&'&notes_write',&dateofclass)
new 1: insert into notes values(501,401,731,'Today there has been discussion about',TO_DATE('22-04-2019','DD/MM/YY'))

1 row created.

SQL> /
Enter value for notes_id: 502
Enter value for subject_id: 402
Enter value for course_id: 732
Enter value for notes_write: It has been address about
Enter value for dateofclass: TO_DATE('22-07-2019','DD/MM/YY')
old 1: insert into notes values(&notes_id,&subject_id,&course_id,&'&notes_write',&dateofclass)
new 1: insert into notes values(502,402,732,'It has been address about',TO_DATE('22-07-2019','DD/MM/YY'))

1 row created.

SQL> /
Enter value for notes_id: 503
Enter value for subject_id: 403
Enter value for course_id: 733
Enter value for notes_write: There is a discussion on
Enter value for dateofclass: TO_DATE('12-06-2019','DD/MM/YY')
old 1: insert into notes values(&notes_id,&subject_id,&course_id,&'&notes_write',&dateofclass)
new 1: insert into notes values(503,403,733,'There is a discussion on',TO_DATE('12-06-2019','DD/MM/YY'))

1 row created.

SQL> /
Enter value for notes_id: 504
Enter value for subject_id: 404
Enter value for course_id: 734
Enter value for notes_write: To know about
Enter value for dateofclass: TO_DATE('12-04-2020','DD/MM/YY')
old 1: insert into notes values(&notes_id,&subject_id,&course_id,&'&notes_write',&dateofclass)
new 1: insert into notes values(504,404,734,'To know about',TO_DATE('12-04-2020','DD/MM/YY'))

1 row created.
```

ALL TABLES ENTRIES/ROWS:

TITLE: STUDENT NOTES MANAGEMENT SYSTEM

```
SQL> select * from college;
```

COLLE	COLLEGE_NAME
101	vasavi
102	vjit
103	jbit
104	gsit
105	cbit
106	cmr

COLLEGE_ADDRESS
ibrahimbagh
miyapur
himampur
uppal
sherling
wargal

6 rows selected.

```
SQL> select * from student;
```

STUDENT_ID	STUDENT_NAME	STUDENT_CO
86	Amar	8272838811
88	Deepu	9273782111
82	Weesa	8428192467
75	Joshua	9639628367
56	Prasad	7538290222

STUDENT_ADDRESS	COLLE
Guntur	101
Karimnagar	102
Siddipet	103
Hyderabad	104
Nizambad	105

```
SQL> select * from subject;
```

SUBJECT_ID	SUBJECT_NAME	TEACHER_CO	TEACHER_NAME
401	Physics	7292887654	Iqbal
402	Maths	9428341782	Imam
403	DBMS	7258279812	Indra
404	PS	7367215267	Esha
405	HumanValues	9364262987	Tarun

```
SQL> select * from course;
```

COURSE_ID	COURSE_NAME	SUBJECT_ID	COURSE_CREDITS	ENROLLMENT_YEAR
731	BE	401	4	2018
12-NOV-18	22-MAY-19	101	86	
732	ME	402	5	2017
20-JUN-17	20-DEC-17	102	88	
733	BCA	403	7	2019
12-JAN-19	11-NOV-19	103	82	

COURSE_ID	COURSE_NAME	SUBJECT_ID	COURSE_CREDITS	ENROLLMENT_YEAR
734	MCA	404	4	2020
11-FEB-20	21-AUG-20	104	75	
735	BBA	405	6	2018
21-MAR-18	07-NOV-18	105	56	

```
SQL> select * from notes;
```

NOTES_ID	SUBJECT_ID	COURSE_ID	DATEOFCLA
501	401	731	11-OCT-18
Today there has been a discussion on			
502	402	732	11-OCT-19
It has been notes			
503	403	733	11-JAN-19
One of the important			
504	404	734	01-JAN-19
There is a clear			
505	405	735	02-NOV-19
We understood the			
506	401	731	29-JUL-19
abc			

6 rows selected.

Implementation:

Front end programs and its connectivity

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
    {
        connection
        =
        DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","it19737086","vasavi");
        statement = connection.createStatement();

    }
    catch (SQLException connectException)
    {
        System.out.println(connectException.getMessage())
        ;
        System.out.println(connectException.getSQLState()
        );
        System.out.println(connectException.getErrorCode());
        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.

Program:

Table: College

```
package SNMS;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.sql.*;
import java.util.Properties;

public class college
{
    private JPanel pn,pn1,pn2,pn3;
    private JFrame jframe;
    private JButton JB_insert,JB_modify,JB_view,JB_delete;
    private JLabel JL_college_code,JL_college_name,JL_college_address;
    private JTextField JTF_college_code,JTF_college_name,JTF_college_address;
    Connection con;
    ResultSet rs;
    Statement stmt;
    private JMenuItem insert1,update1,view1,delete1;
    private List collegelist;

    public college(JPanel pn,JFrame jframe,JMenuItem insert1,JMenuItem
update1,JMenuItem view1,JMenuItem delete1)
    {
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectDatabase();

        this.jframe=jframe;
        this.insert1=insert1;
        this.update1=update1;
        this.view1=view1;
        this.delete1=delete1;

        JL_college_code=new JLabel("college Code:");
        JTF_college_code=new JTextField(10);
        JL_college_name=new JLabel("college Name:");
        JTF_college_name=new JTextField(10);
    }
}
```

```

        JL_college_address=new JLabel("college Address:");
        JTF_college_address=new JTextField(10);

        this.pn=pn;
    }
    public void connectDatabase()
    {
        try
        {
            Connection con=DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe","it19737086","vasavi");

            stmt=con.createStatement();
            stmt.executeUpdate("commit");

        }
        catch (SQLException connectException)
        {
            System.out.println(connectException.getMessage());
            System.out.println(connectException.getSQLState());
            System.out.println(connectException.getErrorCode());
            System.exit(1);
        }
    }

    private void displaySQLErrors(SQLException e)
    {
        JOptionPane.showMessageDialog(pn1,"\nSQLException: " + e.getMessage() +
            "\n"+"SQLState:      " + e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() +
            "\n");
    }
    public void loadcollege()
    {
        try
        {
            collegeList=new List();
            collegeList.removeAll();
            rs=stmt.executeQuery("select * from college");
            while(rs.next())
            {
                collegeList.add(rs.getString("college_code"));
            }
        }
        catch(SQLException e)
        {
            displaySQLErrors(e);
        }
    }
}

```



```

public void buildGUI()
{
    insert1.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent aevt)
        {
            JB_insert=new JButton("Submit");

            JTF_college_code.setText(null);
            JTF_college_name.setText(null);
            JTF_college_address.setText(null);

            loadcollege();

            pn.removeAll();
            jframe.invalidate();
            jframe.validate();
            jframe.repaint();

            pn1=new JPanel();
            pn1.setLayout(new GridLayout(10,10));
            pn1.add(JL_college_code);
            pn1.add(JTF_college_code);
            pn1.add(JL_college_name);
            pn1.add(JTF_college_name);
            pn1.add(JL_college_address);
            pn1.add(JTF_college_address);

            pn3=new JPanel(new FlowLayout());
            pn3.add(JB_insert);
            pn1.setBounds(115,80,300,250);
            pn3.setBounds(200,350,75,35);

            pn2=new JPanel(new FlowLayout());
            collegeList=new List(10);
            loadcollege();
            pn2.add(collegeList);
            pn2.setBounds(200,350,300,180);

            pn.add(pn1);
            pn.add(pn3);
            pn.add(pn2);

            pn.setLayout(new BorderLayout());
            jframe.add(pn);
            jframe.setSize(800,800);
            jframe.validate();

            JB_insert.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent aevt)

```

```

        {
            try
            {
                String query= "INSERT INTO college
VALUES(" + JTF_college_code.getText() + ","
+ ""'+JTF_college_name.getText() + "','
+ ""'+JTF_college_address.getText() + "',')";
                int i = stmt.executeUpdate(query);

                JOptionPane.showMessageDialog(pn, "\nInserted "+i+" rows successfully");
                loadcollege();
                System.out.println("Done");
            }
            catch(SQLException e)
            {
                displaySQLErrors(e);
            }
        }
    });
}

});
update1.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JB_modify=new JButton("Modify");

        JTF_college_code.setText(null);
        JTF_college_name.setText(null);
        JTF_college_address.setText(null);

        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        pn1=new JPanel();
        pn1.setLayout(new GridLayout(10,10));
        pn1.add(JL_college_code);
        pn1.add(JTF_college_code);
        pn1.add(JL_college_name);
        pn1.add(JTF_college_name);
        pn1.add(JL_college_address);
        pn1.add(JTF_college_address);

        pn3=new JPanel(new FlowLayout());
        pn3.add(JB_modify);
        pn1.setBounds(115,80,300,250);
        pn3.setBounds(200,350,75,35);

        pn2=new JPanel(new FlowLayout());
        collegeList=new List(10);
        loadcollege();
        pn2.add(collegeList);
    }
});

```

```

pn2.setBounds(200,350,300,180);

pn.add(pn1);
pn.add(pn3);
pn.add(pn2);

pn.setLayout(new BorderLayout());
jframe.add(pn);
jframe.setSize(800,800);
jframe.validate();

collegelist.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent ievt)
    {
        try
        {
            rs=stmt.executeQuery("select * from
college");

            while (rs.next())
            {
                if
(rs.getString("college_code").equals(collegelist.getSelectedItem()))
                    break;
            }
            if (!rs.isAfterLast())
            {

JTF_college_code.setText(rs.getString("college_code"));

JTF_college_name.setText(rs.getString("college_name"));

JTF_college_address.setText(rs.getString("college_address"));

            }
        }
        catch (SQLException selectException)
        {
            displaySQLErrors(selectException);
        }
    }
});
JB_modify.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        try
        {
            int
a=JOptionPane.showConfirmDialog(pn,"Are you sure want to update:");
            if(a==JOptionPane.YES_OPTION)
            {
                String
pack=JOptionPane.showInputDialog(pn,"Enter New college Name:");
JTF_college_name.setText(pack);

```

```

        String query="update college set
college_name='"+pack+"' where college_code="+JTF_college_code.getText();
        int i=stmt.executeUpdate(query);

JOptionPane.showMessageDialog(pn,"\nUpdated "+i+" rows succesfully");
        loadcollege();
    }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
    }
    });
}
});
delete1.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JB_delete=new JButton("Delete");

        JTF_college_code.setText(null);
        JTF_college_name.setText(null);
        JTF_college_address.setText(null);

        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        pn1=new JPanel();
        pn1.setLayout(new GridLayout(10,10));
        pn1.add(JL_college_code);
        pn1.add(JTF_college_code);
        pn1.add(JL_college_name);
        pn1.add(JTF_college_name);
        pn1.add(JL_college_address);
        pn1.add(JTF_college_address);

        pn3=new JPanel(new FlowLayout());
        pn3.add(JB_delete);
        pn1.setBounds(115,80,300,250);
        pn3.setBounds(200,350,75,35);

        pn2=new JPanel(new FlowLayout());
        collegelist=new List(10);
        loadcollege();
        pn2.add(collegelist);
        pn2.setBounds(200,350,300,200);

        pn.add(pn1);
        pn.add(pn3);
        pn.add(pn2);

```

```

        pn.setLayout(new BorderLayout());
        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        collegeList.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent ievt)
            {
                try
                {
                    rs=stmt.executeQuery("select * from
college");

                    while (rs.next())
                    {
                        if
(rs.getString("college_code").equals(collegeList.getSelectedItem()))
                            break;
                    }
                    if (!rs.isAfterLast())
                    {

                        JTF_college_code.setText(rs.getString("college_code"));
                        JTF_college_name.setText(rs.getString("college_name"));
                        JTF_college_address.setText(rs.getString("college_address"));

                    }
                }
                catch (SQLException selectException)
                {
                    displaySQLErrors(selectException);
                }
            }
        });
        JB_delete.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent aevt)
            {
                try
                {
                    int
a=JOptionPane.showConfirmDialog(pn,"Are you sure want to Delete:");
                    if(a==JOptionPane.YES_OPTION)
                    {
                        //String query="DELETE FROM
college WHERE college_code="+collegeList.getSelectedItem();
                        String query="DELETE FROM college
WHERE college_code="+JTF_college_code.getText();
                        int i=stmt.executeUpdate(query);

                        JOptionPane.showMessageDialog(pn,"\nDeleted "+i+" rows succesfully");
                        loadcollege();
                    }
                }
            }
        });
    }
}

```

```

        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
}
});
}
});
view1.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        JLabel view=new JLabel("college View");
        JB_view=new JButton("View");
        Font myFont = new Font("Serif",Font.BOLD,50);
        view.setFont((myFont));

        pn1=new JPanel();
        pn2=new JPanel();
        pn1.add(view);
        pn2.add(JB_view);
        pn.add(pn1);
        pn.add(pn2);
        pn.setBounds(500,800,300,300);
        pn.setLayout(new FlowLayout());

        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        JB_view.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent aevt)
            {
                JFrame jf=new JFrame("college Details");
                JTable jt;
                DefaultTableModel model = new
DefaultTableModel();

                jt = new JTable(model);
                model.addColumn("college_code");
                model.addColumn("college name");
                model.addColumn("college address");
                try
                {
                    rs=stmt.executeQuery("select * from
college");

                    while(rs.next())

```

```

        {
            model.addRow(new
Object[] {rs.getInt(1),
rs.getString(2),rs.getString(3)});
        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
    jt.setEnabled(false);
    jt.setBounds(30, 40, 300, 300);
    JScrollPane jsp = new JScrollPane(jt);
    jf.add(jsp);
    jf.setSize(800, 400);
    jf.setVisible(true);
}
    });
}
});
}
}

```

Table: Student-

```

package SNMS;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.sql.*;
import java.util.Properties;

public class student
{
    private JPanel pn,pn1,pn2,pn3;
    private JFrame jframe;
    private JButton JB_insert,JB_modify,JB_view,JB_delete;
    private JLabel
JL_student_id,JL_student_name,JL_student_contact,JL_student_address,JL_college_code;
    private JTextField
JTF_student_id,JTF_student_name,JTF_student_contact,JTF_student_address,JTF_college_c
ode;

    Connection con;
    ResultSet rs;
    Statement stmt;
    private JMenuItem insert2,update2,view2,delete2;
    private List studentList;
    private Choice collegeCode;
}

```

```

    public student(JPanel pn,JFrame jframe,JMenuItem insert2,JMenuItem
update2,JMenuItem view2,JMenuItem delete2)
    {
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectDatabase();

        this.jframe=jframe;
        this.insert2=insert2;
        this.update2=update2;
        this.view2=view2;
        this.delete2=delete2;

        JL_student_id=new JLabel("student_id:");
        JTF_student_id=new JTextField(10);
        JL_student_name=new JLabel("student Name:");
        JTF_student_name=new JTextField(10);
        JL_student_contact=new JLabel("student Contact:");
        JTF_student_contact=new JTextField(10);
        JL_student_address=new JLabel("student Address:");
        JTF_student_address=new JTextField(10);
        JL_college_code=new JLabel("college Code:");
        collegeCode=new Choice();
        JTF_college_code=new JTextField(10);

        this.pn=pn;
    }
    public void connectDatabase()
    {
        try
        {
            Connection con=DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe","it19737086","vasavi");

            stmt=con.createStatement();
            stmt.executeUpdate("commit");

        }
        catch (SQLException connectException)
        {
            System.out.println(connectException.getMessage());
            System.out.println(connectException.getSQLState());
            System.out.println(connectException.getErrorCode());
            System.exit(1);
        }
    }
}

```



```

        private void displaySQLErrors(SQLException e)
        {
            JOptionPane.showMessageDialog(pn1, "\nSQLException: " + e.getMessage() +
"\n"+"SQLState:      " + e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() +
"\n");
        }
        public void loadcollege()
        {
            try
            {
                //collegeCode=new Choice();
                collegeCode.removeAll();
                rs=stmt.executeQuery("select * from college");
                while(rs.next())
                {
                    collegeCode.add(rs.getString("college_code"));
                }
            }
            catch(SQLException e)
            {
                displaySQLErrors(e);
            }
        }
        public void loadstudent()
        {
            try
            {
                studentList=new List();
                studentList.removeAll();
                rs=stmt.executeQuery("select * from student");
                while(rs.next())
                {
                    studentList.add(rs.getString("student_id"));
                }
            }
            catch(SQLException e)
            {
                displaySQLErrors(e);
            }
        }
        public void buildGUI()
        {
            insert2.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent aevt)
                {
                    JB_insert=new JButton("Submit");
                    loadcollege();

                    JTF_student_id.setText(null);
                    JTF_student_name.setText(null);
                    JTF_student_contact.setText(null);
                    JTF_student_address.setText(null);
                }
            });
        }
    }

```

```

//JTF_college_code.setText(null);

loadstudent();

pn.removeAll();
jframe.invalidate();
jframe.validate();
jframe.repaint();

pn1=new JPanel();
pn1.setLayout(new GridLayout(10,10));
pn1.add(JL_student_id);
pn1.add(JTF_student_id);
pn1.add(JL_student_name);
pn1.add(JTF_student_name);
pn1.add(JL_student_contact);
pn1.add(JTF_student_contact);
pn1.add(JL_student_address);
pn1.add(JTF_student_address);
pn1.add(JL_college_code);
pn1.add(collegeCode);

pn3=new JPanel(new FlowLayout());
pn3.add(JB_insert);
pn1.setBounds(115,80,300,250);
pn3.setBounds(200,350,75,35);

pn2=new JPanel(new FlowLayout());
studentList=new List(10);
loadstudent();
pn2.add(studentList);
pn2.setBounds(200,350,300,180);

pn.add(pn1);
pn.add(pn3);
pn.add(pn2);

pn.setLayout(new BorderLayout());
jframe.add(pn);
jframe.setSize(800,800);
jframe.validate();

JB_insert.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        try
        {
            String query= "INSERT INTO student
VALUES(" + JTF_student_id.getText() + ","
            + "" +JTF_student_name.getText() + ","
            + ""+JTF_student_contact.getText() + "," + ""+JTF_student_address.getText()
            + ","+""+collegeCode.getSelectedItem()+"");

```

```

        int i = stmt.executeUpdate(query);

JOptionPane.showMessageDialog(pn, "\nInserted "+i+" rows successfully");
        loadstudent();
        System.out.println("Done");
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
    }
    });
}
});
update2.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JB_modify=new JButton("Modify");

        JTF_student_id.setText(null);
        JTF_student_name.setText(null);
        JTF_student_contact.setText(null);
        JTF_student_address.setText(null);
        JTF_college_code.setText(null);

        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        pn1=new JPanel();
        pn1.setLayout(new GridLayout(10,10));
        pn1.add(JL_student_id);
        pn1.add(JTF_student_id);
        pn1.add(JL_student_name);
        pn1.add(JTF_student_name);
        pn1.add(JL_student_contact);
        pn1.add(JTF_student_contact);
        pn1.add(JL_student_address);
        pn1.add(JTF_student_address);
        pn1.add(JL_college_code);
        pn1.add(JTF_college_code);

        pn3=new JPanel(new FlowLayout());
        pn3.add(JB_modify);
        pn1.setBounds(115,80,300,250);
        pn3.setBounds(200,350,75,35);

        pn2=new JPanel(new FlowLayout());
        studentList=new List(10);
        loadstudent();
        pn2.add(studentList);
        pn2.setBounds(200,350,300,180);

```

```

        pn.add(pn1);
        pn.add(pn3);
        pn.add(pn2);

        pn.setLayout(new BorderLayout());
        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        studentList.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent ievt)
            {
                try
                {
                    rs=stmt.executeQuery("select * from
student");

                    while (rs.next())
                    {
                        if
(rs.getString("student_id").equals(studentList.getSelectedItem()))
                            break;
                    }
                    if (!rs.isAfterLast())
                    {

                        JTF_student_id.setText(rs.getString("student_id"));

                        JTF_student_name.setText(rs.getString("student_name"));

                        JTF_student_contact.setText(rs.getString("student_contact"));

                        JTF_student_address.setText(rs.getString("student_address"));

                        JTF_college_code.setText(rs.getString("college_code"));

                    }
                }
                catch (SQLException selectException)
                {
                    displaySQLErrors(selectException);
                }
            }
        });
        JB_modify.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent aevt)
            {
                try
                {
                    int
a=JOptionPane.showConfirmDialog(pn,"Are you sure want to update:");
                    if(a==JOptionPane.YES_OPTION)
                    {

```

```

        String
pack=JOptionPane.showInputDialog(pn,"Enter New student Name:");
        JTF_student_name.setText(pack);
        String query="update student set
student_name='"+pack+"' where student_id="+JTF_student_id.getText();
        int i=stmt.executeUpdate(query);

        JOptionPane.showMessageDialog(pn,"\nUpdated "+i+" rows succesfully");
        loadstudent();
    }
}
catch(SQLException e)
{
    displaySQLErrors(e);
}
});
}
});
delete2.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JB_delete=new JButton("Delete");

        JTF_student_id.setText(null);
        JTF_student_name.setText(null);
        JTF_student_contact.setText(null);
        JTF_student_address.setText(null);
        JTF_college_code.setText(null);

        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        pn1=new JPanel();
        pn1.setLayout(new GridLayout(10,10));
        pn1.add(JL_student_id);
        pn1.add(JTF_student_id);
        pn1.add(JL_student_name);
        pn1.add(JTF_student_name);
        pn1.add(JL_student_contact);
        pn1.add(JTF_student_contact);
        pn1.add(JL_student_address);
        pn1.add(JTF_student_address);
        pn1.add(JL_college_code);
        pn1.add(JTF_college_code);

        pn3=new JPanel(new FlowLayout());
        pn3.add(JB_delete);
        pn1.setBounds(115,80,300,250);
        pn3.setBounds(200,350,75,35);

```

```

        pn2=new JPanel(new FlowLayout());
        studentList=new List(10);
        loadstudent();
        pn2.add(studentList);
        pn2.setBounds(200,350,300,200);

        pn.add(pn1);
        pn.add(pn3);
        pn.add(pn2);

        pn.setLayout(new BorderLayout());
        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        studentList.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent ievt)
            {
                try
                {
                    rs=stmt.executeQuery("select * from
student");

                    while (rs.next())
                    {
                        if
(rs.getString("student_id").equals(studentList.getSelectedItem()))
                            break;
                    }
                    if (!rs.isAfterLast())
                    {

JTF_student_id.setText(rs.getString("student_id"));

JTF_student_name.setText(rs.getString("student_name"));

JTF_student_contact.setText(rs.getString("student_contact"));

JTF_student_address.setText(rs.getString("student_address"));

JTF_college_code.setText(rs.getString("college_code"));

                    }
                }
                catch (SQLException selectException)
                {
                    displaySQLErrors(selectException);
                }
            }
        });
        JB_delete.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent aevt)
            {

```

```

        try
        {
            int
a=JOptionPane.showConfirmDialog(pn,"Are you sure want to Delete:");
            if(a==JOptionPane.YES_OPTION)
            {
                //String query="DELETE FROM
student WHERE student_id="+studentList.getSelectedItem();
                String query="DELETE FROM student
WHERE student_id="+JTF_student_id.getText();
                int i=stmt.executeUpdate(query);

                JOptionPane.showMessageDialog(pn,"\nDeleted "+i+" rows succesfully");
                loadstudent();
            }
        }
        catch(SQLException e)
        {
            displaySQLErrors(e);
        }
    }
});
}
});
view2.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        JLabel view=new JLabel("student View");
        JB_view=new JButton("View");
        Font myFont = new Font("Serif",Font.BOLD,50);
        view.setFont(myFont);

        pn1=new JPanel();
        pn2=new JPanel();
        pn1.add(view);
        pn2.add(JB_view);
        pn.add(pn1);
        pn.add(pn2);
        pn.setBounds(500,800,300,300);
        pn.setLayout(new FlowLayout());

        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        JB_view.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent aevt)

```

```

        {
            JFrame jf=new JFrame("student Details");
            JTable jt;
            DefaultTableModel model = new
DefaultTableModel();

            jt = new JTable(model);
            model.addColumn("student_id");
            model.addColumn("student name");
            model.addColumn("student contact");
            model.addColumn("student address");
            model.addColumn("college Code");
            try
            {
                rs=stmt.executeQuery("select * from
student");

                while(rs.next())
                {
                    model.addRow(new
Object[]{rs.getString("student_id"),
rs.getString("student_name"),rs.getString("student_contact"),rs.getString("stud
ent_address")
,rs.getString("college_code")});
                }
            }
            catch(SQLException e)
            {
                displaySQLErrors(e);
            }
            jt.setEnabled(false);
            jt.setBounds(30, 40, 300, 300);
            JScrollPane jsp = new JScrollPane(jt);
            jf.add(jsp);
            jf.setSize(800, 400);
            jf.setVisible(true);
        }
    });
}
}
}

```


TABLE: SUBJECT-

```

package SNMS;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.sql.*;
import java.util.Properties;

public class subject
{
    private JPanel pn,pn1,pn2,pn3;
    private JFrame jframe;
    private JButton JB_insert,JB_modify,JB_view,JB_delete;
    private JLabel
    JL_subject_id,JL_subject_name,JL_teacher_contact,JL_teacher_name;
    private JTextField
    JTF_subject_id,JTF_subject_name,JTF_teacher_contact,JTF_teacher_name;
    Connection con;
    ResultSet rs;
    Statement stmt;
    private JMenuItem insert3,update3,view3,delete3;
    private List subjectList;

    public subject(JPanel pn,JFrame jframe,JMenuItem insert3,JMenuItem
    update3,JMenuItem view3,JMenuItem delete3)
    {
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectDatabase();

        this.jframe=jframe;
        this.insert3=insert3;
        this.update3=update3;
        this.view3=view3;
        this.delete3=delete3;

        JL_subject_id=new JLabel("subject_id");
        JTF_subject_id=new JTextField(10);
        JL_subject_name=new JLabel("subject Name:");
        JTF_subject_name=new JTextField(10);
    }

```

```

        JL_teacher_contact=new JLabel("teacher contact:");
        JTF_teacher_contact=new JTextField(10);
        JL_teacher_name=new JLabel("teacher name:");
        JTF_teacher_name=new JTextField(10);

        this.pn=pn;
    }
    public void connectDatabase()
    {
        try
        {
            Connection con=DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe","it19737086","vasavi");

            stmt=con.createStatement();
            stmt.executeUpdate("commit");

        }
        catch (SQLException connectException)
        {
            System.out.println(connectException.getMessage());
            System.out.println(connectException.getSQLState());
            System.out.println(connectException.getErrorCode());
            System.exit(1);
        }
    }
    private void displaySQLErrors(SQLException e)
    {
        JOptionPane.showMessageDialog(pn1,"\nSQLException: " + e.getMessage() +
            "\n"+"SQLState:      " + e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() +
            "\n");
    }
    public void loadsubject()
    {
        try
        {
            subjectList=new List();
            subjectList.removeAll();
            rs=stmt.executeQuery("select * from subject");
            while(rs.next())
            {
                subjectList.add(rs.getString("subject_id"));
            }
        }
        catch(SQLException e)
        {
            displaySQLErrors(e);
        }
    }
    public void buildGUI()
    {
        insert3.addActionListener(new ActionListener() {

```

```

@Override
public void actionPerformed(ActionEvent aevt)
{
    JB_insert=new JButton("Submit");

    JTF_subject_id.setText(null);
    JTF_subject_name.setText(null);
    JTF_teacher_contact.setText(null);
    JTF_teacher_name.setText(null);

    loadsubject();

    pn.removeAll();
    jframe.invalidate();
    jframe.validate();
    jframe.repaint();

    pn1=new JPanel();
    pn1.setLayout(new GridLayout(10,10));
    pn1.add(JL_subject_id);
    pn1.add(JTF_subject_id);
    pn1.add(JL_subject_name);
    pn1.add(JTF_subject_name);
    pn1.add(JL_teacher_contact);
    pn1.add(JTF_teacher_contact);
    pn1.add(JL_teacher_name);
    pn1.add(JTF_teacher_name);

    pn3=new JPanel(new FlowLayout());
    pn3.add(JB_insert);
    pn1.setBounds(115,80,300,250);
    pn3.setBounds(200,350,75,35);

    pn2=new JPanel(new FlowLayout());
    subjectList=new List(10);
    loadsubject();
    pn2.add(subjectList);
    pn2.setBounds(200,350,300,180);

    pn.add(pn1);
    pn.add(pn3);
    pn.add(pn2);

    pn.setLayout(new BorderLayout());
    jframe.add(pn);
    jframe.setSize(800,800);
    jframe.validate();

    JB_insert.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent aevt)
        {
            try

```

```

        {
            String query= "INSERT INTO subject
VALUES(" + JTF_subject_id.getText() + ","
            + "'" +JTF_subject_name.getText() + "'",
            + "'" +JTF_teacher_contact.getText() + "'", " + "'" +JTF_teacher_name.getText() + "'"");
            int i = stmt.executeUpdate(query);

            JOptionPane.showMessageDialog(pn, "\nInserted "+i+" rows successfully");
            loadsubject();
            System.out.println("Done");
        }
        catch(SQLException e)
        {
            displaySQLErrors(e);
        }
    }
});
});
update3.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JB_modify=new JButton("Modify");

        JTF_subject_id.setText(null);
        JTF_subject_name.setText(null);
        JTF_teacher_contact.setText(null);
        JTF_teacher_name.setText(null);

        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        pn1=new JPanel();
        pn1.setLayout(new GridLayout(10,10));
        pn1.add(JL_subject_id);
        pn1.add(JTF_subject_id);
        pn1.add(JL_subject_name);
        pn1.add(JTF_subject_name);
        pn1.add(JL_teacher_contact);
        pn1.add(JTF_teacher_contact);
        pn1.add(JL_teacher_name);
        pn1.add(JTF_teacher_name);

        pn3=new JPanel(new FlowLayout());
        pn3.add(JB_modify);
        pn1.setBounds(115,80,300,250);
        pn3.setBounds(200,350,75,35);

        pn2=new JPanel(new FlowLayout());
        subjectList=new List(10);
        loadsubject();

```

```

        pn2.add(subjectList);
        pn2.setBounds(200,350,300,180);

        pn.add(pn1);
        pn.add(pn3);
        pn.add(pn2);

        pn.setLayout(new BorderLayout());
        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        subjectList.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent ievt)
            {
                try
                {
                    rs=stmt.executeQuery("select * from
subject");

                    while (rs.next())
                    {
                        if
(rs.getString("subject_id").equals(subjectList.getSelectedItem()))
                            break;
                    }
                    if (!rs.isAfterLast())
                    {

JTF_subject_id.setText(rs.getString("subject_id"));

JTF_subject_name.setText(rs.getString("subject_name"));

JTF_teacher_contact.setText(rs.getString("teacher_contact"));

JTF_teacher_name.setText(rs.getString("teacher_name"));

                    }
                }
                catch (SQLException selectException)
                {
                    displaySQLErrors(selectException);
                }
            }
        });
        JB_modify.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent aevt)
            {
                try
                {
                    int
a=JOptionPane.showConfirmDialog(pn,"Are you sure want to update:");
                    if(a==JOptionPane.YES_OPTION)
                    {

```

```

        String
pack=JOptionPane.showInputDialog(pn,"Enter New subject Name:");
        JTF_subject_name.setText(pack);
        String query="update subject set
subject_name='"+pack+"' where subject_id="+JTF_subject_id.getText();
        int i=stmt.executeUpdate(query);

        JOptionPane.showMessageDialog(pn,"\nUpdated "+i+" rows succesfully");
        loadsubject();
    }
}
catch(SQLException e)
{
    displaySQLErrors(e);
}
});
}
});
delete3.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JB_delete=new JButton("Delete");

        JTF_subject_id.setText(null);
        JTF_subject_name.setText(null);
        JTF_teacher_contact.setText(null);
        JTF_teacher_name.setText(null);

        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        pn1=new JPanel();
        pn1.setLayout(new GridLayout(10,10));
        pn1.add(JL_subject_id);
        pn1.add(JTF_subject_id);
        pn1.add(JL_subject_name);
        pn1.add(JTF_subject_name);
        pn1.add(JL_teacher_contact);
        pn1.add(JTF_teacher_contact);
        pn1.add(JL_teacher_name);
        pn1.add(JTF_teacher_name);

        pn3=new JPanel(new FlowLayout());
        pn3.add(JB_delete);
        pn1.setBounds(115,80,300,250);
        pn3.setBounds(200,350,75,35);

        pn2=new JPanel(new FlowLayout());
        subjectList=new List(10);
        loadsubject();
    }
});

```

```

        pn2.add(subjectList);
        pn2.setBounds(200,350,300,200);

        pn.add(pn1);
        pn.add(pn3);
        pn.add(pn2);

        pn.setLayout(new BorderLayout());
        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        subjectList.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent ievt)
            {
                try
                {
                    rs=stmt.executeQuery("select * from
subject");

                    while (rs.next())
                    {
                        if
(rs.getString("subject_id").equals(subjectList.getSelectedItem()))
                            break;
                    }
                    if (!rs.isAfterLast())
                    {

JTF_subject_id.setText(rs.getString("subject_id"));

JTF_subject_name.setText(rs.getString("subject_name"));

JTF_teacher_contact.setText(rs.getString("teacher_contact"));

JTF_teacher_name.setText(rs.getString("teacher_name"));

                    }
                }
                catch (SQLException selectException)
                {
                    displaySQLErrors(selectException);
                }
            }
        });
        JB_delete.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent aevt)
            {
                try
                {
                    int
a=JOptionPane.showConfirmDialog(pn,"Are you sure want to Delete:");
                    if(a==JOptionPane.YES_OPTION)
                    {

```

```

//String query="DELETE FROM
subject WHERE subject_id="+subjectList.getSelectedItem();
String query="DELETE FROM subject
WHERE subject_id="+JTF_subject_id.getText();

int i=stmt.executeUpdate(query);

JOptionPane.showMessageDialog(pn,"\nDeleted "+i+" rows succesfully");
loadsubject();
    }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
    }
    });
    }
    });
view3.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        JLabel view=new JLabel("subject View");
        JB_view=new JButton("View");
        Font myFont = new Font("Serif",Font.BOLD,50);
        view.setFont((myFont));

        pn1=new JPanel();
        pn2=new JPanel();
        pn1.add(view);
        pn2.add(JB_view);
        pn.add(pn1);
        pn.add(pn2);
        pn.setBounds(500,800,300,300);
        pn.setLayout(new FlowLayout());

        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        JB_view.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent aevt)
            {
                JFrame jf=new JFrame("subject Details");
                JTable jt;
                DefaultTableModel model = new
DefaultTableModel();

                jt = new JTable(model);

```



```

        model.addColumn("subject_id");
        model.addColumn("subject name");
        model.addColumn("teacher contact");
        model.addColumn("teacher name");
        try
        {
            rs=stmt.executeQuery("select * from
subject");

            while(rs.next())
            {
                model.addRow(new
Object[]{rs.getInt(1),
rs.getString(2),rs.getString(3),rs.getString(4)});
            }
        }
        catch(SQLException e)
        {
            displaySQLErrors(e);
        }
        jt.setEnabled(false);
        jt.setBounds(30, 40, 300, 300);
        JScrollPane jsp = new JScrollPane(jt);
        jf.add(jsp);
        jf.setSize(800, 400);
        jf.setVisible(true);
    }
});
}
});
}
}

```

TABLE: COURSE-

```
package SNMS;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.sql.*;
import java.util.Properties;

public class course
{
    private JPanel pn,pn1,pn2,pn3;
    private JFrame jframe;
    private JButton JB_insert,JB_modify,JB_view,JB_delete;
    private JLabel
JL_course_id,JL_course_name,JL_subject_id,JL_course_credits,JL_enrollment_year,JL_cou
rse_start_date,JL_course_end_date,JL_college_code,JL_student_id;
    private JTextField
JTF_course_id,JTF_course_name,JTF_subject_id,JTF_course_credits,JTF_enrollment_year,J
TF_course_start_date,JTF_course_end_date,JTF_college_code,JTF_student_id;
```

```

Connection con;
ResultSet rs;
Statement stmt;
private JMenuItem insert4,update4,view4,delete4;
private List courselist;
private Choice subjectId,collegeCode,studentId;

public course(JPanel pn,JFrame jframe,JMenuItem insert4,JMenuItem
update4,JMenuItem view4,JMenuItem delete4)
{
    try
    {
        Class.forName("oracle.jdbc.driver.OracleDriver");
    }
    catch (Exception e)
    {
        System.err.println("Unable to find and load driver");
        System.exit(1);
    }
    connectDatabase();

    this.jframe=jframe;
    this.insert4=insert4;
    this.update4=update4;
    this.view4=view4;
    this.delete4=delete4;

    JL_course_id=new JLabel("course_id:");
    JTF_course_id=new JTextField(10);
    JL_course_name=new JLabel("course Name:");
    JTF_course_name=new JTextField(10);
    JL_subject_id=new JLabel("subject Id:");
    subjectId=new Choice();
    JTF_subject_id=new JTextField(10);
    JL_course_credits=new JLabel("course credits:");
    JTF_course_credits=new JTextField(10);
    JL_enrollment_year=new JLabel("Enrollment Year:");
    JTF_enrollment_year=new JTextField(10);
    JL_course_start_date=new JLabel("Start Date:");
    JTF_course_start_date=new JTextField(10);
    JL_course_end_date=new JLabel("End Date:");
    JTF_course_end_date=new JTextField(10);
    JL_college_code=new JLabel("college Code:");
    collegeCode=new Choice();
    JTF_college_code=new JTextField(10);
    JL_student_id=new JLabel("student Id:");
    studentId=new Choice();
    JTF_student_id=new JTextField(10);

    this.pn=pn;
}
public void connectDatabase()
{

```

```

        try
        {
            Connection con=DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe","it19737086","vasavi");

            stmt=con.createStatement();
            stmt.executeUpdate("commit");

        }
        catch (SQLException connectException)
        {
            System.out.println(connectException.getMessage());
            System.out.println(connectException.getSQLState());
            System.out.println(connectException.getErrorCode());
            System.exit(1);
        }
    }
    private void displaySQLErrors(SQLException e)
    {
        JOptionPane.showMessageDialog(pn1,"\nSQLException: " + e.getMessage() +
            "\n"+"SQLState:      " + e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() +
            "\n");
    }
    public void loadsubject()
    {
        try
        {
            //subjectId=new Choice();
            subjectId.removeAll();
            rs=stmt.executeQuery("select * from subject");
            while(rs.next())
            {
                subjectId.add(rs.getString("subject_id"));
            }
        }
        catch(SQLException e)
        {
            displaySQLErrors(e);
        }
    }
    public void loadcollege()
    {
        try
        {
            //collegeCode=new Choice();
            collegeCode.removeAll();
            rs=stmt.executeQuery("select * from college");
            while(rs.next())
            {
                collegeCode.add(rs.getString("college_code"));
            }
        }
        catch(SQLException e)
        {

```

```

        displaySQLErrors(e);
    }
}
public void loadstudent()
{
    try
    {
        //studentId=new Choice();
        studentId.removeAll();
        rs=stmt.executeQuery("select * from student");
        while(rs.next())
        {
            studentId.add(rs.getString("student_id"));
        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
}
public void loadcourse()
{
    try
    {
        courseList=new List();
        courseList.removeAll();
        rs=stmt.executeQuery("select * from course");
        while(rs.next())
        {
            courseList.add(rs.getString("course_id"));
        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
}
public void buildGUI()
{
    insert4.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent aevt)
        {
            JB_insert=new JButton("Submit");
            loadsubject();
            loadcollege();
            loadstudent();
            loadcourse();

            JTF_course_id.setText(null);
            JTF_course_name.setText(null);
            //JTF_subject_id.setText(null);
            JTF_course_credits.setText(null);
        }
    });
}

```

```

JTF_enrollment_year.setText(null);
JTF_course_start_date.setText(null);
JTF_course_end_date.setText(null);
//JTF_college_code.setText(null);
//JTF_student_id.setText(null);

loadcourse();

pn.removeAll();
jframe.invalidate();
jframe.validate();
jframe.repaint();

pn1=new JPanel();
pn1.setLayout(new GridLayout(10,10));
pn1.add(JL_course_id);
pn1.add(JTF_course_id);
pn1.add(JL_course_name);
pn1.add(JTF_course_name);
pn1.add(JL_subject_id);
pn1.add(subjectId);
pn1.add(JL_course_credits);
pn1.add(JTF_course_credits);
pn1.add(JL_enrollment_year);
pn1.add(JTF_enrollment_year);
pn1.add(JL_course_start_date);
pn1.add(JTF_course_start_date);
pn1.add(JL_course_end_date);
pn1.add(JTF_course_end_date);
pn1.add(JL_college_code);
pn1.add(collegeCode);
pn1.add(JL_student_id);
pn1.add(studentId);

pn3=new JPanel(new FlowLayout());
pn3.add(JB_insert);
pn1.setBounds(115,80,300,250);
pn3.setBounds(200,350,75,35);

pn2=new JPanel(new FlowLayout());
courseList=new List(10);
loadcourse();
pn2.add(courseList);
pn2.setBounds(200,350,300,180);

pn.add(pn1);
pn.add(pn3);
pn.add(pn2);

pn.setLayout(new BorderLayout());
jframe.add(pn);
jframe.setSize(800,800);
jframe.validate();

```

```

        JB_insert.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent aevt)
            {
                try
                {
                    String query= "INSERT INTO course
VALUES(" + JTF_course_id.getText() + ","
                    + "" +JTF_course_name.getText()
+ "','"+subjectId.getSelectedItem()+"',"+JTF_course_credits.getText()+","+JTF_enrol
lment_year.getText()+","+JTF_course_start_date.getText()+","+JTF_course_end_date.getT
ext()+","+""+collegeCode.getSelectedItem()+"',"+""+studentId.getSelectedItem()+"')";
                    ;

                    int i = stmt.executeUpdate(query);

                    JOptionPane.showMessageDialog(pn,"\nInserted "+i+" rows successfully");
                    loadcourse();
                    System.out.println("Done");
                }
                catch(SQLException e)
                {
                    displaySQLErrors(e);
                }
            }
        });
    }
});
update4.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JB_modify=new JButton("Modify");

        JTF_course_id.setText(null);
        JTF_course_name.setText(null);
        JTF_subject_id.setText(null);
        JTF_course_credits.setText(null);
        JTF_enrollment_year.setText(null);
        JTF_course_start_date.setText(null);
        JTF_course_end_date.setText(null);
        JTF_college_code.setText(null);
        JTF_student_id.setText(null);

        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        pn1=new JPanel();
        pn1.setLayout(new GridLayout(10,10));
        pn1.add(JL_course_id);
        pn1.add(JTF_course_id);
        pn1.add(JL_course_name);

```

```

pn1.add(JTF_course_name);
pn1.add(JL_subject_id);
pn1.add(JTF_subject_id);
pn1.add(JL_course_credits);
pn1.add(JTF_course_credits);
pn1.add(JL_enrollment_year);
pn1.add(JTF_enrollment_year);
pn1.add(JL_course_start_date);
pn1.add(JTF_course_start_date);
pn1.add(JL_course_end_date);
pn1.add(JTF_course_end_date);
pn1.add(JL_college_code);
pn1.add(JTF_college_code);
pn1.add(JL_student_id);
pn1.add(JTF_student_id);

pn3=new JPanel(new FlowLayout());
pn3.add(JB_modify);
pn1.setBounds(115,80,300,250);
pn3.setBounds(200,350,75,35);

pn2=new JPanel(new FlowLayout());
courseList=new List(10);
loadcourse();
pn2.add(courseList);
pn2.setBounds(200,350,300,180);

pn.add(pn1);
pn.add(pn3);
pn.add(pn2);

pn.setLayout(new BorderLayout());
jframe.add(pn);
jframe.setSize(800,800);
jframe.validate();

courseList.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent ievt)
    {
        try
        {
            rs=stmt.executeQuery("select * from
course");

            while (rs.next())
            {
                if
(rs.getString("course_id").equals(courseList.getSelectedItem()))
                    break;
            }
            if (!rs.isAfterLast())
            {
                JTF_course_id.setText(rs.getString("course_id"));

```

```

JTF_course_name.setText(rs.getString("course_name"));
JTF_subject_id.setText(rs.getString("subject_id"));
JTF_course_credits.setText(rs.getString("course_credits"));
JTF_enrollment_year.setText(rs.getString("enrollment_year"));
JTF_course_start_date.setText(rs.getString("course_start_date"));
JTF_course_end_date.setText(rs.getString("course_end_date"));
JTF_college_code.setText(rs.getString("college_code"));
JTF_student_id.setText(rs.getString("student_id"));

        }
    }
    catch (SQLException selectException)
    {
        displaySQLErrors(selectException);
    }
}
});
JB_modify.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        try
        {
            int
a=JOptionPane.showConfirmDialog(pn,"Are you sure want to update:");
            if(a==JOptionPane.YES_OPTION)
            {
                String
pack=JOptionPane.showInputDialog(pn,"Enter New course name:");
                JTF_course_name.setText(pack);
                String query="update course set
course_name='"+pack+"' where course_id="+JTF_course_id.getText();
                int i=stmt.executeUpdate(query);

                JOptionPane.showMessageDialog(pn,"\nUpdated "+i+" rows succesfully");
                loadcourse();
            }
        }
        catch(SQLException e)
        {
            displaySQLErrors(e);
        }
    }
});
}
});
});

```



```

delete4.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JB_delete=new JButton("Delete");

        JTF_course_id.setText(null);
        JTF_course_name.setText(null);
        JTF_subject_id.setText(null);
        JTF_course_credits.setText(null);
        JTF_enrollment_year.setText(null);
        JTF_course_start_date.setText(null);
        JTF_course_end_date.setText(null);
        JTF_college_code.setText(null);
        JTF_student_id.setText(null);

        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        pn1=new JPanel();
        pn1.setLayout(new GridLayout(10,10));
        pn1.add(JL_course_id);
        pn1.add(JTF_course_id);
        pn1.add(JL_course_name);
        pn1.add(JTF_course_name);
        pn1.add(JL_subject_id);
        pn1.add(JTF_subject_id);
        pn1.add(JL_course_credits);
        pn1.add(JTF_course_credits);
        pn1.add(JL_enrollment_year);
        pn1.add(JTF_enrollment_year);
        pn1.add(JL_course_start_date);
        pn1.add(JTF_course_start_date);
        pn1.add(JL_course_end_date);
        pn1.add(JTF_course_end_date);
        pn1.add(JL_college_code);
        pn1.add(JTF_college_code);
        pn1.add(JL_student_id);
        pn1.add(JTF_student_id);

        pn3=new JPanel(new FlowLayout());
        pn3.add(JB_delete);
        pn1.setBounds(115,80,300,250);
        pn3.setBounds(200,350,75,35);

        pn2=new JPanel(new FlowLayout());
        courseList=new List(10);
        loadcourse();
        pn2.add(courseList);
        pn2.setBounds(200,350,300,200);
    }
});

```

```

        pn.add(pn1);
        pn.add(pn3);
        pn.add(pn2);

        pn.setLayout(new BorderLayout());
        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        courseList.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent ievt)
            {
                try
                {
                    rs=stmt.executeQuery("select * from
course");

                    while (rs.next())
                    {
                        if
(rs.getString("course_id").equals(courseList.getSelectedItem()))
                            break;
                    }
                    if (!rs.isAfterLast())
                    {

                        JTF_course_id.setText(rs.getString("course_id"));

                        JTF_course_name.setText(rs.getString("course_name"));

                        JTF_subject_id.setText(rs.getString("subject_id"));

                        JTF_course_credits.setText(rs.getString("course_credits"));

                        JTF_enrollment_year.setText(rs.getString("enrollment_year"));

                        JTF_course_start_date.setText(rs.getString("course_start_date"));

                        JTF_course_end_date.setText(rs.getString("course_end_date"));

                        JTF_college_code.setText(rs.getString("college_code"));

                        JTF_student_id.setText(rs.getString("student_id"));

                    }
                }
                catch (SQLException selectException)
                {
                    displaySQLErrors(selectException);
                }
            }
        });
        JB_delete.addActionListener(new ActionListener() {
            @Override

```

```

        public void actionPerformed(ActionEvent aevt)
        {
            try
            {
                int
a=JOptionPane.showConfirmDialog(pn,"Are you sure want to Delete:");
                if(a==JOptionPane.YES_OPTION)
                {
                    //String query="DELETE FROM
course WHERE course_id="+courseList.getSelectedItem();
                    String query="DELETE FROM course
WHERE course_id="+JTF_course_id.getText();
                    int i=stmt.executeUpdate(query);

JOptionPane.showMessageDialog(pn,"\nDeleted "+i+" rows succesfully");
                    loadcourse();
                }
            }
            catch(SQLException e)
            {
                displaySQLErrors(e);
            }
        }
    });
}
});
view4.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        JLabel view=new JLabel("course View");
        JB_view=new JButton("View");
        Font myFont = new Font("Serif",Font.BOLD,50);
        view.setFont((myFont));

        pn1=new JPanel();
        pn2=new JPanel();
        pn1.add(view);
        pn2.add(JB_view);
        pn.add(pn1);
        pn.add(pn2);
        pn.setBounds(500,800,300,300);
        pn.setLayout(new FlowLayout());

        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        JB_view.addActionListener(new ActionListener() {

```

```

@Override
public void actionPerformed(ActionEvent aevt)
{
    JFrame jf=new JFrame("course Details");
    JTable jt;
    DefaultTableModel model = new

DefaultTableModel();

    jt = new JTable(model);
    model.addColumn("course_id");
    model.addColumn("course name");
    model.addColumn("subject Id");
    model.addColumn("course credits");
    model.addColumn("Enrollment Year");
    model.addColumn("Course Start Date");
    model.addColumn("Course End Date");
    model.addColumn("college Code");
    model.addColumn("student Id");
    try
    {
        rs=stmt.executeQuery("select * from

course");

        while(rs.next())
        {
            model.addRow(new

Object[] {rs.getString("course_id"),

            rs.getString("course_name"),rs.getString("subject_id"),rs.getString("course_cre
dits"),rs.getString("enrollment_year")

            ,rs.getString("course_start_date"),rs.getString("course_end_date"),rs.getString
("college_code"),rs.getString("student_id")});
        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
    jt.setEnabled(false);
    jt.setBounds(30, 40, 300, 300);
    JScrollPane jsp = new JScrollPane(jt);
    jf.add(jsp);
    jf.setSize(800, 400);
    jf.setVisible(true);
}

});

}

});

}
}

```

TABLE: NOTES-

```

package SNMS;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.sql.*;
import java.util.Properties;

public class notes
{
    private JPanel pn,pn1,pn2,pn3;
    private JFrame jframe;
    private JButton JB_insert,JB_modify,JB_view,JB_delete;
    private JLabel
    JL_notes_id,JL_subject_id,JL_course_id,JL_notes_write,JL_dateofclass;
    private JTextField
    JTF_notes_id,JTF_subject_id,JTF_course_id,JTF_notes_write,JTF_dateofclass;
    Connection con;
    ResultSet rs;
    Statement stmt;
    private JMenuItem insert5,update5,view5,delete5;
    private List notesList;
    private Choice subjectId,courseId;
    public notes(JPanel pn,JFrame jframe,JMenuItem insert5,JMenuItem
    update5,JMenuItem view5,JMenuItem delete5)
    {
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectDatabase();

        this.jframe=jframe;
        this.insert5=insert5;
        this.update5=update5;
        this.view5=view5;
        this.delete5=delete5;

        JL_notes_id=new JLabel("notes_Id:");
        JTF_notes_id=new JTextField(10);
        JL_subject_id=new JLabel("subject Id:");
        JTF_subject_id=new JTextField(10);
        JL_course_id=new JLabel("course id:");
        JTF_course_id=new JTextField(10);
        JL_notes_write=new JLabel("notes write:");
        JTF_notes_write=new JTextField(10);
        JL_dateofclass=new JLabel("dateofclass:");
        JTF_dateofclass=new JTextField(10);
    }
}

```

```

        this.pn=pn;
    }
    public void connectDatabase()
    {
        try
        {
            Connection con=DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe","it19737086","vasavi");

            stmt=con.createStatement();
            stmt.executeUpdate("commit");

        }
        catch (SQLException connectException)
        {
            System.out.println(connectException.getMessage());
            System.out.println(connectException.getSQLState());
            System.out.println(connectException.getErrorCode());
            System.exit(1);
        }
    }
    private void displaySQLErrors(SQLException e)
    {
        JOptionPane.showMessageDialog(pn1,"\nSQLException: " + e.getMessage() +
            "\n"+"SQLState:      " + e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() +
            "\n");
    }
    public void loadsubject()
    {
        try
        {
            subjectId=new Choice();
            subjectId.removeAll();
            rs=stmt.executeQuery("select * from subject");
            while(rs.next())
            {
                subjectId.add(rs.getString("subject_id"));
            }
        }
        catch(SQLException e)
        {
            displaySQLErrors(e);
        }
    }
    public void loadcourse()
    {
        try
        {
            courseId=new Choice();
            courseId.removeAll();
            rs=stmt.executeQuery("select * from course");
            while(rs.next())
            {
                courseId.add(rs.getString("course_id"));
            }
        }
    }

```

```

        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
}
public void loadnotes()
{
    try
    {
        notesList=new List();
        notesList.removeAll();
        rs=stmt.executeQuery("select * from notes");
        while(rs.next())
        {
            notesList.add(rs.getString("notes_id"));
        }
    }
    catch(SQLException e)
    {
        displaySQLErrors(e);
    }
}
public void buildGUI()
{
    insert5.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent aevt)
        {
            JB_insert=new JButton("Submit");
            loadsubject();
            loadcourse();
            loadnotes();

            JTF_notes_id.setText(null);
            //JTF_subject_id.setText(null);
            JTF_course_id.setText(null);
            JTF_notes_write.setText(null);
            JTF_dateofclass.setText(null);

            loadnotes();

            pn.removeAll();
            jframe.invalidate();
            jframe.validate();
            jframe.repaint();

            pn1=new JPanel();
            pn1.setLayout(new GridLayout(10,10));
            pn1.add(JL_notes_id);
            pn1.add(JTF_notes_id);
            pn1.add(JL_subject_id);

```

```

        pn1.add(subjectId);
        pn1.add(JL_course_id);
        pn1.add(courseId);
        pn1.add(JL_notes_write);
        pn1.add(JTF_notes_write);
        pn1.add(JL_dateofclass);
        pn1.add(JTF_dateofclass);

        pn3=new JPanel(new FlowLayout());
        pn3.add(JB_insert);
        pn1.setBounds(115,80,300,250);
        pn3.setBounds(200,350,75,35);

        pn2=new JPanel(new FlowLayout());
        notesList=new List(10);
        loadnotes();
        pn2.add(notesList);
        pn2.setBounds(200,350,300,180);

        pn.add(pn1);
        pn.add(pn3);
        pn.add(pn2);

        pn.setLayout(new BorderLayout());
        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        JB_insert.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent aevt)
            {
                try
                {
                    String query= "INSERT INTO notes
VALUES(" + JTF_notes_id.getText() + ","
+"""+subjectId.getSelectedItem()+""',"+""""+courseId.getSelectedItem()+""',"+""""+JTF_not
es_write.getText() +""'," +JTF_dateofclass.getText() +")";
                    int i = stmt.executeUpdate(query);

                    JOptionPane.showMessageDialog(pn,"\nInserted "+i+" rows successfully");
                    loadnotes();
                    System.out.println("Done");
                }
                catch(SQLException e)
                {
                    displaySQLErrors(e);
                }
            }
        });
    }
});
});

```



```

update5.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JB_modify=new JButton("Modify");

        JTF_notes_id.setText(null);
        JTF_subject_id.setText(null);
        JTF_course_id.setText(null);
        JTF_notes_write.setText(null);
        JTF_dateofclass.setText(null);

        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        pn1=new JPanel();
        pn1.setLayout(new GridLayout(10,10));
        pn1.add(JL_notes_id);
        pn1.add(JTF_notes_id);
        pn1.add(JL_subject_id);
        pn1.add(JTF_subject_id);
        pn1.add(JL_course_id);
        pn1.add(JTF_course_id);
        pn1.add(JL_notes_write);
        pn1.add(JTF_notes_write);
        pn1.add(JL_dateofclass);
        pn1.add(JTF_dateofclass);

        pn3=new JPanel(new FlowLayout());
        pn3.add(JB_modify);
        pn1.setBounds(115,80,300,250);
        pn3.setBounds(200,350,75,35);

        pn2=new JPanel(new FlowLayout());
        notesList=new List(10);
        loadnotes();
        pn2.add(notesList);
        pn2.setBounds(200,350,300,180);

        pn.add(pn1);
        pn.add(pn3);
        pn.add(pn2);

        pn.setLayout(new BorderLayout());
        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        notesList.addItemListener(new ItemListener()
        {
            public void itemStateChanged(ItemEvent ievt)

```

```

        {
            try
            {
                rs=stmt.executeQuery("select * from
notes");

                while (rs.next())
                {
                    if
(rs.getString("notes_id").equals(notesList.getSelectedItem()))
                        break;
                }
                if (!rs.isAfterLast())
                {

JTF_notes_id.setText(rs.getString("notes_id"));

JTF_subject_id.setText(rs.getString("subject_id"));

JTF_course_id.setText(rs.getString("course_id"));

JTF_notes_write.setText(rs.getString("notes_write"));

JTF_dateofclass.setText(rs.getString("dateofclass"));
                }
            }
            catch (SQLException selectException)
            {
                displaySQLErrors(selectException);
            }
        }
    });
    JB_modify.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent aevt)
        {
            try
            {

                int
a=JOptionPane.showConfirmDialog(pn,"Are you sure do you want to update:");
                if(a==JOptionPane.YES_OPTION)
                {
                    String
pack=JOptionPane.showInputDialog(pn,"Enter New notes_write:");
                    JTF_notes_write.setText(pack);
                    String query="update notes set
notes_write='"+pack+"' where notes_id="+JTF_notes_id.getText();
                    int i=stmt.executeUpdate(query);

                    JOptionPane.showMessageDialog(pn,"\nUpdated "+i+" rows succesfully");
                    loadnotes();
                }
            }
            catch(SQLException e)

```

```

        {
            displaySQLErrors(e);
        }
    }
});
}
});
delete5.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        JB_delete=new JButton("Delete");

        JTF_notes_id.setText(null);
        JTF_subject_id.setText(null);
        JTF_course_id.setText(null);
        JTF_notes_write.setText(null);
        JTF_dateofclass.setText(null);

        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        pn1=new JPanel();
        pn1.setLayout(new GridLayout(10,10));
        pn1.add(JL_notes_id);
        pn1.add(JTF_notes_id);
        pn1.add(JL_subject_id);
        pn1.add(JTF_subject_id);
        pn1.add(JL_course_id);
        pn1.add(JTF_course_id);
        pn1.add(JL_notes_write);
        pn1.add(JTF_notes_write);
        pn1.add(JL_dateofclass);
        pn1.add(JTF_dateofclass);

        pn3=new JPanel(new FlowLayout());
        pn3.add(JB_delete);
        pn1.setBounds(115,80,300,250);
        pn3.setBounds(200,350,75,35);

        pn2=new JPanel(new FlowLayout());
        notesList=new List(10);
        loadnotes();
        pn2.add(notesList);
        pn2.setBounds(250,350,300,180);

        pn.add(pn1);
        pn.add(pn3);
        pn.add(pn2);

```

```

        pn.setLayout(new BorderLayout());
        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        notesList.addItemListener(new ItemListener()
        {
            public void itemStateChanged(ItemEvent ievt)
            {
                try
                {
                    rs=stmt.executeQuery("select *
from notes");

                    while (rs.next())
                    {
                        if
(rs.getString("notes_id").equals(notesList.getSelectedItem()));
                        break;

                    }
                    if (!rs.isAfterLast())
                    {

                        JTF_notes_id.setText(rs.getString("notes_id"));
                        JTF_subject_id.setText(rs.getString("subject_id"));
                        JTF_course_id.setText(rs.getString("course_id"));
                        JTF_notes_write.setText(rs.getString("notes_write"));
                        JTF_dateofclass.setText(rs.getString("dateofclass"));
                    }
                }
                catch (SQLException selectException)
                {

                    displaySQLErrors(selectException);
                }
            }
        });
        JB_delete.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e)
            {
                try
                {
                    int
a=JOptionPane.showConfirmDialog(pn,"Are you sure do you want to delete:");
                    if(a==JOptionPane.YES_OPTION)
                    {

                        String query="DELETE FROM notes
WHERE notes_id="+JTF_course_id.getText();

```

```

        int i=stmt.executeUpdate(query);

OptionPane.showMessageDialog(pn,"\nDeleted "+i+" rows succesfully");
        loadnotes();
    }
}
catch(SQLException exp)
{
    displaySQLErrors(exp);
}
}
});
}
});
view5.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt)
    {
        pn.removeAll();
        jframe.invalidate();
        jframe.validate();
        jframe.repaint();

        JLabel view=new JLabel("notes View");
        JB_view=new JButton("View");
        Font myFont = new Font("Serif",Font.BOLD,50);
        view.setFont((myFont));

        pn1=new JPanel();
        pn2=new JPanel();
        pn1.add(view);
        pn2.add(JB_view);
        pn.add(pn1);
        pn.add(pn2);
        pn.setLayout(new FlowLayout());

        jframe.add(pn);
        jframe.setSize(800,800);
        jframe.validate();

        JB_view.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent
e)
            {
                JFrame jf=new JFrame("notes
Details");

                JTable jt;
                DefaultTableModel model = new
DefaultTableModel();

                jt= new JTable(model);
                model.addColumn("notes Id");
                model.addColumn("subject Id");

```

```

        model.addColumn("course_id");
        model.addColumn("notes_write");
        model.addColumn("dateofclass");
        try
        {

            rs=stmt.executeQuery("select * from notes");

            while(rs.next())
            {
                model.addRow(new
Object[]{rs.getString("notes_id"),rs.getString("subject_id"),rs.getString("course_id"
),
rs.getString("notes_write"),rs.getString("dateofclass")});
            }
        }
        catch(SQLException exp)
        {
            displaySQLErrors(exp);
        }

        jt.setEnabled(false);
        jt.setBounds(30, 40, 180, 150);
        JScrollPane sp = new

JScrollPane(jt);

        jf.add(sp);
        jf.setSize(800, 400);
        jf.setVisible(true);
    }

    });
}

});
}

}

```

SNMSUI :

```
package SNMS;
import java.awt.*;
import javax.swing.*;
public class SNMSUI extends JFrame
{
    private JPanel Jpn0,Jpn1;
    private JMenuBar mbar;
    private JMenu college,student,subject,course,notes;
    private JMenuItem insert1,update1,view1,delete1;
    private JMenuItem insert2,update2,view2,delete2;
    private JMenuItem insert3,update3,view3,delete3;
    private JMenuItem insert4,update4,view4,delete4;
    private JMenuItem insert5,update5,view5,delete5;
    /*private JMenuItem insert6,update6,view6,delete6;
    private JMenuItem s1,s2,s3,s4,s5;*/
}
```

```

private JLabel labelName;

public void defineComponents()
{
    Jpn0=new JPanel();
    Jpn1=new JPanel();

    mbar=new JMenuBar();

    //statesAvailable=new JMenu("List Of States Available");//d_1
    college=new JMenu("college");
    student=new JMenu("student");
    subject=new JMenu("subject");
    course=new JMenu("course");
    notes=new JMenu("notes");

    labelName=new JLabel("Student Notes Management System");
}
public void addComponents()
{
    add(Jpn0);
    Jpn1.add(labelName);
    Jpn1.setAlignmentY(CENTER_ALIGNMENT);
    Jpn1.setBounds(500,500,800,100);
    Jpn0.add(Jpn1);

    setJMenuBar(mbar);

    mbar.add(college);
    mbar.add(student);
    mbar.add(subject);
    mbar.add(course);
    mbar.add(notes);

    college.add(insert1=new JMenuItem("Insert"));
    college.add(update1=new JMenuItem("Update"));
    college.add(view1=new JMenuItem("View"));
    college.add(delete1=new JMenuItem("Delete"));

    student.add(insert2=new JMenuItem("Insert"));
    student.add(update2=new JMenuItem("Update"));
    student.add(view2=new JMenuItem("View"));
    student.add(delete2=new JMenuItem("Delete"));

    subject.add(insert3=new JMenuItem("Insert"));
    subject.add(update3=new JMenuItem("Update"));
    subject.add(view3=new JMenuItem("View"));
    subject.add(delete3=new JMenuItem("Delete"));

    course.add(insert4=new JMenuItem("Insert"));
    course.add(update4=new JMenuItem("Update"));

```

```

        course.add(view4=new JMenuItem("View"));
        course.add(delete4=new JMenuItem("Delete"));

        notes.add(insert5=new JMenuItem("Insert"));
        notes.add(update5=new JMenuItem("Update"));
        notes.add(view5=new JMenuItem("View"));
        notes.add(delete5=new JMenuItem("Delete"));
    }
    public void registerComponents()
    {
        college obj_college=new
college(Jpn0,SNMSUI.this,insert1,update1,view1,delete1);
        obj_college.buildGUI();
        student obj_student=new
student(Jpn0,SNMSUI.this,insert2,update2,view2,delete2);
        obj_student.buildGUI();
        subject obj_subject=new
subject(Jpn0,SNMSUI.this,insert3,update3,view3,delete3);
        obj_subject.buildGUI();
        course obj_course=new course(Jpn0,SNMSUI.this,insert4,update4,view4,delete4);
        obj_course.buildGUI();
        notes obj_notes=new notes(Jpn0,SNMSUI.this,insert5,update5,view5,delete5);
        obj_notes.buildGUI();
    }
    public SNMSUI()
    {
        defineComponents();
        addComponents();
        registerComponents();
        setSize(400,500);
        setBackground(Color.GRAY);
        setVisible(true);
        setTitle("STUDENT NOTES MANAGEMENT SYSTEM");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

```

MAIN:

```

package SNMS;
public class Main
{
    public static void main(String []args)
    {
        new SNMSUI();
    }
}






```


GITHUB LINK:


<https://github.com/Prajwalreddydornala/STUDENT-NOTES-MANAGEMENT-SYSTEM>

FOLDER STRUCTURE:








> This PC > Documents > SNMS

Name	Status	Date modified	Type	Size
 .settings		22-06-2021 12:58	File folder	
 bin		23-06-2021 10:08	File folder	
 src		23-06-2021 10:18	File folder	
 .classpath		22-06-2021 20:08	CLASSPATH File	1 KB
 .project		22-06-2021 12:58	PROJECT File	1 KB


> This PC > Documents > SNMS > src >

Name	Status	Date modified	Type	Size
 SNMS		23-06-2021 10:59	File folder	

This PC > Documents > SNMS > src > SNMS

Name	Status	Date modified	Type	Size
 college		23-06-2021 10:11	JAVA File	11 KB
 course		23-06-2021 10:57	JAVA File	16 KB
 Main		23-06-2021 10:00	JAVA File	1 KB
 notes		23-06-2021 14:29	JAVA File	13 KB
 SNMSUI		23-06-2021 11:10	JAVA File	4 KB
 student		23-06-2021 10:27	JAVA File	13 KB
 subject		23-06-2021 10:44	JAVA File	12 KB

Testing:

 STUDENT NOTES MANAGEMENT SYSTEM

college student subject course notes

student_id:

student Name:

student Contact:

student Address:

college Code:

Submit


86

88

75

56

Message

 **SQLException: ORA-00001: unique constraint (IT19737086.PK_STUDENT_ID) violated**

SQLState: 23000

VendorError: 1

OK

TITLE: STUDENT NOTES MANAGEMENT SYSTEM

The screenshot shows the 'STUDENT NOTES MANAGEMENT SYSTEM' window with a navigation bar containing 'college', 'student', 'subject', 'course', and 'notes'. The 'student' tab is active. The form contains the following fields:

- student_id:** 75
- student Name:** esha
- student Contact:** 6192534178
- student Address:** grethapur
- college Code:** 101

A 'Submit' button is present. To its right is a dropdown menu with the following options: 86, 88 (selected), 75, and 56.

A 'Message' dialog box is displayed with the following text:

Message

SQLException: ORA-00001: unique constraint (IT19737086.PK_STUDENT_ID) violated

SQLState: 23000
VendorError: 1

OK

The second screenshot shows the same system window with the 'subject' tab active. The form contains the following fields:

- subject_id:** 404
- subject Name:** SE
- teacher contact:** 7398909012
- teacher name:** varun

A 'Submit' button is present. To its right is a dropdown menu with the following options: 401, 402, 403 (selected), and 404.

A 'Message' dialog box is displayed with the following text:

Message

SQLException: ORA-00001: unique constraint (IT19737086.PK_SUBJECT_ID) violated

SQLState: 23000
VendorError: 1

OK

TITLE: STUDENT NOTES MANAGEMENT SYSTEM

STUDENT NOTES MANAGEMENT SYSTEM

college student subject course notes

course_id: 733

course Name: BM

subject Id: 402

course credits: 8

Enrollment Year: 2019

Start Date: TE('11/12/2019','DD/MM/YY')

End Date: TE('18/11/2020','DD/MM/YY')

college Code: 103

student Id: 56

Submit

731
732
734
733

Message

SQLException: ORA-00001: unique constraint (IT19737086.PK_COURSE_ID) violated

SQLState: 23000

VendorError: 1

OK

STUDENT NOTES MANAGEMENT SYSTEM

college student subject course notes

notes_Id: 506

subject Id: 404

course id: 732

notes write: Sir explained about

dateofclass: TO_DATE('11/04/2019','DD/MM/YY')

Submit

501
502
505
504

Message

SQLException: ORA-00001: unique constraint (IT19737086.PK_NOTES_NI) violated

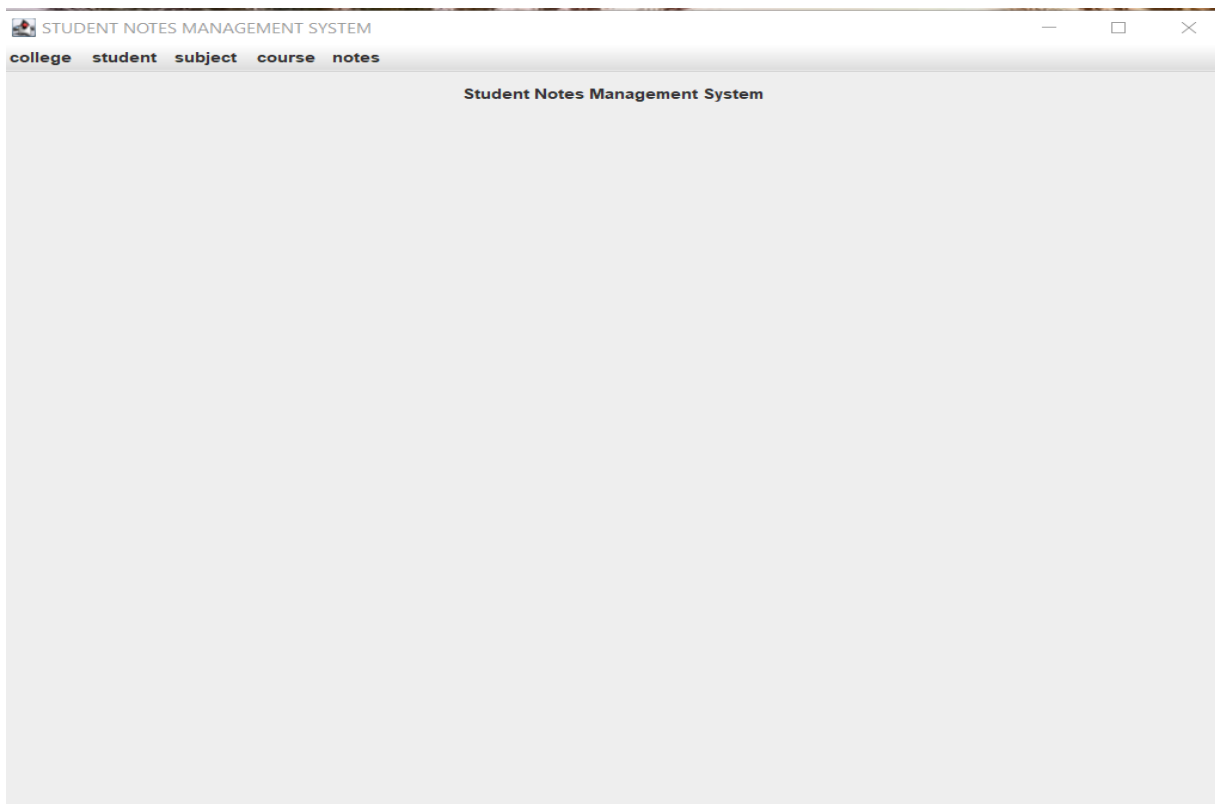
SQLState: 23000

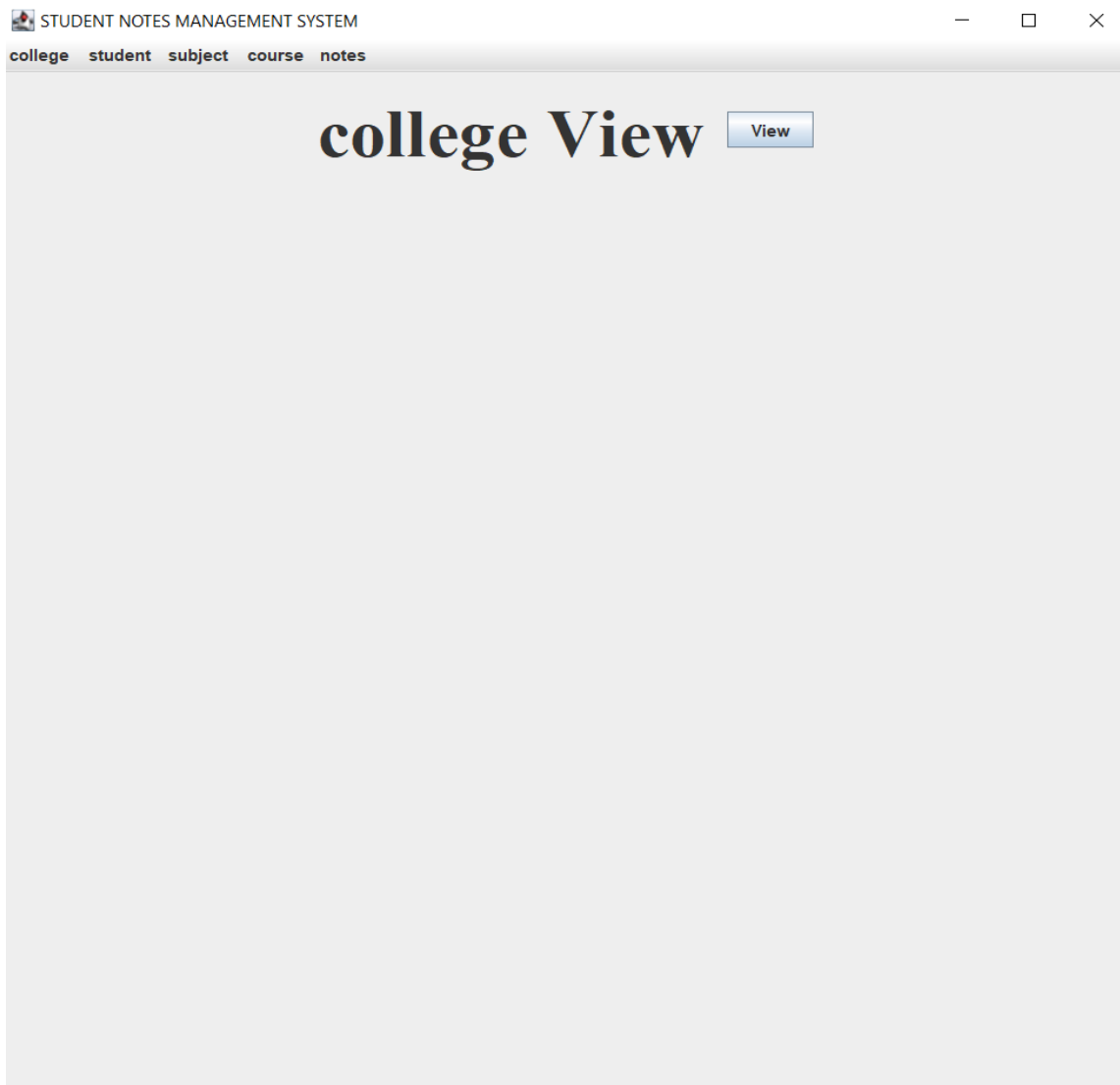
VendorError: 1

OK

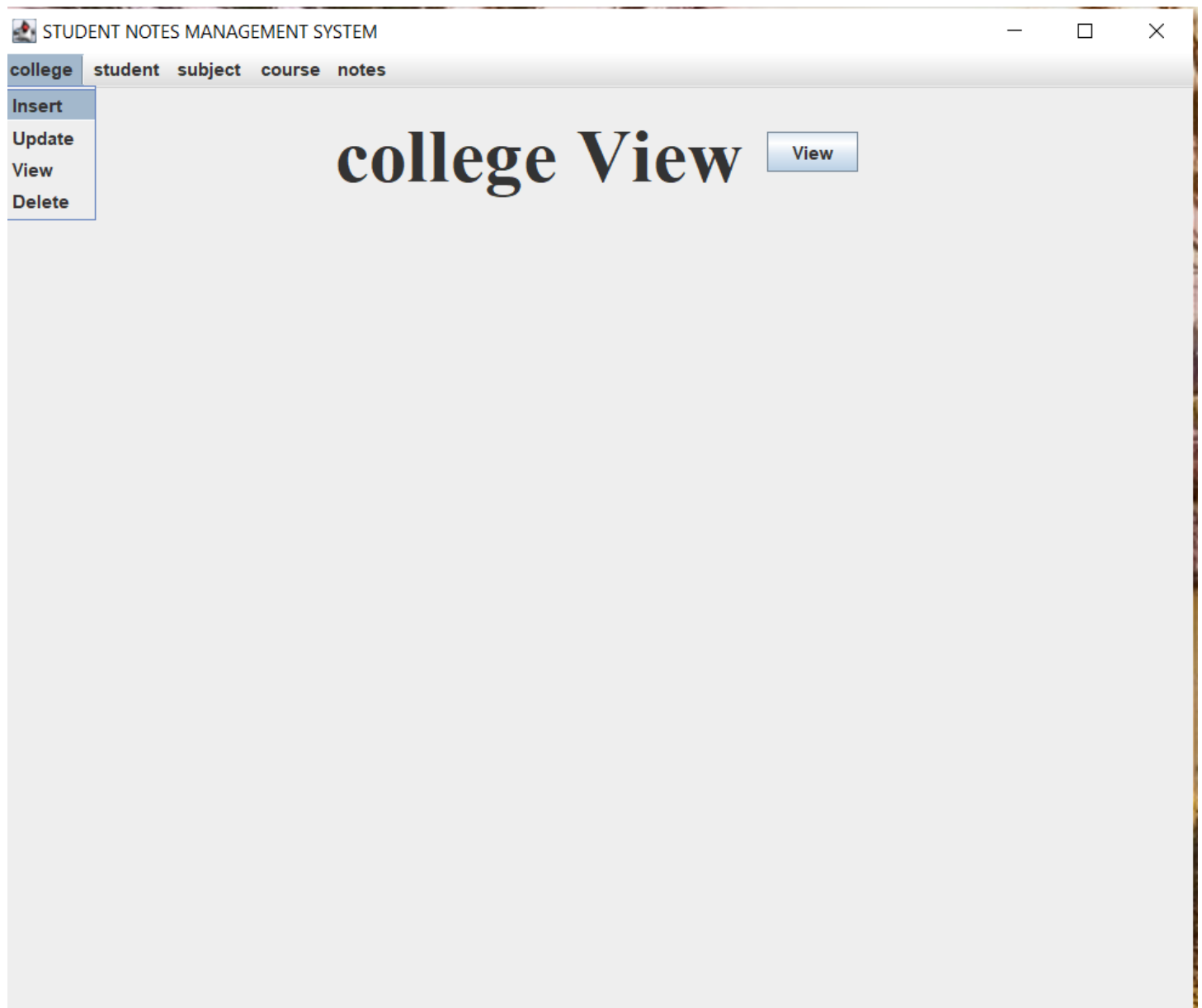
OUTPUT SCREENSHOTS:


TITLE: STUDENT NOTES MANAGEMENT SYSTEM





college Details		
college_code	college name	college address
101	vasavi	ibrahimbagh
102	vjit	miyapur
103	jbit	himampur
104	gsit	uppal
105	cbit	sherling
106	cmr	wargal



 STUDENT NOTES MANAGEMENT SYSTEM

college

student

subject

course

notes

college Code:

107

college Name:

mrit

college Address:

malyanagar


Submit

103

104

105

106

 STUDENT NOTES MANAGEMENT SYSTEM

college

student

subject

course

notes

college Code:

107

college Name:

mrit

college Address:

malyanagar

Submit


103

104

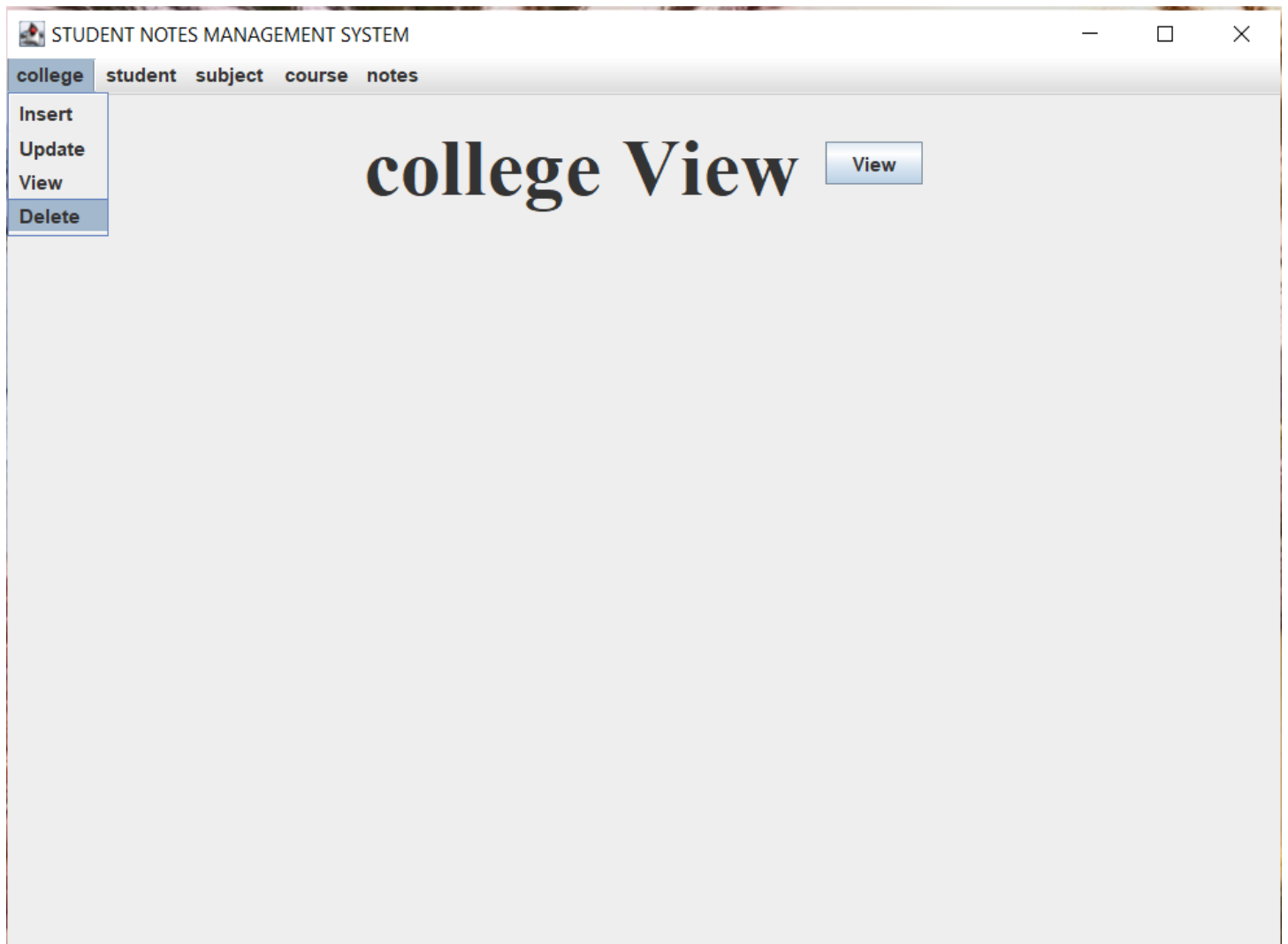
105


106

Message

 Inserted 1 rows successfully

OK



 STUDENT NOTES MANAGEMENT SYSTEM

college

student

subject

course

notes

college Code:

106

college Name:

cmr

college Address:

wargal

Delete

Select an Option


?

Are you sure want to Delete:

Yes

No

Cancel

 STUDENT NOTES MANAGEMENT SYSTEM

college

student

subject

course

notes

Insert

Update

View

Delete

college Code:

college Name:

college Address:

Modify

103

104

105

107

STUDENT NOTES MANAGEMENT SYSTEM

college student subject course notes

college Code:
107

college Name:
mrit

college Address:
malyanagar

Modify

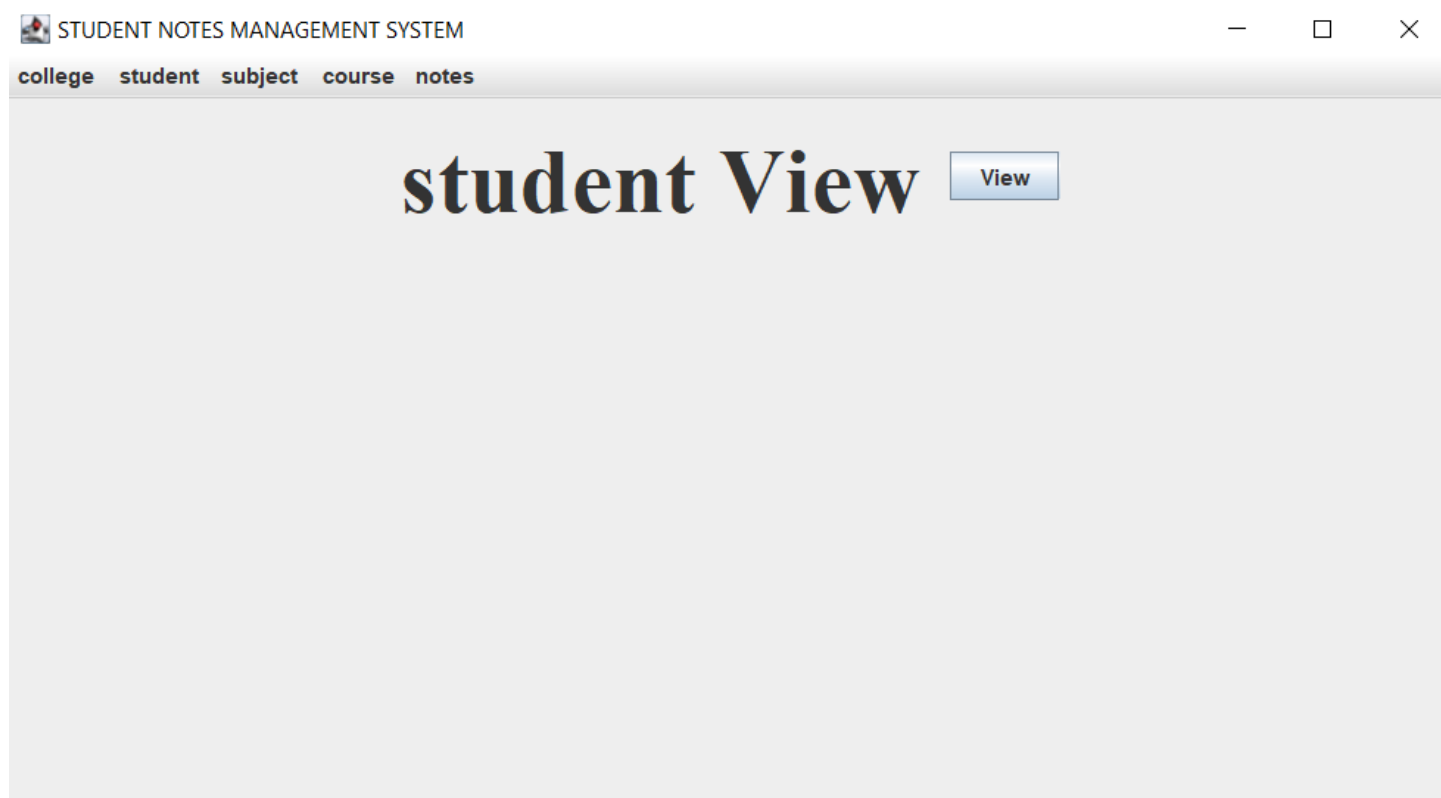
Input

Enter New college Name:
mvrits

OK Cancel

college Details

college_code	college name	college address
101	vasavi	ibrahimbagh
102	vjit	miyapur
103	jbit	himampur
104	gsit	uppal
105	cbit	sherling
107	mvrits	malyanagar



The screenshot shows a window titled "student Details". It contains a table with the following data:

student_id	student name	student contact	student address	college Code
86	Amar	8272838811	Guntur	101
88	Deepu	9273782111	Karimnagar	102
82	Weesa	8428192467	Siddipet	103
75	Joshua	9639628367	Hyderabad	104
56	Prasad	7538290222	Nizambad	105

STUDENT NOTES MANAGEMENT SYSTEM

college

student

subject

course

notes

Insert

Update

View

Delete

student_id:

student Name:

student Contact:

student Address:

college Code:

101


Submit

86

88

82

75

 STUDENT NOTES MANAGEMENT SYSTEM

college

student

subject

course

notes

student_id:

071

student Name:

rishi

student Contact:

8271267124

student Address:

ahmedpur

college Code:

107

Submit


86

88


82

75

Message

 Inserted 1 rows successfully

OK

 STUDENT NOTES MANAGEMENT SYSTEM

college

student

subject

course

notes

Insert

Update

View

Delete

student_id:

student Name:

student Contact:

student Address:

college Code:


Modify

86

88

82

75

 STUDENT NOTES MANAGEMENT SYSTEM

college student subject course notes

student_id:

student Name:


student Contact:

student Address:

college Code:

Modify


Input



Enter New student Name:

OK

Cancel

 STUDENT NOTES MANAGEMENT SYSTEM

college

student

subject

course

notes

student_id:

56

student Name:

Viswak

student Contact:

7538290222

student Address:


Nizambad

college Code:


105

Modify

Message

 Updated 1 rows succesfully

OK

 STUDENT NOTES MANAGEMENT SYSTEM

college

student

subject

course

notes

Insert

Update

View

Delete

student_id:

student Name:

student Contact:

student Address:

college Code:

Delete

86

88

82

75

STUDENT NOTES MANAGEMENT SYSTEM

college student subject course notes

student_id:
82

student Name:
Weesa

student Contact:
8428192467

student Address:
Siddipet

college Code:
103

Delete

Message
Deleted 1 rows succesfully
OK

STUDENT NOTES MANAGEMENT SYSTEM


college student subject course notes

student View


View

student Details

student_id	student name	student contact	student address	college Code
86	Amar	8272838811	Guntur	101
88	Deepu	9273782111	Karimnagar	102
75	Joshua	9639628367	Hyderabad	104
56	Viswak	7538290222	Nizambad	105
71	rishi	8271267124	ahmedpur	107



subject_id	subject name	teacher contact	teacher name
401	Physics	7292887654	Iqbal
402	Maths	9428341782	Imam
403	DBMS	7258279812	Indra
404	PS	7367215267	Esha
405	HumanValues	9364262987	Tarun



college student **subject** course notes


subject_id:


subject Name:

teacher contact:

teacher name:

Message

 Inserted 1 rows successfully

 STUDENT NOTES MANAGEMENT SYSTEM

college student subject course notes

subject_id:


subject Name:

teacher contact:

teacher name:

Modify

Input



Enter New subject Name:

OK

Cancel

STUDENT NOTES MANAGEMENT SYSTEM

college student subject course notes

subject_id:
405

subject Name:
HumanValues

teacher contact:
9364262987

teacher name:
Tarun

Delete


Select an Option

Are you sure want to Delete:

Yes No Cancel


subject Details

subject_id	subject name	teacher contact	teacher name
401	Physics	7292887654	Iqbal
402	Discrete Math	9428341782	Imam
403	DBMS	7258279812	Indra
404	PS	7367215267	Esha
406	English	Subhash	9872728102

 course Details

— □ ×

course_id	course name	subject Id	course credits	Enrollment Year	Course Start ...	Course End D...	college Code	student Id
731	BE	401	4	2018	2018-11-12 00...	2019-05-22 00...	101	86
732	ME	402	5	2017	2017-06-20 00...	2017-12-20 00...	102	88
734	MCA	404	4	2020	2020-02-11 00...	2020-08-21 00...	104	75

 STUDENT NOTES MANAGEMENT SYSTEM

college student subject course notes

course_id:

733

course Name:

BLA

subject Id:

402

course credits:

7

Enrollment Year:

2019

Start Date:

E('18/07/2019','DD/MM/YY')

End Date:

E('12/12/2020','DD/MM/YY')

college Code:

103

student Id:

71


Submit

731

732

734

Message

 Inserted 1 rows successfully

OK

STUDENT NOTES MANAGEMENT SYSTEM

college student subject course notes

course_id: 733

course Name: BLA

subject Id: 402

course credits: 7

Enrollment Year: 2019

Start Date: 2019-07-18 00:00:00

End Date: 2020-12-12 00:00:00

college Code: 103

student Id: 71

Modify

Input


Enter New course name:

BCA


OK Cancel

course Details

course_id	course name	subject Id	course credits	Enrollment Year	Course Start ...	Course End D...	college Code	student Id
731	BE	401	4	2018	2018-11-12 00:...	2019-05-22 00:...	101	86
732	ME	402	5	2017	2017-06-20 00:...	2017-12-20 00:...	102	88
734	MCA	404	4	2020	2020-02-11 00:...	2020-08-21 00:...	104	75
733	BCA	402	7	2019	2019-07-18 00:...	2020-12-12 00:...	103	71

 notes Details

notes Id	subject Id	course id	notes_write	dateofclass
501	401	731	Today there has been a disc..	2018-10-11 00:00:00
502	402	732	It has been notes	2019-10-11 00:00:00
504	404	734	There is a clear	2019-01-01 00:00:00
506	401	731	abc	2019-07-29 00:00:00

 STUDENT NOTES MANAGEMENT SYSTEM

college student subject course notes

notes_id:

subject Id:


course id:

notes write:

dateofclass:

501
502
504
506

Message

 Inserted 1 rows successfully

STUDENT NOTES MANAGEMENT SYSTEM

college student subject course notes

notes_id:
505

subject Id:
404

course id:
732

notes write:
There was no class

dateofclass:
2019-11-12 00:00:00

Modify

501
502
505
504


Message

Updated 1 rows succesfully

OK

notes Details


notes Id	subject Id	course id	notes_write	dateofclass
501	401	731	Today there has been a disc..	2018-10-11 00:00:00
502	402	732	It has been notes	2019-10-11 00:00:00
505	404	732	There was no class	2019-11-12 00:00:00
504	404	734	There is a clear	2019-01-01 00:00:00
506	401	731	abc	2019-07-29 00:00:00

 STUDENT NOTES MANAGEMENT SYSTEM

college student subject course notes

notes View

View

 notes Details

notes Id	subject Id	course id	notes_write	dateofclass
501	401	731	Today there has been a disc...	2018-10-11 00:00:00
502	402	732	It has been notes	2019-10-11 00:00:00
505	404	732	We had a debate on	2019-11-12 00:00:00
504	404	734	There is a clear	2019-01-01 00:00:00
506	401	731	abc	2019-07-29 00:00:00

The data entered in the above form is updated in the “college” table of the Oracle database 11g as:

```
SQL> select * from college;

COLLEGE_NAME
-----
COLLEGE_ADDRESS
-----
101  vasavi
ibrahimbagh

102  vjit
miyapur

103  jbit
himampur

COLLEGE_NAME
-----
COLLEGE_ADDRESS
-----
104  gsit
uppal

105  cbit
sherling

107  mvrits
malyanagar

6 rows selected.
```

The data entered in the above form is updated in the “student” table of the Oracle database 11g as:

```
SQL> select * from student;

STUDENT_ID STUDENT_NAME STUDENT_CO
-----
STUDENT_ADDRESS COLLEGE
-----
86 Amar 8272838811 101
Guntur

88 Deepu 9273782111 102
Karimnagar

75 Joshua 9639628367 104
Hyderabad

STUDENT_ID STUDENT_NAME STUDENT_CO
-----
STUDENT_ADDRESS COLLEGE
-----
56 Viswak 7538290222 105
Nizambad

71 rishi 8271267124 107
ahmedpur
```

The data entered in the above form is updated in the “subject” table of the

Oracle database 11g as:

```
SQL> select * from subject;
```

SUBJECT_ID	SUBJECT_NAME	TEACHER_CO	TEACHER_NAME
401	Physics	7292887654	Iqbal
402	Discrete Math	9428341782	Imam
403	DBMS	7258279812	Indra
404	PS	7367215267	Esha
406	English	9872728102	Subhash

The data entered in the above form is updated in the “course” table of the Oracle database 11g as:

```
SQL> select * from course;
```

COURSE_ID	COURSE_NAME	SUBJECT_ID	COURSE_CREDITS	ENROLLMENT_YEAR
731	BE	401	4	2018
732	ME	402	5	2017
734	MCA	404	4	2020

COURSE_ID	COURSE_NAME	SUBJECT_ID	COURSE_CREDITS	ENROLLMENT_YEAR
733	BCA	402	7	2019

The data entered in the above form is updated in the “notes” table of the Oracle database 11g as:

```
SQL> select * from notes;
```

NOTES_ID	SUBJECT_ID	COURSE_ID	NOTES_WRITE	DATEOFCLA
501	401	731	Today there has been a discussion on	11-OCT-18
502	402	732	It has been notes	11-OCT-19
505	404	732	There was no class	12-NOV-19
504	404	734	There is a clear	01-JAN-19
506	401	731	abc	29-JUL-19

Results:

I had successfully completed PROJECT on “ STUDENT NOTE MANAGEMENT SYSTEM”.

Discussion and future Work:

This application provides the students to store and study notes respectively for various subject details and course details. While working on this application , I imagined of having this application store the marks in various subjects in various examinations and give feedback on performance. I also would like this application to be able to conduct quizzes on its own from the notes written. It would be very helpful if a html version is developed of this.

CONCLUSION:

Thus, a Java SWING based STUDENT NOTES MANAGEMENT SYSTEM is created which is connected to the Oracle 11g database. Therefore, all the entries and details are directly updated on their respective tables created in the database.

REFERENCES:

- <https://docs.oracle.com/javase/7/docs/api/>
- <https://www.javatpoint.com/dbms-tutorial>
- <https://en.wikipedia.org/wiki/Note-taking>