EDUCATION ECOSYSTEM FOR SPECIALLY ABLED STUDENTS

A
Report
Submitted in partial fulfilment of the
Requirements for the award of the Degree of

BACHELOR OF ENGINEERING
IN
INFORMATION TECHNOLOGY
By

B. VINAY KUMAR REDDY<1602-21-737-065>
Under the guidance of Ms B. Leelavathy

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

2022-2023

ABSTRACT

This project aims to identify provisions and improvements that are needed in the education ecosystem for specially abled students to ensure compliance governance, and conduct. The project will involve conducting research and analysis to understand the current state of the education system for specially abled students, including policies, infrastructure, programs. Based on this analysis, recommendations will be made to improve compliance with regulations. The ultimate goal is to create a more inclusive and supportive education environment for students with disabilities, enabling them to achieve their full potential.

Requirement Analysis

List of Tables:

1.STUDENT

2.FACULTY

3.ACCOMMODATION

4.COURSE

5.FACILITY

List of attributes with their domain types:

Student:

- Student _id number(20)
- Name varchar(20)
- Gender varchar(10)
- Disability Type varchar(200)
- Contact info Number(10)
- Enrollment_id Integer

Faculty:

- Teacher id Integer
- Name varchar(25)
- Contact info Number(10)
- Specialisation varchar(200)

Course:

- Course Id Integer
- Course name varchar(30)

- Faculty Id Integer

Accommodation:

- Student Id Integer
- Accommodation Type varchar(200)

- Course_id INTEGER

Facilities:

- Student_id Number(20)
- Disability type varchar(200)
- Facility id number(20)

DATABASE DESIGN

Mapping Cardinality and Participation Constraints

1.Student - Course:

Cardinality: Many-to-Many(A student can be enrolled in many courses).

Participation Constraints: partial (since a student can exist without a particular disability)

2.Student-Facility:

Cardinality: One-to-Many (since a student can have multiple disabilities)

Participation Constraints: Total (since every student have Disability)

3.Faculty-Course:

Cardinality: One-to-Many (since a Faculty member can teach many courses)

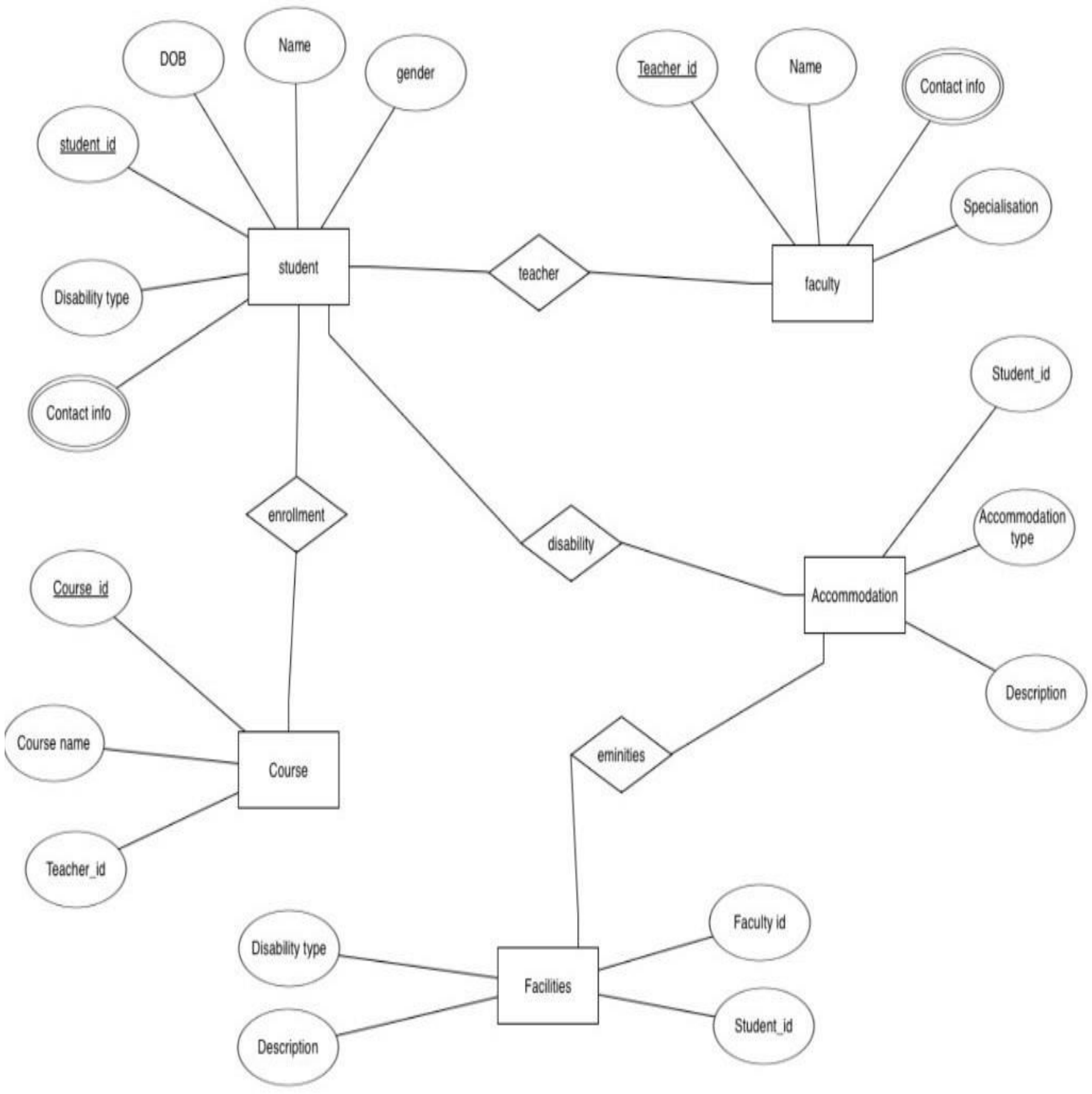
4. Student-Accommodation:

Cardinality: One-to-Many (A student can have many accommodations)

5.Accommodation-Student:

Cardinality: Many-to-One (An Accommodation can be provided to one student)

Entity Relationship Diagram



DDL Commands:

SQL> create table students_1(

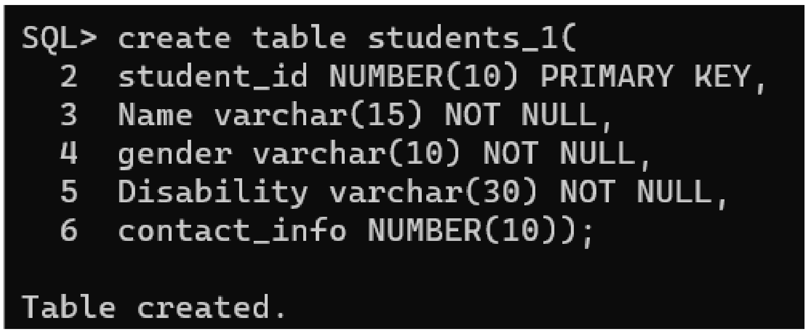
2 student_id NUMBER(10) PRIMARY KEY,

3 Name varchar(15) NOT NULL,

4 gender varchar(10) NOT NULL,

5 Disability varchar(30) NOT NULL,

6 contact_info NUMBER(10));



```
SQL> create table students_1(
  2  student_id NUMBER(10) PRIMARY KEY,
  3  Name varchar(15) NOT NULL,
  4  gender varchar(10) NOT NULL,
  5  Disability varchar(30) NOT NULL,
  6  contact_info NUMBER(10));

Table created.
```

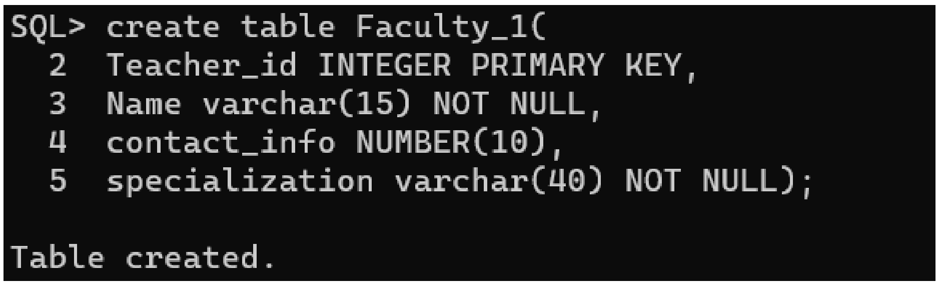
SQL> create table Faculty_1(

2 Teacher_id INTEGER PRIMARY KEY,

3 Name varchar(15) NOT NULL,

4 contact_info NUMBER (10),

5 specialization varchar (40) NOT NULL);

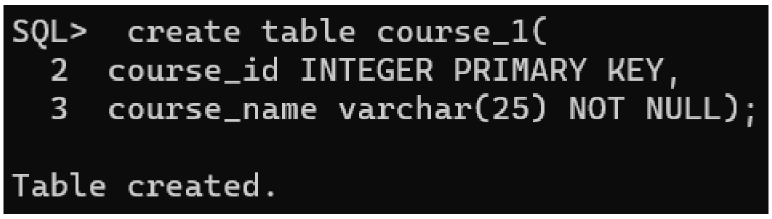


```
SQL> create table Faculty_1(
  2  Teacher_id INTEGER PRIMARY KEY,
  3  Name varchar(15) NOT NULL,
  4  contact_info NUMBER(10),
  5  specialization varchar(40) NOT NULL);

Table created.
```

SQL> create table course_1(

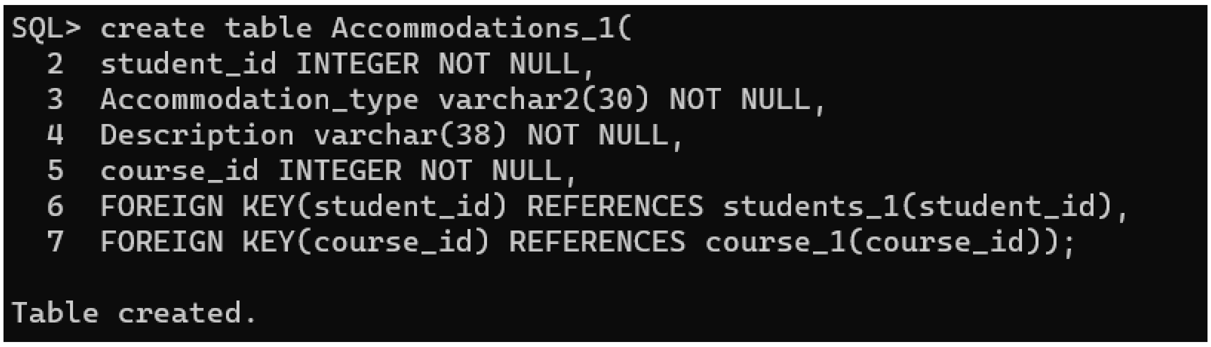
2 course_id INTEGER PRIMARY KEY,

3 course_name varchar(25) NOT NULL);



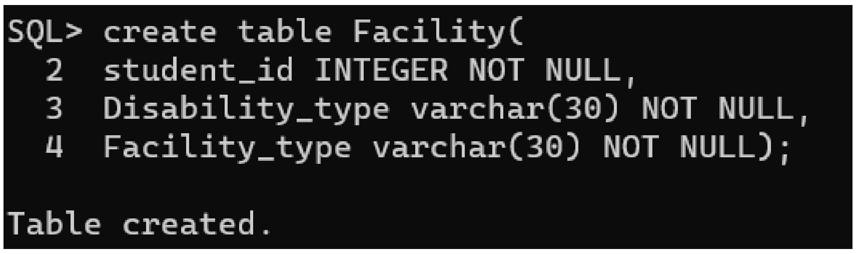
```
SQL>  create table course_1(
  2  course_id INTEGER PRIMARY KEY,
  3  course_name varchar(25) NOT NULL);

Table created.
```

```
SQL> create table Accommodations_1(
  2  studnt_id INTEGER NOT NULL,
  3  Accommodation_type varchar2(30) NOT NULL,
  4  Description varchar(38) NOT NULL,
  5  course_id INTEGER NOT NULL,
  6  FOREIGN KEY(student_id) REFERENCES students_1(student_id),
  7  FOREIGN KEY(course_id) REFERENCES course_1(course_id));
```



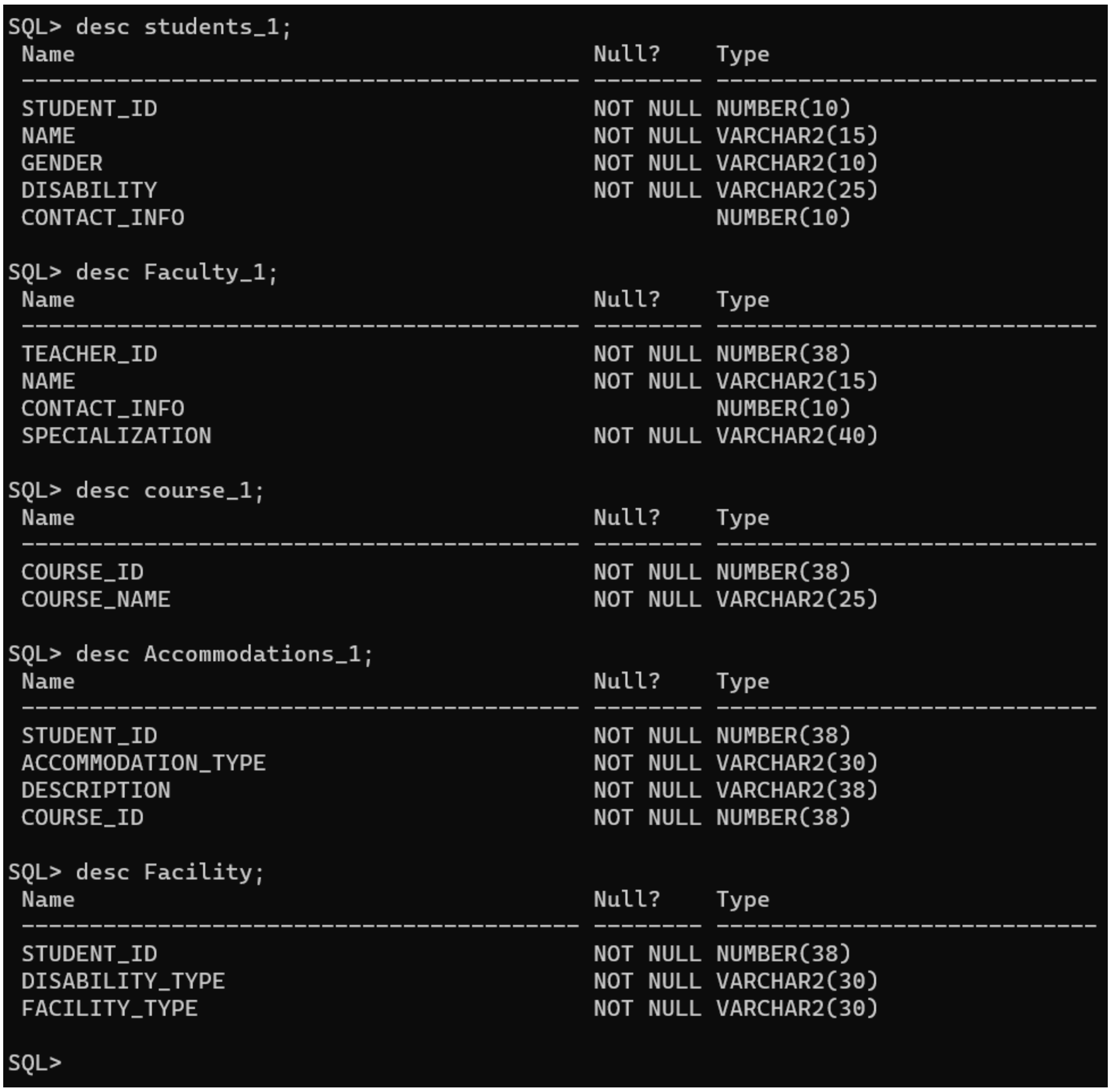
```
SQL> create table Accommodations_1(
  2  student_id INTEGER NOT NULL,
  3  Accommodation_type varchar2(30) NOT NULL,
  4  Description varchar(38) NOT NULL,
  5  course_id INTEGER NOT NULL,
  6  FOREIGN KEY(student_id) REFERENCES students_1(student_id),
  7  FOREIGN KEY(course_id) REFERENCES course_1(course_id));

Table created.
```

```
SQL> create table Facility(
  2  student_id INTEGER NOT NULL,
  3  Disability_type varchar(30) NOT NULL,
  4  Facility_type varchar(30) NOT NULL);
```



```
SQL> create table Facility(
  2  student_id INTEGER NOT NULL,
  3  Disability_type varchar(30) NOT NULL,
  4  Facility_type varchar(30) NOT NULL);

Table created.
```

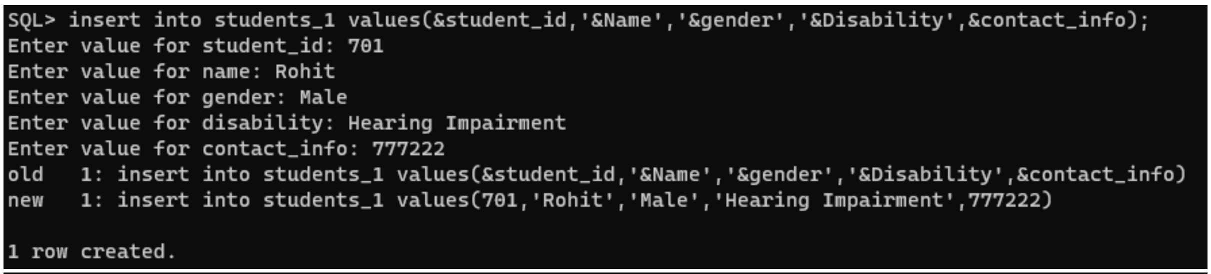
TABLES:



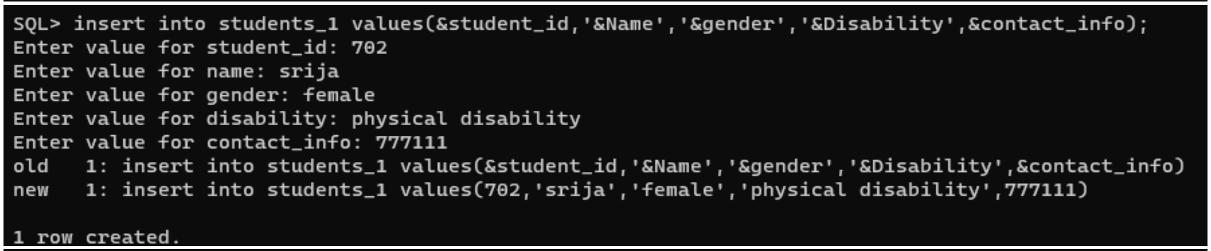
```
SQL> desc students_1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STUDENT_ID                                NOT NULL NUMBER(10)
 NAME                                      NOT NULL VARCHAR2(15)
 GENDER                                    NOT NULL VARCHAR2(10)
 DISABILITY                                NOT NULL VARCHAR2(25)
 CONTACT_INFO                                       NUMBER(10)

SQL> desc Faculty_1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TEACHER_ID                                NOT NULL NUMBER(38)
 NAME                                      NOT NULL VARCHAR2(15)
 CONTACT_INFO                                       NUMBER(10)
 SPECIALIZATION                            NOT NULL VARCHAR2(40)

SQL> desc course_1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COURSE_ID                                 NOT NULL NUMBER(38)
 COURSE_NAME                               NOT NULL VARCHAR2(25)

SQL> desc Accommodations_1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STUDENT_ID                                NOT NULL NUMBER(38)
 ACCOMMODATION_TYPE                        NOT NULL VARCHAR2(30)
 DESCRIPTION                               NOT NULL VARCHAR2(38)
 COURSE_ID                                 NOT NULL NUMBER(38)

SQL> desc Facility;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STUDENT_ID                                NOT NULL NUMBER(38)
 DISABILITY_TYPE                           NOT NULL VARCHAR2(30)
 FACILITY_TYPE                             NOT NULL VARCHAR2(30)

SQL>
```

DML COMMANDS:

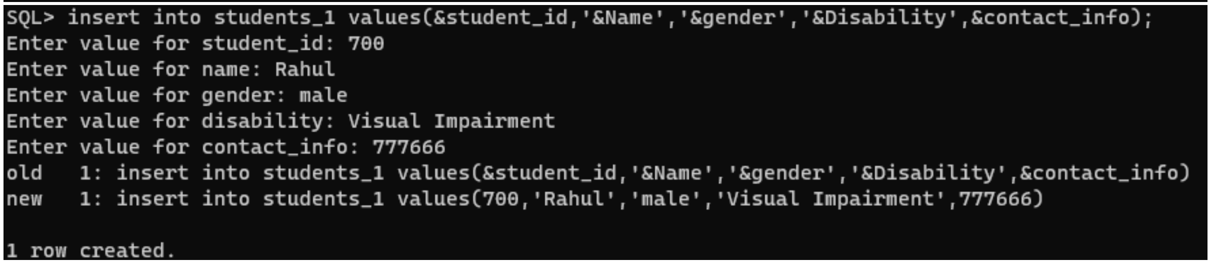
SQL> insert into students_1 values(&student_id,'&Name','&gender','&Disability',&contact_info);



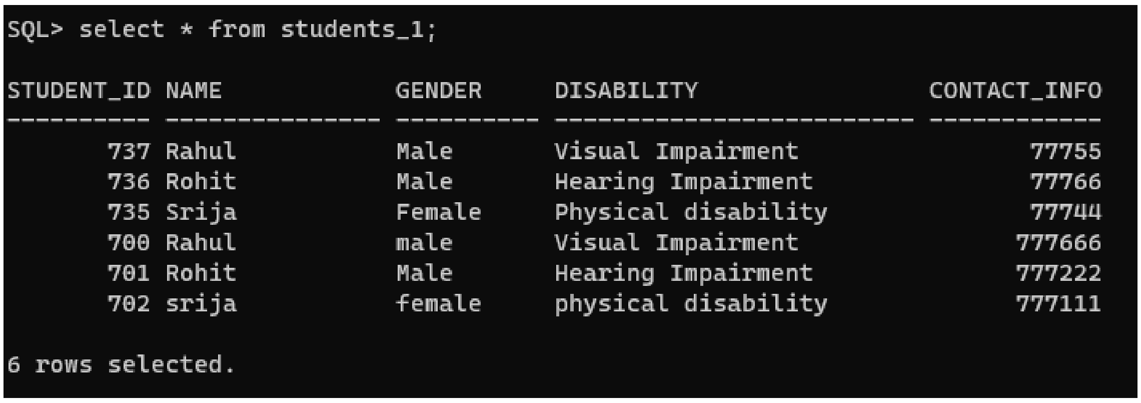
```
SQL> insert into students_1 values(&student_id,'&Name','&gender','&Disability',&contact_info);
Enter value for student_id: 701
Enter value for name: Rohit
Enter value for gender: Male
Enter value for disability: Hearing Impairment
Enter value for contact_info: 777222
old   1: insert into students_1 values(&student_id,'&Name','&gender','&Disability',&contact_info)
new   1: insert into students_1 values(701,'Rohit','Male','Hearing Impairment',777222)

1 row created.
```



```
SQL> insert into students_1 values(&student_id,'&Name','&gender','&Disability',&contact_info);
Enter value for student_id: 702
Enter value for name: srija
Enter value for gender: female
Enter value for disability: physical disability
Enter value for contact_info: 777111
old   1: insert into students_1 values(&student_id,'&Name','&gender','&Disability',&contact_info)
new   1: insert into students_1 values(702,'srija','female','physical disability',777111)

1 row created.
```

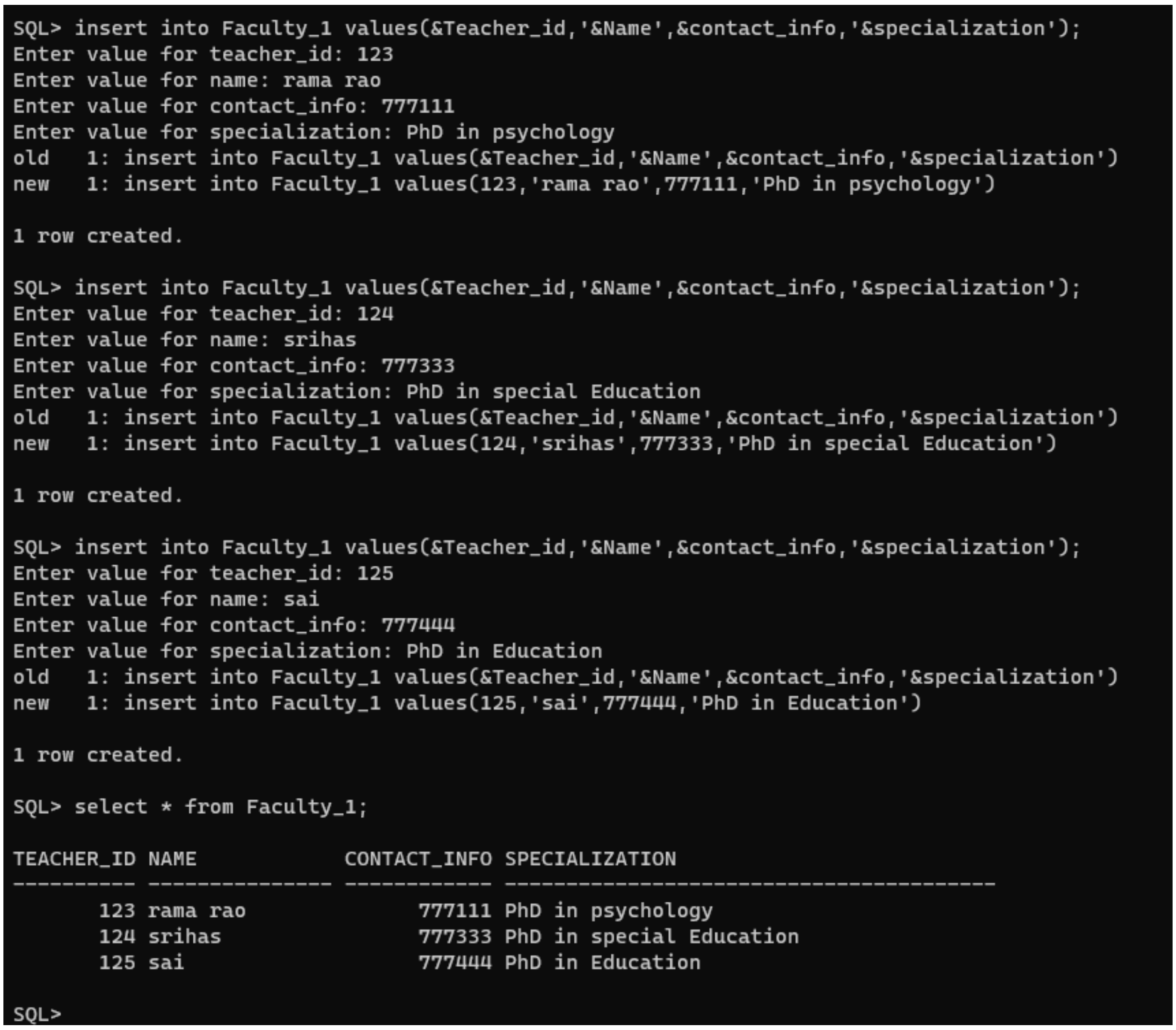


```
SQL> insert into students_1 values(&student_id,'&Name','&gender','&Disability',&contact_info);
Enter value for student_id: 700
Enter value for name: Rahul
Enter value for gender: male
Enter value for disability: Visual Impairment
Enter value for contact_info: 777666
old   1: insert into students_1 values(&student_id,'&Name','&gender','&Disability',&contact_info)
new   1: insert into students_1 values(700,'Rahul','male','Visual Impairment',777666)

1 row created.
```



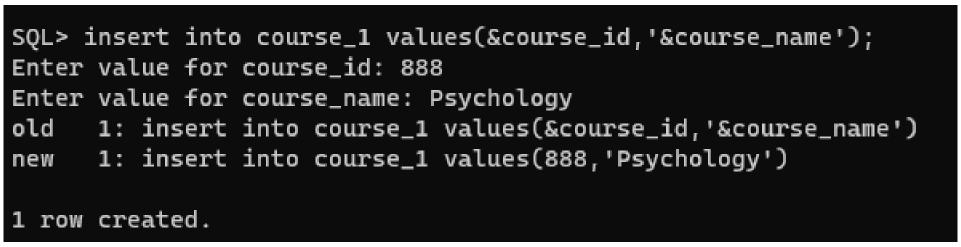
```
SQL> select * from students_1;

STUDENT_ID NAME            GENDER     DISABILITY                CONTACT_INFO
---------- --------------- ---------- ------------------------- ------------
       737 Rahul           Male       Visual Impairment                77755
       736 Rohit           Male       Hearing Impairment               77766
       735 Srija           Female     Physical disability              77744
       700 Rahul           male       Visual Impairment               777666
       701 Rohit           Male       Hearing Impairment              777222
       702 srija           female     physical disability             777111

6 rows selected.
```

SQL> insert into Faculty_1 values(&Teacher_id,'&Name',&contact_info,'&specialization');

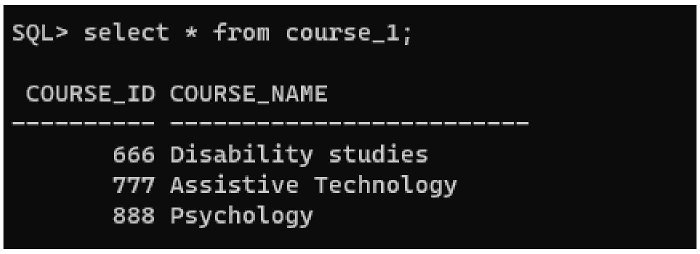


```
SQL> insert into Faculty_1 values(&Teacher_id,'&Name',&contact_info,'&specialization');
Enter value for teacher_id: 123
Enter value for name: rama rao
Enter value for contact_info: 777111
Enter value for specialization: PhD in psychology
old   1: insert into Faculty_1 values(&Teacher_id,'&Name',&contact_info,'&specialization')
new   1: insert into Faculty_1 values(123,'rama rao',777111,'PhD in psychology')

1 row created.

SQL> insert into Faculty_1 values(&Teacher_id,'&Name',&contact_info,'&specialization');
Enter value for teacher_id: 124
Enter value for name: srihas
Enter value for contact_info: 777333
Enter value for specialization: PhD in special Education
old   1: insert into Faculty_1 values(&Teacher_id,'&Name',&contact_info,'&specialization')
new   1: insert into Faculty_1 values(124,'srihas',777333,'PhD in special Education')

1 row created.

SQL> insert into Faculty_1 values(&Teacher_id,'&Name',&contact_info,'&specialization');
Enter value for teacher_id: 125
Enter value for name: sai
Enter value for contact_info: 777444
Enter value for specialization: PhD in Education
old   1: insert into Faculty_1 values(&Teacher_id,'&Name',&contact_info,'&specialization')
new   1: insert into Faculty_1 values(125,'sai',777444,'PhD in Education')

1 row created.

SQL> select * from Faculty_1;

TEACHER_ID NAME            CONTACT_INFO SPECIALIZATION
---------- --------------- ------------ ----------------------------------------
       123 rama rao              777111 PhD in psychology
       124 srihas                777333 PhD in special Education
       125 sai                   777444 PhD in Education

SQL>
```

SQL> insert into course_1 values(&course_id,'&course_name');

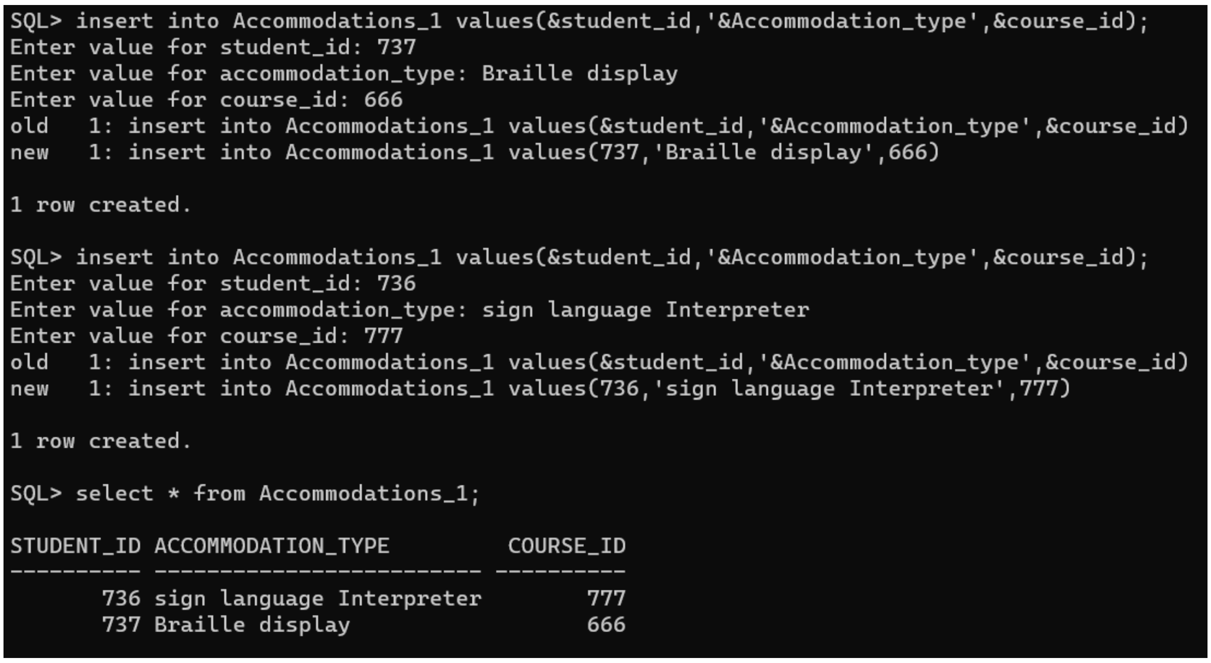


```
SQL> insert into course_1 values(&course_id,'&course_name');
Enter value for course_id: 888
Enter value for course_name: Psychology
old   1: insert into course_1 values(&course_id,'&course_name')
new   1: insert into course_1 values(888,'Psychology')

1 row created.
```



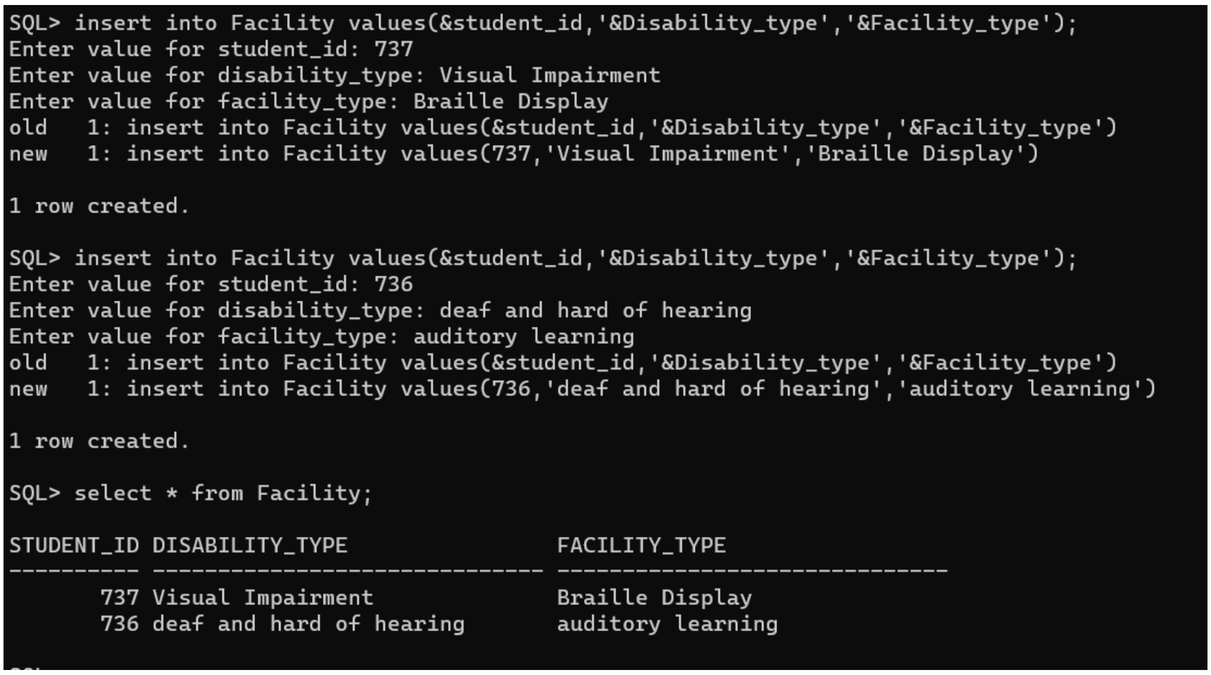
```
SQL> select * from course_1;

 COURSE_ID COURSE_NAME
---------- -------------------------
       666 Disability studies
       777 Assistive Technology
       888 Psychology
```

SQL>insert into Accommodations_1values(&student_id,'&Accommodation_type',&course_id)



```
SQL> insert into Accommodations_1 values(&student_id,'&Accommodation_type',&course_id);
Enter value for student_id: 737
Enter value for accommodation_type: Braille display
Enter value for course_id: 666
old   1: insert into Accommodations_1 values(&student_id,'&Accommodation_type',&course_id)
new   1: insert into Accommodations_1 values(737,'Braille display',666)

1 row created.

SQL> insert into Accommodations_1 values(&student_id,'&Accommodation_type',&course_id);
Enter value for student_id: 736
Enter value for accommodation_type: sign language Interpreter
Enter value for course_id: 777
old   1: insert into Accommodations_1 values(&student_id,'&Accommodation_type',&course_id)
new   1: insert into Accommodations_1 values(736,'sign language Interpreter',777)

1 row created.

SQL> select * from Accommodations_1;

STUDENT_ID ACCOMMODATION_TYPE         COURSE_ID
---------- ------------------------- ----------
       736 sign language Interpreter        777
       737 Braille display                  666
```

SQL> insert into Facility values(&student_id,'&Disability_type','&Facility_type');



```
SQL> insert into Facility values(&student_id,'&Disability_type','&Facility_type');
Enter value for student_id: 737
Enter value for disability_type: Visual Impairment
Enter value for facility_type: Braille Display
old   1: insert into Facility values(&student_id,'&Disability_type','&Facility_type')
new   1: insert into Facility values(737,'Visual Impairment','Braille Display')

1 row created.

SQL> insert into Facility values(&student_id,'&Disability_type','&Facility_type');
Enter value for student_id: 736
Enter value for disability_type: deaf and hard of hearing
Enter value for facility_type: auditory learning
old   1: insert into Facility values(&student_id,'&Disability_type','&Facility_type')
new   1: insert into Facility values(736,'deaf and hard of hearing','auditory learning')

1 row created.

SQL> select * from Facility;

STUDENT_ID DISABILITY_TYPE                FACILITY_TYPE
---------- ------------------------------ ------------------------------
       737 Visual Impairment              Braille Display
       736 deaf and hard of hearing       auditory learning
```

IMPLEMENTATION

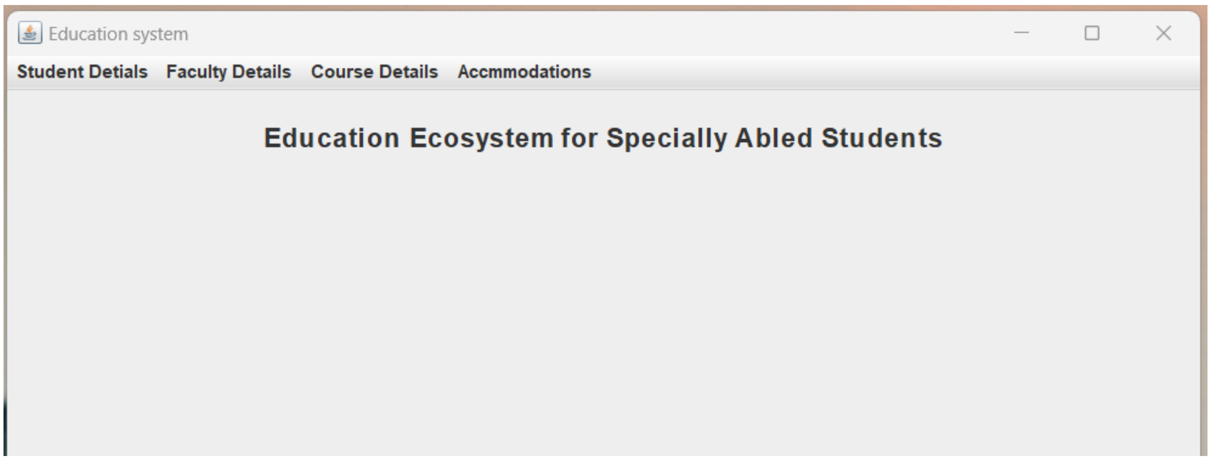
JAVA-SQL Connectivity using JDBC:

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

```
{ DriverManager.registerDriver (new oracle.jdbc.driver.OracleDriver()); //
Connect to Oracle Database Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE"
,"vinay","vinay"); Statementstatement = con.createStatement() String query =
"UPDATE SKILLS SET SS1=" +"'"+ jTextField3.getText() +"',SS2=" +"'"+
jTextField5.getText() +"', AOI ="+" '"+ jTextField2.getText() +"' WHERE SID =+" +
jTextField4.getText(); ResultSet rs = statement.executeQuery(query);
JOptionPane.showMessageDialog(new JFrame(),

"Upadated Successfully", "INFORMATION",
JOptionPane.INFORMATION_MESSAGE); rs.close(); statement.close();
con.close(); }
```

Front-end Programs (User Interfaces) Home Page:

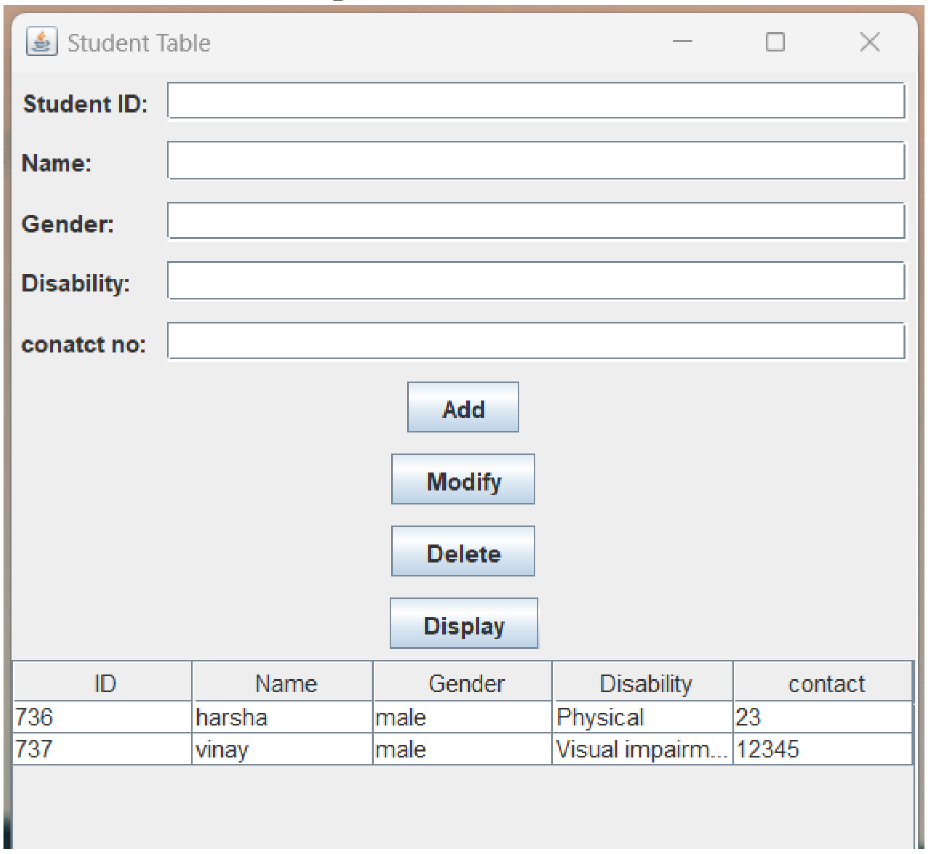
1.Main Page



Education Ecosystem for Specially Abled Students

```
package EducationSystem;

import javax.swing.*;
import javax.swing.ImageIcon;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.util.ArrayList;
import java.util.List;

import java.awt.event.*;

public class MainPage extends JFrame {
     private static final long serialVersionUID = 1L;


    public MainPage() {

        setTitle("Education system");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


        JLabel welcomeLabel = new JLabel("Education Ecosystem
for Specially Abled Students");
        welcomeLabel.setFont(new Font("Arial", Font.BOLD,
18));

welcomeLabel.setHorizontalAlignment(SwingConstants.CENTER);
welcomeLabel.setBorder(BorderFactory.createEmptyBorder(20, 0,
20, 0));
        add(welcomeLabel, BorderLayout.NORTH);




        JMenuBar menuBar = new JMenuBar();

        JMenu StudentMenu = new JMenu("Student Detials");
        JMenu FacultyMenu = new JMenu("Faculty Details");
        JMenu CourseMenu = new JMenu("Course Details");
        JMenu AccoMenu = new JMenu("Accmmodations");

        JMenuItem Viewstudentdetails = new JMenuItem("View
student details");
         Viewstudentdetails.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new StudentTableGUI();
            }
        });


        JMenuItem viewFacultydetails = new JMenuItem("View
Faculty details");
        viewFacultydetails.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new Faculty();
            }
        });

        JMenuItem viewCoursedetails = new JMenuItem("View
Course Details");
        viewCoursedetails.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new Course();
            }
        });


        JMenuItem viewAccommodationsdetails = new
JMenuItem("View Accommodations Details");
        viewAccommodationsdetails.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new Accommodations();
            }
        });


        StudentMenu.add(Viewstudentdetails);
        FacultyMenu.add(viewFacultydetails);
        CourseMenu.add(viewCoursedetails);
        AccoMenu.add(viewAccommodationsdetails);
        //bankMenu.add(viewbankDetails);


        menuBar.add(StudentMenu);
        menuBar.add(FacultyMenu);
        menuBar.add(CourseMenu);
        menuBar.add(AccoMenu);

        setJMenuBar(menuBar);


        addWindowStateListener(new WindowStateListener() {
            public void windowStateChanged(WindowEvent e) {
                if ((e.getNewState() & Frame.MAXIMIZED_BOTH)
== Frame.MAXIMIZED_BOTH) {
                    System.out.println("Window maximized");
                } else {
                    System.out.println("Window not
maximized");
                }
            }
        });


        setSize(800, 600);
        setVisible(true);
    }


    public static void main(String[] args) {

        new MainPage();
    }
}
```

Student Page:


Student Table

Student ID:

Name:

Gender:

Disability:

conatct no:

ID	Name	Gender	Disability	contact
736	harsha	male	Physical	23
737	vinay	male	Visual impairm...	12345

```
package EducationSystem;

import javax.swing.*;

import javax.swing.table.DefaultTableModel;

import java.awt.*;

import java.sql.*;

import java.util.ArrayList;

import java.util.List;


public class StudentTableGUI extends JFrame {

    /**

      *
```

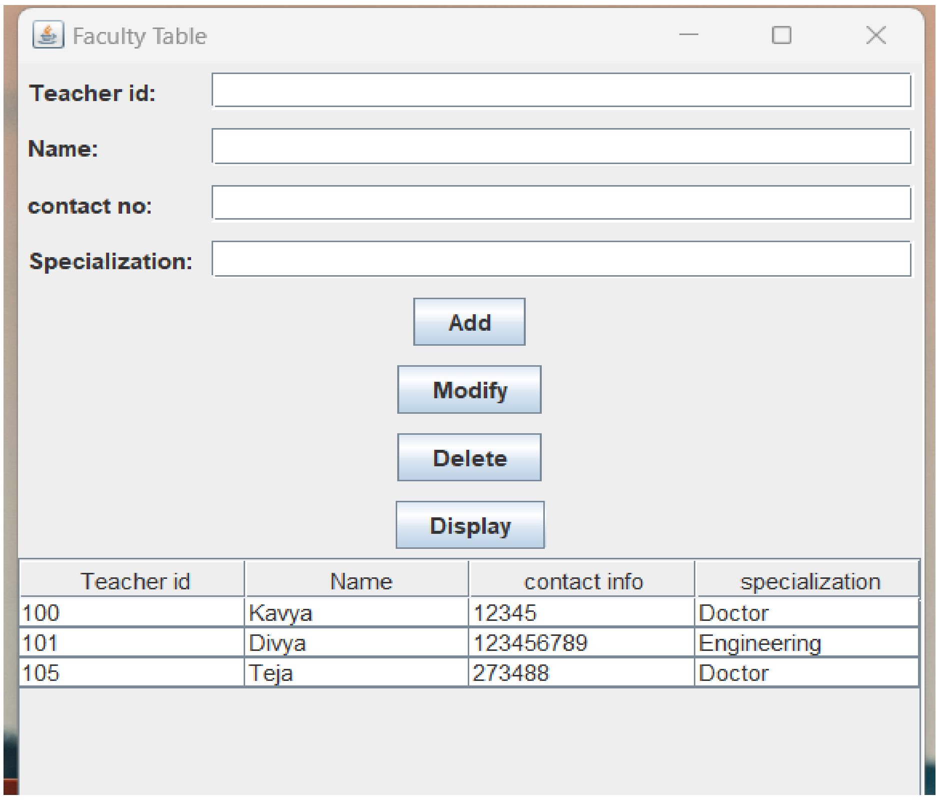
```
       */

      private static final long serialVersionUID = 1L;

      private JTextField txtId, txtName, txtgender, txtDisability,
txtcontact;

    private JTable tblStudent;

    private JButton btnAdd, btnModify, btnDelete, btnDisplay;



    private Connection connection;

      private String gender;

      private String Disability;




    public void Student(String id, String name, String gender2,
String disability2, String aadharNo) {

            // TODO Auto-generated constructor stub

      }



      public StudentTableGUI() {

           //this.MainPage = MainPage;

         initializeUI();

         connectToDatabase();

         displayStudents();

    }


    private void initializeUI() {

         txtId = new JTextField();

         txtName = new JTextField();
```

```
        txtgender = new JTextField();

        txtDisability = new JTextField();

        txtcontact = new JTextField();




        tblStudent = new JTable();

tblStudent.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);

        tblStudent.getSelectionModel().addListSelectionListener(e ->
selectStudent());


        JScrollPane scrollPane = new JScrollPane(tblStudent);


        btnAdd = new JButton("Add");

        btnModify = new JButton("Modify");

        btnDelete = new JButton("Delete");

        btnDisplay = new JButton("Display");


        JPanel panel = new JPanel(new GridBagLayout());

        GridBagConstraints gbc = new GridBagConstraints();

        gbc.gridx = 0;

        gbc.gridy = 0;

        gbc.anchor = GridBagConstraints.WEST;

        gbc.insets = new Insets(5, 5, 5, 5);


        panel.add(new JLabel("Student ID:"), gbc);

        gbc.gridy++;
```

```
panel.add(new JLabel("Name:"), gbc);

gbc.gridy++;

panel.add(new JLabel("Gender:"), gbc);

gbc.gridy++;

panel.add(new JLabel("Disability:"), gbc);

gbc.gridy++;

panel.add(new JLabel("conatct no:"), gbc);

gbc.gridy++;

gbc.gridx = 1;

gbc.gridy = 0;

gbc.fill = GridBagConstraints.HORIZONTAL;

gbc.weightx = 1;


panel.add(txtId, gbc);

gbc.gridy++;

panel.add(txtName, gbc);

gbc.gridy++;

panel.add(txtgender, gbc);

gbc.gridy++;

panel.add(txtDisability, gbc);

gbc.gridy++;

panel.add(txtcontact, gbc);

gbc.gridy++;




gbc.gridx = 0;
gbc.gridy++;

gbc.gridwidth = 2;

gbc.fill = GridBagConstraints.NONE;

gbc.anchor = GridBagConstraints.CENTER;

gbc.weightx = 0;


panel.add(btnAdd, gbc);

gbc.gridy++;

panel.add(btnModify, gbc);

gbc.gridy++;

panel.add(btnDelete, gbc);

gbc.gridy++;

panel.add(btnDisplay, gbc);


setLayout(new BorderLayout());

add(panel, BorderLayout.NORTH);

add(scrollPane, BorderLayout.CENTER);


btnAdd.addActionListener(e -> insertStudents());


btnModify.addActionListener(e -> modifyStudents());


btnDelete.addActionListener(e -> deleteStudents());


btnDisplay.addActionListener(e -> displayStudents());
```

```
        setTitle("Customer Table App");

        pack();

        setLocationRelativeTo(null);

        setVisible(true);

    }


    private void connectToDatabase() {

        String url = "jdbc:oracle:thin:@localhost:1521:xe";

        String username = "vinay";

        String password = "vinay";


        try {

            connection = DriverManager.getConnection(url, username, 
password);

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }


    private void insertStudents() {

        String id = txtId.getText();

        String name = txtName.getText();

        String gender = txtgender.getText();

        String Disability = txtDisability.getText();

        String contact = txtcontact.getTextt();


        try {
            String query = "INSERT INTO student (student_id, Name,
gender, Disability, contact_info) VALUES (?, ?, ?, ?, ?)";

            PreparedStatement statement =
connection.prepareStatement(query);

            statement.setString(1, id);

            statement.setString(2, name);



                statement.setString(3, gender);

            statement.setString(4, Disability);

            statement.setString(5, contact);



            statement.executeUpdate();



            clearFields();

            displayStudents();

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }



    private void modifyStudents() {

        int selectedRow = tblStudent.getSelectedRow();

        if (selectedRow >= 0) {

            String id = txtId.getText();

            String name = txtName.getText();

            String gender = txtgender.getText();

            String Disability = txtDisability.getText();
            String contact = txtcontact.getText();



            try {

                String query = "UPDATE student SET Name=?, gender=?, Disability=?, contact_info=? WHERE student_id=?";

                PreparedStatement statement = connection.prepareStatement(query);

                statement.setString(1, name);

                statement.setString(2, gender);

                statement.setString(3, Disability);

                statement.setString(4, contact);



                statement.setString(5, id);

                statement.executeUpdate();



                clearFields();

                displayStudents();

            } catch (SQLException e) {

                e.printStackTrace();

            }

        } else {

            JOptionPane.showMessageDialog(this, "Please select a student details to modify.");

        }

    }



    private void deleteStudents() {

        int selectedRow = tblStudent.getSelectedRow();
        if (selectedRow >= 0) {

            String id = tblStudent.getValueAt(selectedRow,
0).toString();



            int option = JOptionPane.showConfirmDialog(this, "Are
you sure you want to delete this customer?", "Confirmation",
JOptionPane.YES_NO_OPTION);

            if (option == JOptionPane.YES_OPTION) {

                try {

                    String query = "DELETE FROM student WHERE
student_id=?";

                    PreparedStatement statement =
connection.prepareStatement(query);

                    statement.setString(1, id);

                    statement.executeUpdate();



                    clearFields();

                    displayStudents();

                } catch (SQLException e) {

                    e.printStackTrace();

                }

            }

        } else {

            JOptionPane.showMessageDialog(this, "Please select a
student to delete.");

        }

    }



    private void displayStudents() {

        try {
            String query = "SELECT * FROM student";

            Statement statement = connection.createStatement();

            ResultSet resultSet = statement.executeQuery(query);



            List<Student> student = new ArrayList<>();

            while (resultSet.next()) {

                String id = resultSet.getString("student_id");

                String name = resultSet.getString("Name");

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

                String Disability =
resultSet.getString("Disability");

                String contact =
resultSet.getString("contact_info");




                      student.add(new Student(id, name, gender,
Disability, contact));

            }



            DefaultTableModel model = new DefaultTableModel();

            model.setColumnIdentifiers(new String[]{"ID", "Name",
"Gender", "Disability", "contact"});



            for (Student students : student) {

                model.addRow(new String[]{students.getId(),
students.getName(), students.getgender(),

                        students.getDisability(),
students.getcontact()});

            }
            tblStudent.setModel(model);

        } catch (SQLException e) {

            e.printStackTrace();

        }

    }


    private void selectStudent() {

        int selectedRow = tblStudent.getSelectedRow();

        if (selectedRow >= 0) {

            String id = tblStudent.getValueAt(selectedRow,
0).toString();

            String name = tblStudent.getValueAt(selectedRow,
1).toString();

            String address = tblStudent.getValueAt(selectedRow,
2).toString();

            String contactNo = tblStudent.getValueAt(selectedRow,
3).toString();

            String aadharNo = tblStudent.getValueAt(selectedRow,
4).toString();




            txtId.setText(id);

            txtName.setText(name);

            txtgender.setText(address);

            txtDisability.setText(contactNo);

            txtcontact.setText(aadharNo);



        }

    }
    private void clearFields() {
        txtId.setText("");
        txtName.setText("");
        txtgender.setText("");
        txtDisability.setText("");
        txtcontact.setText("");

    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(StudentTableGUI::new);
    }

    private class Student {
        private String id;
        private String name;
        private String gender;
        private String Disability;
        private String contact;


        public Student(String id, String name, String gender, String
Disability, String contact) {
            this.id = id;
            this.name = name;
            this.gender = gender;
            this.Disability = Disability;

            this.contact = contact;


        }


        public String getId() {

            return id;

        }


        public String getName() {

            return name;

        }


        public String getgender() {

            return gender;

        }


        public String getDisability() {

            return Disability;

        }


        public String getcontact() {

            return contact;

        }

}

}
```

Faculty Table:



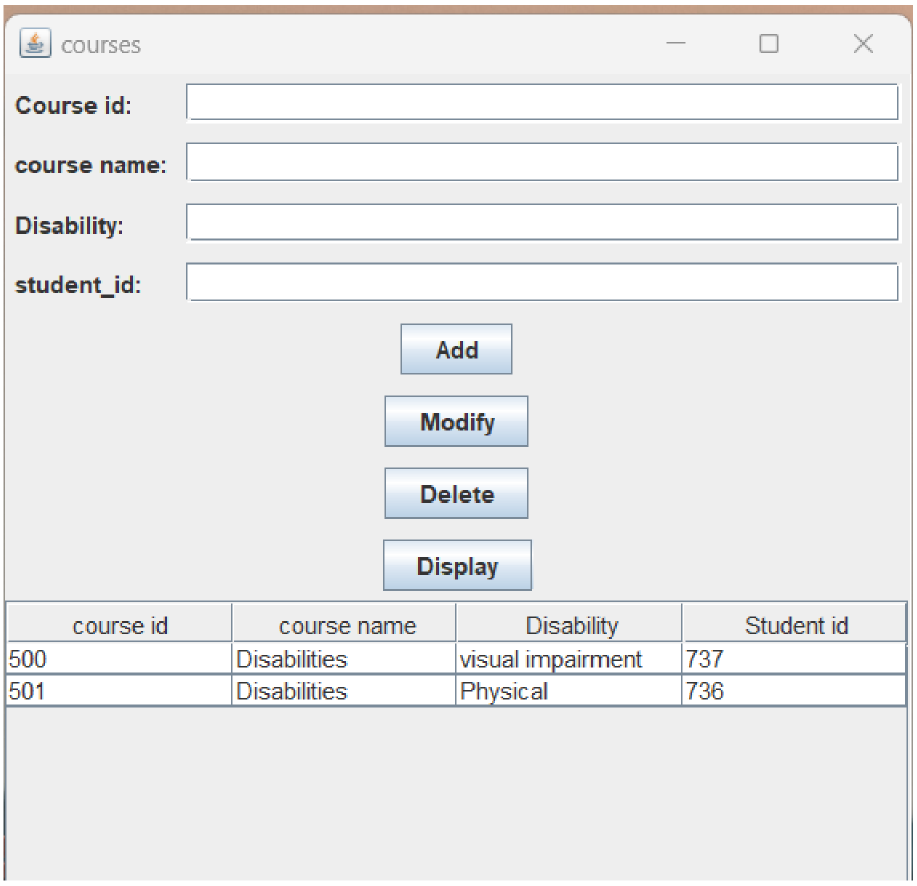
Teacher id:
Name:
contact no:
Specialization:

Teacher id	Name	contact info	specialization
100	Kavya	12345	Doctor
101	Divya	123456789	Engineering
105	Teja	273488	Doctor

```
package EducationSystem;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class Faculty extends JFrame {
    /**
       *
       */
      private static final long serialVersionUID = 1L;
      private JTextField txtTeacher, txtName, txtcontact,
txtspecialization;
    private JTable tblFaculty;
    private JButton btnAdd, btnModify, btnDelete, btnDisplay;

    private Connection connection;

    public Faculty() {
        initializeUI();
        connectToDatabase();
        displayFaculty();
    }

    private void initializeUI() {
        txtTeacher = new JTextField();
        txtName = new JTextField();
        txtcontact = new JTextField();
        txtspecialization = new JTextField();

        tblFaculty = new JTable();
        tblFaculty.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        tblFaculty.getSelectionModel().addListSelectionListener(e ->
selectAccount());

        JScrollPane scrollPane = new JScrollPane(tblFaculty);

        btnAdd = new JButton("Add");
        btnModify = new JButton("Modify");
        btnDelete = new JButton("Delete");
        btnDisplay = new JButton("Display");

        JPanel panel = new JPanel(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.anchor = GridBagConstraints.WEST;
        gbc.insets = new Insets(5, 5, 5, 5);

        panel.add(new JLabel("Teacher id:"), gbc);
        gbc.gridy++;
        panel.add(new JLabel("Name:"), gbc);
        gbc.gridy++;
        panel.add(new JLabel("contact no:"), gbc);
        gbc.gridy++;
        panel.add(new JLabel("Specialization:"), gbc);

        gbc.gridx = 1;
        gbc.gridy = 0;
        gbc.fill = GridBagConstraints.HORIZONTAL;
        gbc.weightx = 1;

        panel.add(txtTeacher, gbc);
        gbc.gridy++;
        panel.add(txtName, gbc);
        gbc.gridy++;
        panel.add(txtcontact, gbc);
        gbc.gridy++;
        panel.add(txtspecialization, gbc);

        gbc.gridx = 0;
        gbc.gridy++;
        gbc.gridwidth = 2;
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.CENTER;
        gbc.weightx = 0;

        panel.add(btnAdd, gbc);
        gbc.gridy++;
        panel.add(btnModify, gbc);
        gbc.gridy++;
        panel.add(btnDelete, gbc);
        gbc.gridy++;
        panel.add(btnDisplay, gbc);

        setLayout(new BorderLayout());
        add(panel, BorderLayout.NORTH);
        add(scrollPane, BorderLayout.CENTER);

        btnAdd.addActionListener(e -> insertFaculty());

        btnModify.addActionListener(e -> modifyFaculty());

        btnDelete.addActionListener(e -> deleteFaculty());

        btnDisplay.addActionListener(e -> displayFaculty());

        setTitle("Faculty Table");
        pack();
        setLocationRelativeTo(null);
        setVisible(true);
    }

    private void connectToDatabase() {
        String url = "jdbc:oracle:thin:@localhost:1521:xe";
        String username = "vinay";
        String password = "vinay";

        try {
            connection = DriverManager.getConnection(url, username,
password);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private void insertFaculty() {
        String Teacherid = txtTeacher.getText();
        String Name = txtName.getText();
        String contact = txtcontact.getText();
        String specialization = txtspecialization.getText();

        try {
            String query = "INSERT INTO Faculty (Teacher_id, Name,
contact_info, specialization) VALUES (?, ?, ?, ?)";
            PreparedStatement statement =
connection.prepareStatement(query);
            statement.setString(1, Teacherid);
            statement.setString(2, Name);
            statement.setString(3, contact);
            statement.setString(4, specialization);
            statement.executeUpdate();

            clearFields();
            displayFaculty();
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "please enter the values");

        }
    }

    private void modifyFaculty() {
        int selectedRow = tblFaculty.getSelectedRow();
        if (selectedRow >= 0) {
            String teacher = txtTeacher.getText();
            String name = txtName.getText();
            String contact = txtcontact.getText();
            String specialization = txtspecialization.getText();

            try {
                String query = "UPDATE Faculty SET Name=?, contact_info=?,
specialization=? WHERE Teacher_id=?";
                PreparedStatement statement =
connection.prepareStatement(query);
                statement.setString(1, name);
                statement.setString(2, contact);
                statement.setString(3, specialization);
                statement.setString(4, teacher);
                statement.executeUpdate();

                clearFields();
                displayFaculty();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        } else {
            JOptionPane.showMessageDialog(this, "Please select an Teacher
to modify.");
        }
    }

    private void deleteFaculty() {
        int selectedRow = tblFaculty.getSelectedRow();
        if (selectedRow >= 0) {
            String Teacher = tblFaculty.getValueAt(selectedRow,
0).toString();

            int option = JOptionPane.showConfirmDialog(this, "Are you sure
you want to delete this Information?", "Confirmation",
JOptionPane.YES_NO_OPTION);
            if (option == JOptionPane.YES_OPTION) {
                try {
                    String query = "DELETE FROM Faculty WHERE
Teacher_id=?";
                    PreparedStatement statement =
connection.prepareStatement(query);
                    statement.setString(1, Teacher);
                    statement.executeUpdate();

                    clearFields();
                    displayFaculty();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        } else {
            JOptionPane.showMessageDialog(this, "Please select an Teacher
id to delete.");
        }
    }

    private void displayFaculty() {
        try {
            String query = "SELECT * FROM Faculty";
            Statement statement = connection.createStatement();
            ResultSet resultSet = statement.executeQuery(query);

            List<Faculty1> Facul = new ArrayList<>();
            while (resultSet.next()) {
                String Teacher = resultSet.getString("Teacher_id");
                String Name = resultSet.getString("Name");
                String contact = resultSet.getString("contact_info");
                String specialization =
resultSet.getString("specialization");
                Facul.add(new Faculty1(Teacher, Name, contact,
specialization));
            }

            DefaultTableModel model = new DefaultTableModel();
            model.setColumnIdentifiers(new String[]{"Teacher id", "Name",
"contact info", "specialization"});

            for (Faculty1 facul : Facul) {
                model.addRow(new String[]{  facul.getTeacher(),
facul.getName(), facul.getcontact(), facul.getspecialization()});
            }

            tblFaculty.setModel(model);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private void selectAccount() {
        int selectedRow = tblFaculty.getSelectedRow();
        if (selectedRow >= 0) {
            String Teacher = tblFaculty.getValueAt(selectedRow,
0).toString();
            String Name = tblFaculty.getValueAt(selectedRow, 1).toString();
            String contact = tblFaculty.getValueAt(selectedRow,
2).toString();
            String specialization = tblFaculty.getValueAt(selectedRow,
3).toString();

            txtTeacher.setText(Teacher);
            txtName.setText(Name);
            txtcontact.setText(contact);
            txtspecialization.setText(specialization);
        }
    }

    private void clearFields() {
        txtTeacher.setText("");
        txtName.setText("");
        txtcontact.setText("");
        txtspecialization.setText("");
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(Faculty::new);
    }

    private class Faculty1{
        private String Teacher;
        private String Name;
        private String contact;
       private String specialization;
```

```
        public Faculty1(String Teacher, String Name, String contact, String
specialization) {
            this.Teacher = Teacher;
            this.Name= Name;
            this.contact = contact;
            this.specialization = specialization;
        }

        public String getTeacher() {
            return Teacher;
        }

        public String getName() {
            return Name;
        }

        public String getcontact() {
            return contact;
        }

        public String getspecialization() {
            return specialization;
        }
    }
}
```

Course Table:



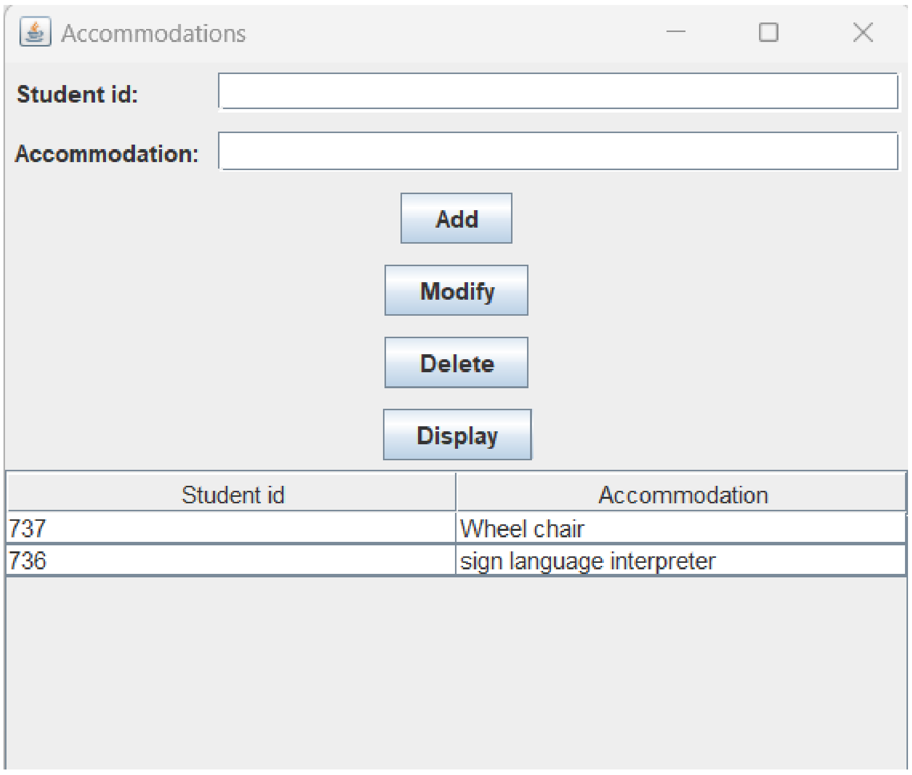
course id	course name	Disability	Student id
500	Disabilities	visual impairment	737
501	Disabilities	Physical	736

```
package EducationSystem;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class Course extends JFrame {
    /**
       *
       */
      private static final long serialVersionUID = 1L;
      private JTextField txtcourse, txtname, txtdis,txtsid;
    private JTable tblcourse;
    private JButton btnAdd, btnModify, btnDelete, btnDisplay;

    private Connection connection;

    public Course() {
        initializeUI();
        connectToDatabase();
        displaycourse();
    }

    private void initializeUI() {
        txtcourse = new JTextField();
        txtname = new JTextField();
        txtdis = new JTextField();
        txtsid=new JTextField();

        tblcourse = new JTable();
        tblcourse.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        tblcourse.getSelectionModel().addListSelectionListener(e ->
selectAtm());

        JScrollPane scrollPane = new JScrollPane(tblcourse);

        btnAdd = new JButton("Add");
        btnModify = new JButton("Modify");
        btnDelete = new JButton("Delete");
        btnDisplay = new JButton("Display");

        JPanel panel = new JPanel(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.anchor = GridBagConstraints.WEST;
        gbc.insets = new Insets(5, 5, 5, 5);

        panel.add(new JLabel("Course id:"), gbc);
        gbc.gridy++;
        panel.add(new JLabel("course name:"), gbc);
        gbc.gridy++;
        panel.add(new JLabel("Disability:"), gbc);
        gbc.gridy++;
        panel.add(new JLabel("student_id:"), gbc);
        gbc.gridx = 1;
        gbc.gridy = 0;
        gbc.fill = GridBagConstraints.HORIZONTAL;
        gbc.weightx = 1;

        panel.add(txtcourse, gbc);
        gbc.gridy++;
        panel.add(txtname, gbc);
        gbc.gridy++;
        panel.add(txtdis, gbc);
        gbc.gridy++;
        panel.add(txtsid,gbc);


        gbc.gridx = 0;
        gbc.gridy++;
        gbc.gridwidth = 2;
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.CENTER;
        gbc.weightx = 0;

        panel.add(btnAdd, gbc);
        gbc.gridy++;
        panel.add(btnModify, gbc);
        gbc.gridy++;
        panel.add(btnDelete, gbc);
        gbc.gridy++;
        panel.add(btnDisplay, gbc);

        setLayout(new BorderLayout());
        add(panel, BorderLayout.NORTH);
        add(scrollPane, BorderLayout.CENTER);

        btnAdd.addActionListener(e -> insertcourse());

        btnModify.addActionListener(e -> modifycourse());

        btnDelete.addActionListener(e -> deletecourse());

        btnDisplay.addActionListener(e -> displaycourse());

        setTitle("courses");
        pack();
        setLocationRelativeTo(null);
        setVisible(true);
    }

    private void connectToDatabase() {
        String url = "jdbc:oracle:thin:@localhost:1521:xe";
        String username = "vinay";
        String password = "vinay";

        try {
            connection = DriverManager.getConnection(url, username,
password);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    private void insertcourse() {
        String course = txtcourse.getText();
        String name = txtname.getText();
        String dis = txtdis.getText();
        String sid=txtsid.getText();

        try {
            String query = "INSERT INTO course (course_id, course_name,
Disability_type,student_id) VALUES (?, ?, ?,?)";
            PreparedStatement statement =
connection.prepareStatement(query);
            statement.setString(1, course);
            statement.setString(2, name);
            statement.setString(3, dis);
            statement.setString(4,sid);
            statement.executeUpdate();

            clearFields();
            displaycourse();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private void modifycourse() {
        int selectedRow = tblcourse.getSelectedRow();
        if (selectedRow >= 0) {
            String course = txtcourse.getText();
            String name = txtname.getText();
            String dis = txtdis.getText();
            String sid=txtsid.getText();

            try {
                String query = "UPDATE course SET
course_name=?,Disability_type=?,student_id=? WHERE course_id=?";
                PreparedStatement statement =
connection.prepareStatement(query);
                statement.setString(1, name);
                statement.setString(2, dis);
                statement.setString(3, sid);
                statement.setString(4,course);
                statement.executeUpdate();

                clearFields();
                displaycourse();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        } else {
            JOptionPane.showMessageDialog(this, "Please select an course to
modify.");
        }
    }

    private void deletecourse() {
        int selectedRow = tblcourse.getSelectedRow();
        if (selectedRow >= 0) {
            String course = tblcourse.getValueAt(selectedRow,
0).toString();
```

```
            int option = JOptionPane.showConfirmDialog(this, "Are you sure
you want to delete this course?", "Confirmation",
JOptionPane.YES_NO_OPTION);
            if (option == JOptionPane.YES_OPTION) {
                try {
                    String query = "DELETE FROM course WHERE course_id=?";
                    PreparedStatement statement =
connection.prepareStatement(query);
                    statement.setString(1, course);
                    statement.executeUpdate();

                    clearFields();
                    displaycourse();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        } else {
            JOptionPane.showMessageDialog(this, "Please select an course to
delete.");
        }
    }

    private void displaycourse() {
        try {
            String query = "SELECT * FROM course";
            Statement statement = connection.createStatement();
            ResultSet resultSet = statement.executeQuery(query);

            List<course> courses = new ArrayList<>();
            while (resultSet.next()) {
                String course = resultSet.getString("course_id");
                String name = resultSet.getString("course_name");
                String dis = resultSet.getString("Disability_type");
                String sid = resultSet.getString("student_id");

                courses.add(new course(course, name, dis,sid));
            }

            DefaultTableModel model = new DefaultTableModel();
            model.setColumnIdentifiers(new String[]{"course id", "course
name", "Disability","Student id"});

            for (course cou : courses) {
                model.addRow(new String[]{cou.getcourse(), cou.getname(),
cou.getdis(),cou.getsid()});
            }

            tblcourse.setModel(model);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private void selectAtm() {
        int selectedRow = tblcourse.getSelectedRow();
        if (selectedRow >= 0) {
            String course = tblcourse.getValueAt(selectedRow,
0).toString();
            String name = tblcourse.getValueAt(selectedRow, 1).toString();
            String dis = tblcourse.getValueAt(selectedRow, 2).toString();
```

```java
            String sid = tblcourse.getValueAt(selectedRow, 3).toString();


            txtcourse.setText(course);
            txtname.setText(name);
            txtdis.setText(dis);
            txtsid.setText(sid);
        }
    }

    private void clearFields() {
        txtcourse.setText("");
        txtname.setText("");
        txtdis.setText("");
        txtsid.setText("");
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(Course::new);
    }

    private class course {
        private String course;
        private String name;
        private String dis;
        private String sid;

        public course(String course, String name, String dis,String sid) {
            this.course = course;
            this.name = name;
            this.dis = dis;
            this.sid=sid;
        }

        public String getcourse() {
            return course;
        }

        public String getname() {
            return name;
        }

        public String getdis() {
            return dis;
        }
        public String getsid() {
            return sid;
        }
    }
}
```

Accommodations Table:



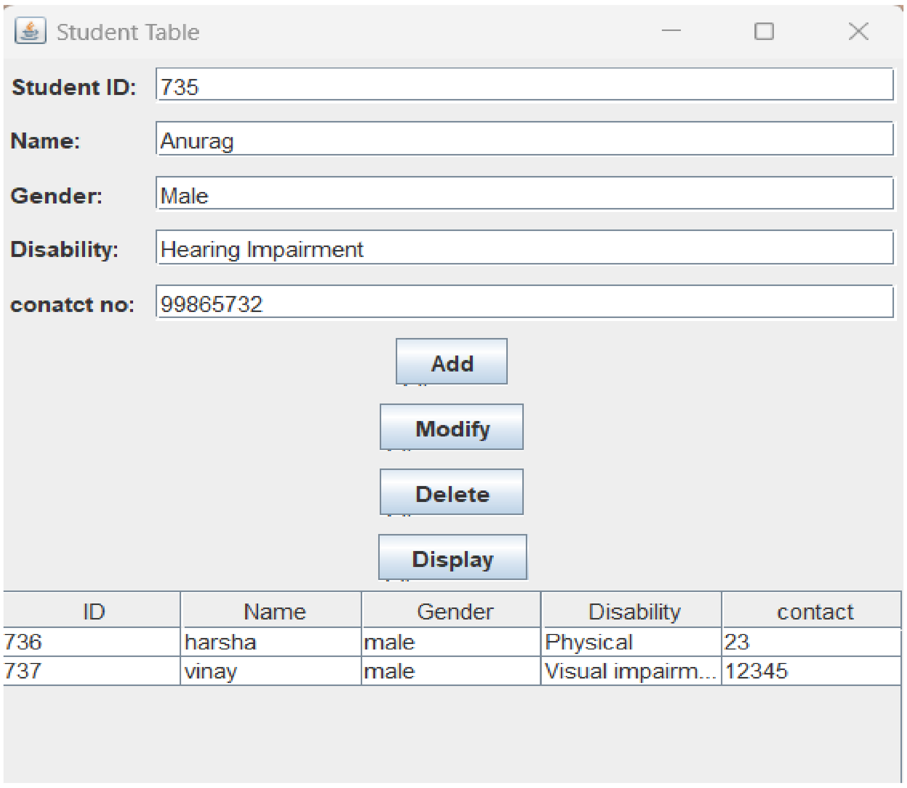
Student id	Accommodation
737	Wheel chair
736	sign language interpreter

```java
package EducationSystem;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class Accommodations extends JFrame {
    /**
       *
       */
      private static final long serialVersionUID = 1L;
      private JTextField txtsid, txtacc;
    private JTable tblAcc;
    private JButton btnAdd, btnModify, btnDelete, btnDisplay;

    private Connection connection;

    public Accommodations() {
        initializeUI();
        connectToDatabase();
        displayAcc();
    }

    private void initializeUI() {
        txtsid = new JTextField();
        txtacc = new JTextField();


        tblAcc = new JTable();
        tblAcc.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        tblAcc.getSelectionModel().addListSelectionListener(e ->
selectAtm());

        JScrollPane scrollPane = new JScrollPane(tblAcc);

        btnAdd = new JButton("Add");
        btnModify = new JButton("Modify");
        btnDelete = new JButton("Delete");
        btnDisplay = new JButton("Display");

        JPanel panel = new JPanel(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.anchor = GridBagConstraints.WEST;
        gbc.insets = new Insets(5, 5, 5, 5);

        panel.add(new JLabel("Student id:"), gbc);
        gbc.gridy++;
        panel.add(new JLabel("Accommodation:"), gbc);


        gbc.gridx = 1;
        gbc.gridy = 0;
        gbc.fill = GridBagConstraints.HORIZONTAL;
        gbc.weightx = 1;

        panel.add(txtsid, gbc);
        gbc.gridy++;
        panel.add(txtacc, gbc);


        gbc.gridx = 0;
        gbc.gridy++;
        gbc.gridwidth = 2;
        gbc.fill = GridBagConstraints.NONE;
        gbc.anchor = GridBagConstraints.CENTER;
        gbc.weightx = 0;

        panel.add(btnAdd, gbc);
        gbc.gridy++;
        panel.add(btnModify, gbc);
        gbc.gridy++;
        panel.add(btnDelete, gbc);
        gbc.gridy++;
        panel.add(btnDisplay, gbc);

        setLayout(new BorderLayout());
        add(panel, BorderLayout.NORTH);
        add(scrollPane, BorderLayout.CENTER);

        btnAdd.addActionListener(e -> insertAcc());

        btnModify.addActionListener(e -> modifyAcc());

        btnDelete.addActionListener(e -> deleteAcc());

        btnDisplay.addActionListener(e -> displayAcc());

        setTitle("Accommodations");
        pack();
        setLocationRelativeTo(null);
        setVisible(true);
    }

    private void connectToDatabase() {
        String url = "jdbc:oracle:thin:@localhost:1521:xe";
        String username = "vinay";
        String password = "vinay";

        try {
            connection = DriverManager.getConnection(url, username, 
password);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private void insertAcc() {
        String student = txtsid.getText();
        String accommo = txtacc.getText();


        try {
            String query = "INSERT INTO Accommodations (student_id, 
Accommodation_type) VALUES (?, ?)";
            PreparedStatement statement = 
connection.prepareStatement(query);
            statement.setString(1,student );
            statement.setString(2, accommo);

            statement.executeUpdate();

            clearFields();
            displayAcc();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private void modifyAcc() {
        int selectedRow = tblAcc.getSelectedRow();
        if (selectedRow >= 0) {
            String stuid = txtsid.getText();
            String acc = txtacc.getText();


            try {
                String query = "UPDATE Accommodations SET 
Accommodation_type=? WHERE student_id=?";
                PreparedStatement statement = 
connection.prepareStatement(query);
                statement.setString(1, acc);
                statement.setString(2, stuid);

                statement.executeUpdate();

                clearFields();
                displayAcc();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        } else {
            JOptionPane.showMessageDialog(this, "Please select an student
to modify.");
        }
    }

    private void deleteAcc() {
        int selectedRow = tblAcc.getSelectedRow();
        if (selectedRow >= 0) {
            String acc = tblAcc.getValueAt(selectedRow, 0).toString();

            int option = JOptionPane.showConfirmDialog(this, "Are you sure
you want to delete this ?", "Confirmation", JOptionPane.YES_NO_OPTION);
            if (option == JOptionPane.YES_OPTION) {
                try {
                    String query = "DELETE FROM Accommodations WHERE
student_id=?";
                    PreparedStatement statement =
connection.prepareStatement(query);
                    statement.setString(1, acc);
                    statement.executeUpdate();

                    clearFields();
                    displayAcc();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        } else {
            JOptionPane.showMessageDialog(this, "Please select an student
to delete.");
        }
    }

    private void displayAcc() {
        try {
            String query = "SELECT * FROM Accommodations";
            Statement statement = connection.createStatement();
            ResultSet resultSet = statement.executeQuery(query);

            List<Accommodations1> accs = new ArrayList<>();
            while (resultSet.next()) {
                String stuId = resultSet.getString("student_id");
                String acc = resultSet.getString("Accommodation_type");

                accs.add(new Accommodations1(stuId, acc));
            }

            DefaultTableModel model = new DefaultTableModel();
            model.setColumnIdentifiers(new String[]{"Student id",
"Accommodation"});

            for (Accommodations1 acc : accs) {
                model.addRow(new String[]{acc.getstdId(), acc.getacc()});
            }

            tblAcc.setModel(model);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
```

```
        private void selectAtm() {
          int selectedRow = tblAcc.getSelectedRow();
          if (selectedRow >= 0) {
              String stuId = tblAcc.getValueAt(selectedRow, 0).toString();
              String acc = tblAcc.getValueAt(selectedRow, 1).toString();


              txtsid.setText(stuId);
              txtacc.setText(acc);

          }
      }

      private void clearFields() {
          txtsid.setText("");
          txtacc.setText("");

      }

      public static void main(String[] args) {
          SwingUtilities.invokeLater(Accommodations::new);
      }

      private class Accommodations1 {
          private String stdId;
          private String acc;


          public Accommodations1(String stdId, String acc) {
              this.stdId = stdId;
              this.acc = acc;

          }

          public String getstdId() {
              return stdId;
          }

          public String getacc() {
              return acc;
          }


      }
  }
```

TESTING

STUDENT PAGE

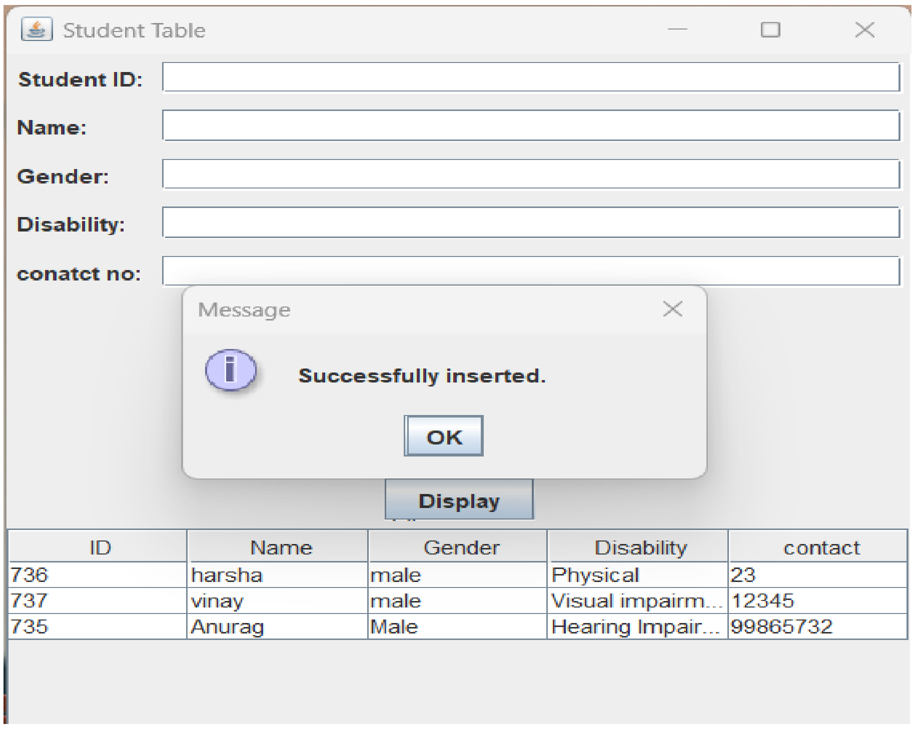


Student Table

Student ID:	735
Name:	Anurag
Gender:	Male
Disability:	Hearing Impairment
conatct no:	99865732

ID	Name	Gender	Disability	contact
736	harsha	male	Physical	23
737	vinay	male	Visual impairm...	12345

After Insertion:



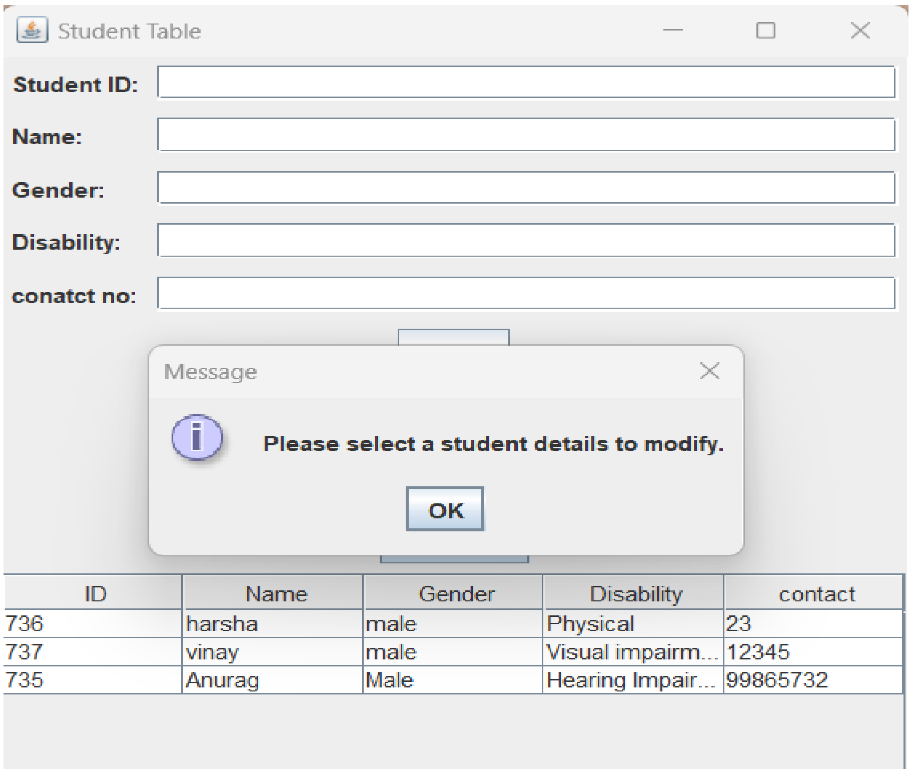
Student Table

Student ID:	
Name:	
Gender:	
Disability:	
conatct no:	

Message

Successfully inserted.

ID	Name	Gender	Disability	contact
736	harsha	male	Physical	23
737	vinay	male	Visual impairm...	12345
735	Anurag	Male	Hearing Impair...	99865732


Student Table

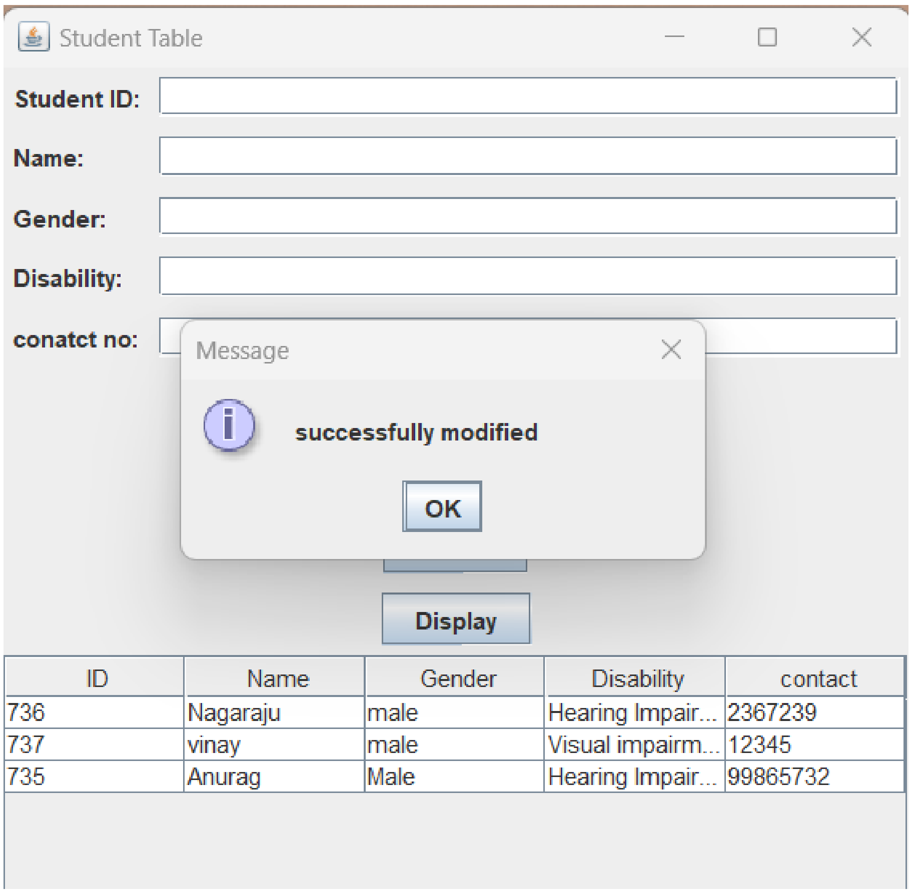
Student ID:	
Name:	
Gender:	
Disability:	
conatct no:	

Message

Please select a student details to modify.

ID	Name	Gender	Disability	contact
736	harsha	male	Physical	23
737	vinay	male	Visual impairm...	12345
735	Anurag	Male	Hearing Impair...	99865732

After Modifying

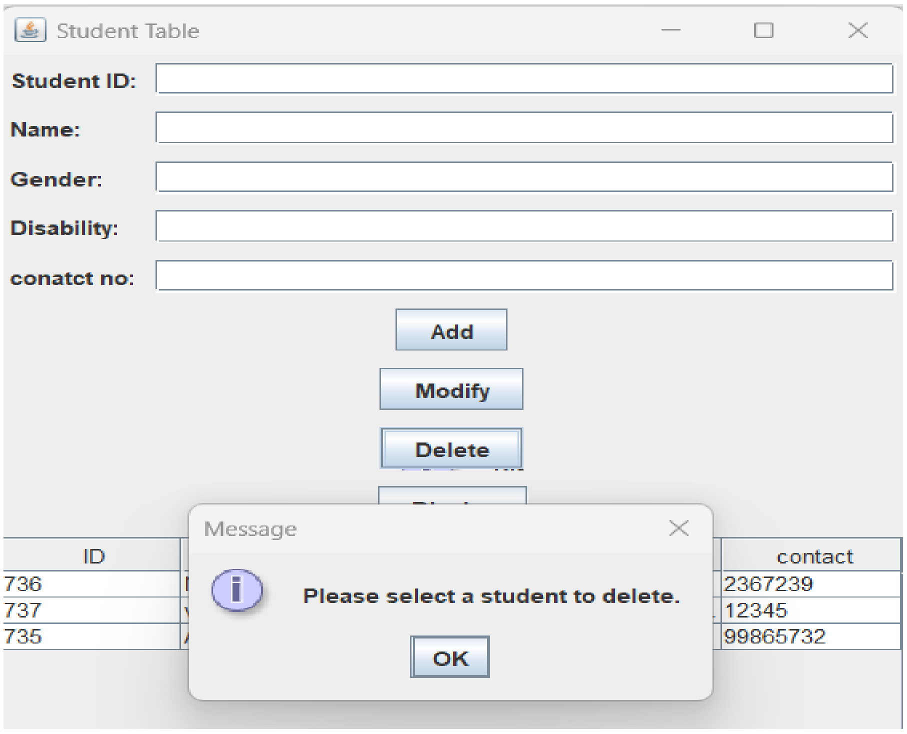

Student Table

Student ID:	
Name:	
Gender:	
Disability:	
conatct no:	

Message

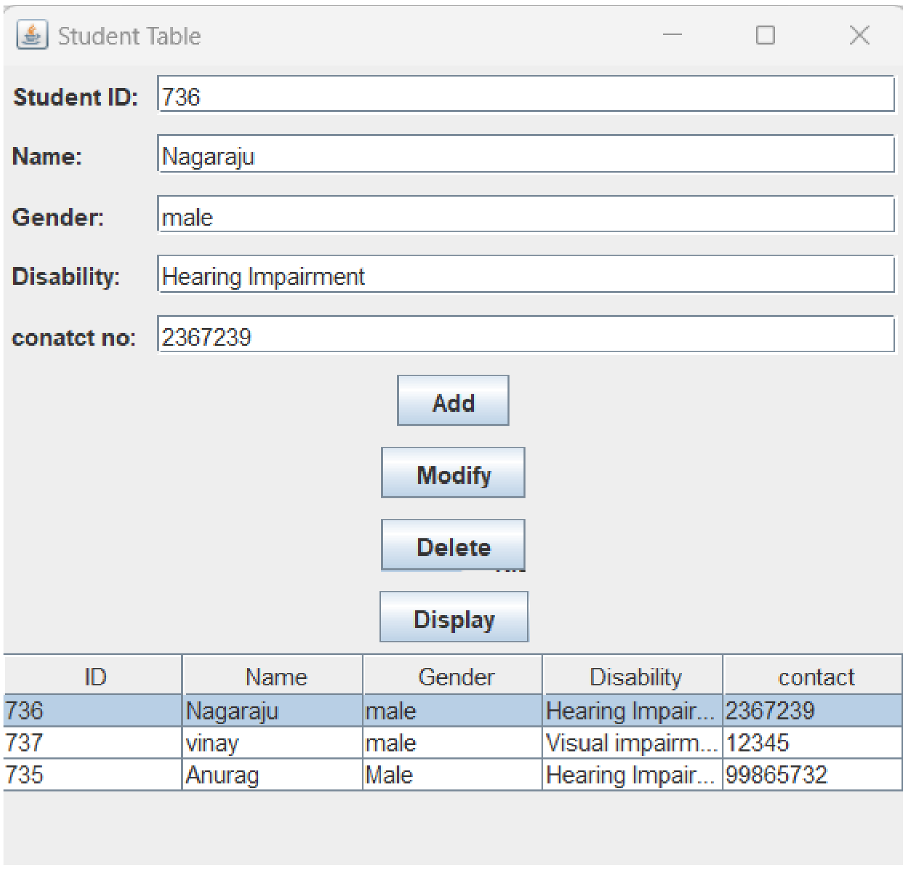
successfully modified

ID	Name	Gender	Disability	contact
736	Nagaraju	male	Hearing Impair...	2367239
737	vinay	male	Visual impairm...	12345
735	Anurag	Male	Hearing Impair...	99865732



Please select a student to delete.

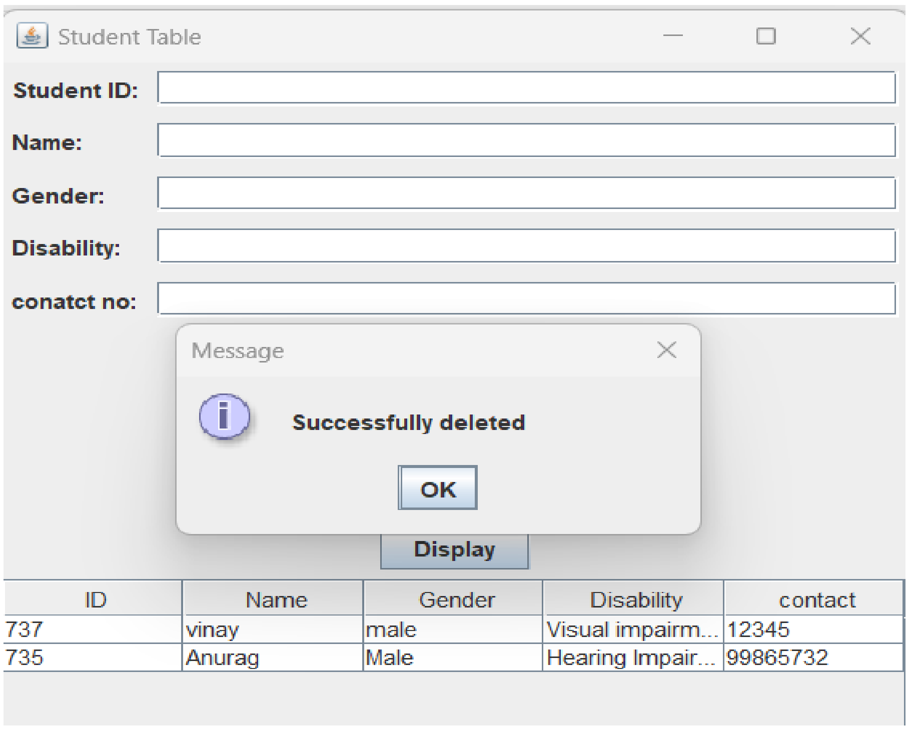
ID	contact
736	2367239
737	12345
735	99865732



Student ID: 736
Name: Nagaraju
Gender: male
Disability: Hearing Impairment
conatct no: 2367239

ID	Name	Gender	Disability	contact
736	Nagaraju	male	Hearing Impair...	2367239
737	vinay	male	Visual impairm...	12345
735	Anurag	Male	Hearing Impair...	99865732

After deleting:

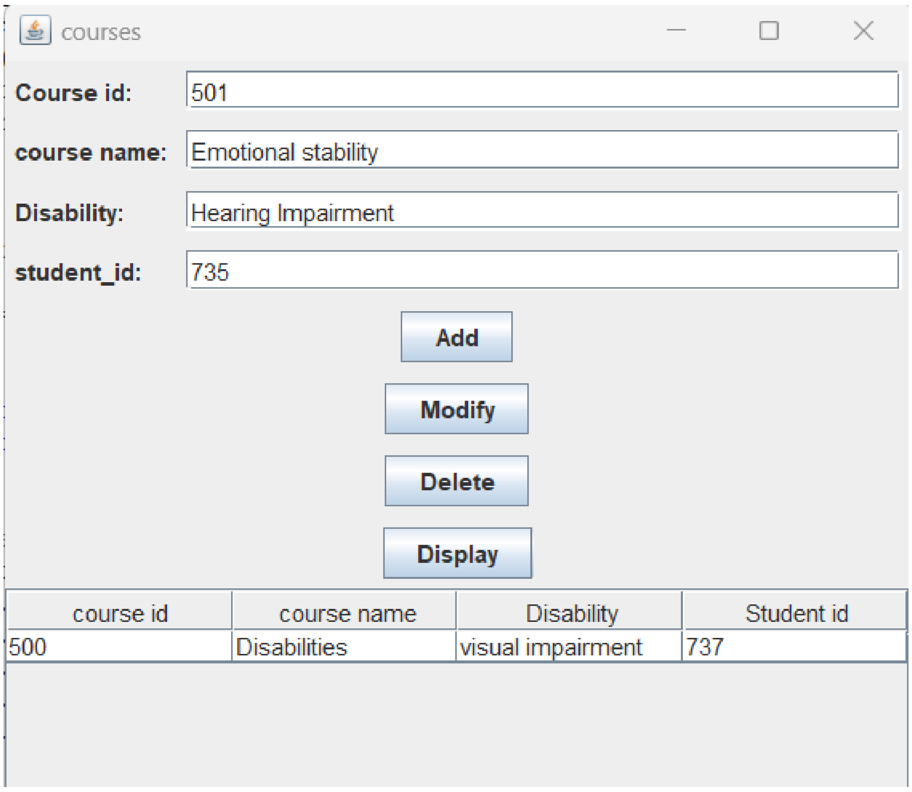


Student ID:
Name:
Gender:
Disability:
conatct no:

Successfully deleted

ID	Name	Gender	Disability	contact
737	vinay	male	Visual impairm...	12345
735	Anurag	Male	Hearing Impair...	99865732

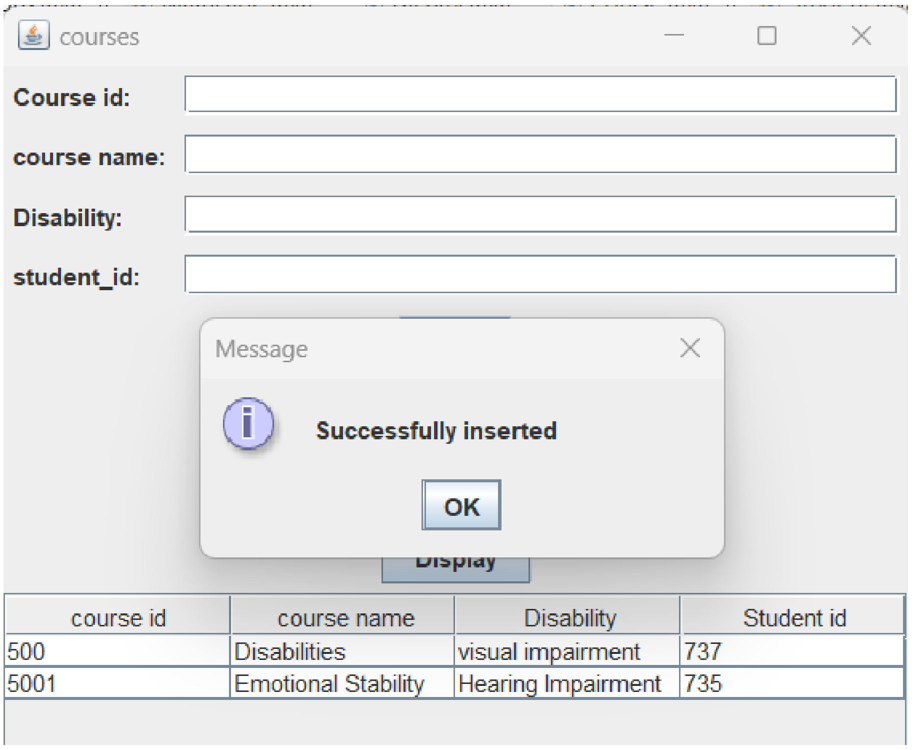
COURSE TABLE:



Course id:	501
course name:	Emotional stability
Disability:	Hearing Impairment
student_id:	735

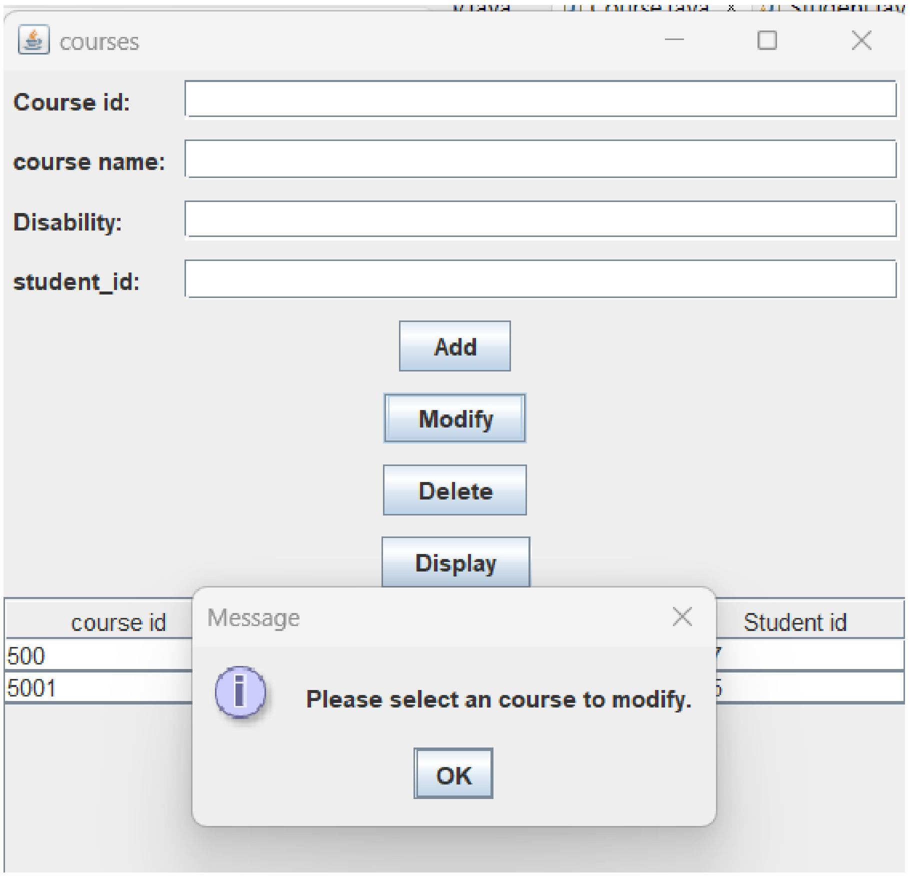
course id	course name	Disability	Student id
500	Disabilities	visual impairment	737

After insertion:



Successfully inserted

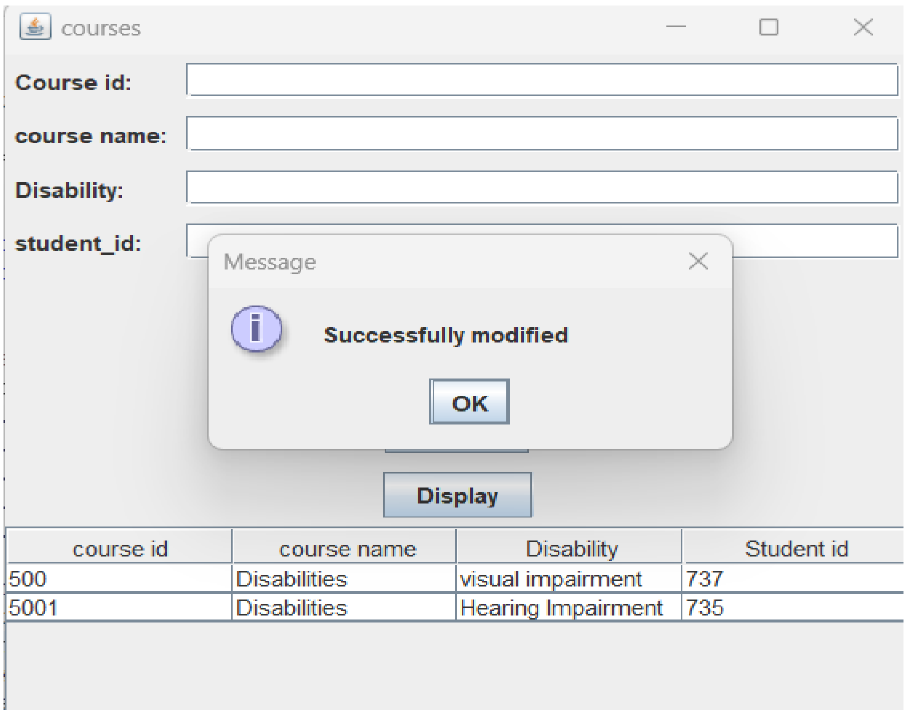
course id	course name	Disability	Student id
500	Disabilities	visual impairment	737
5001	Emotional Stability	Hearing Impairment	735



Please select an course to modify.

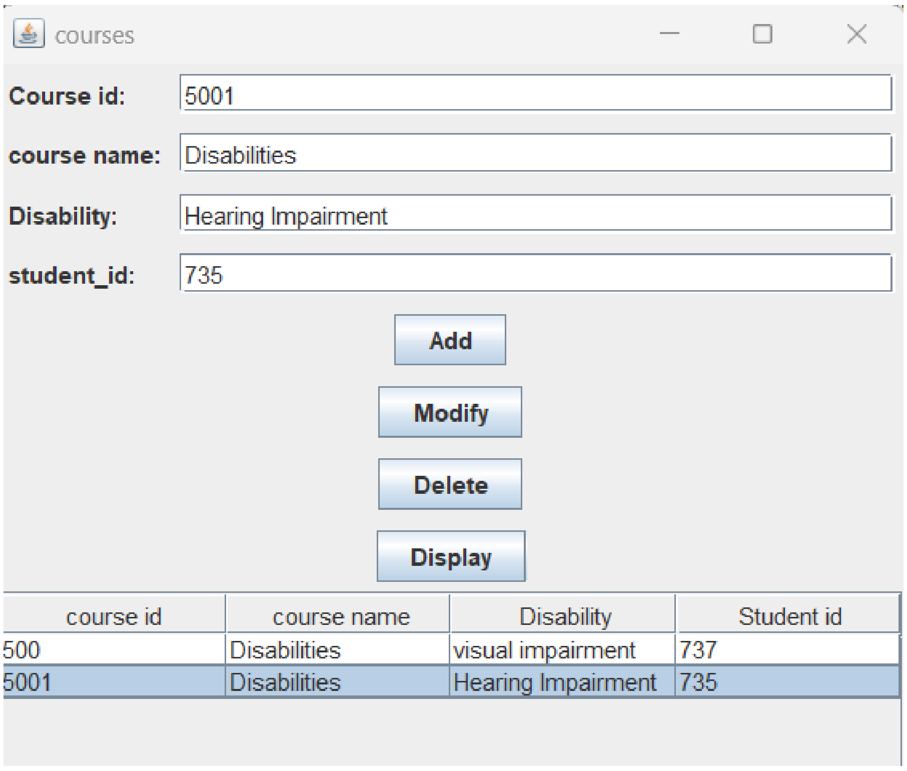
course id	Student id
500	7
5001	5

After modifying:



Successfully modified

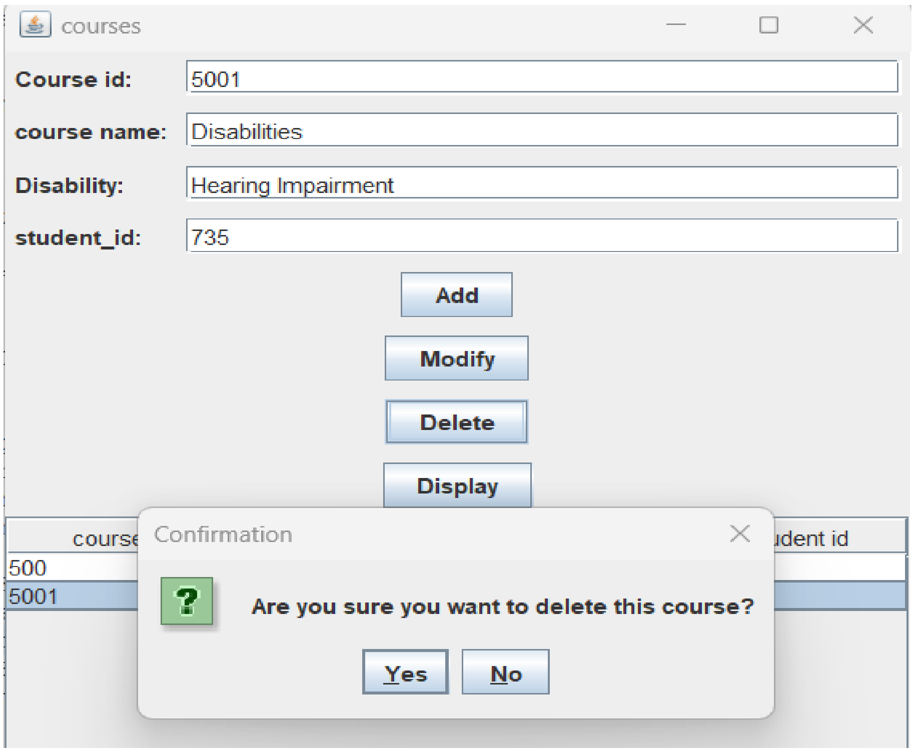
course id	course name	Disability	Student id
500	Disabilities	visual impairment	737
5001	Disabilities	Hearing Impairment	735



Course id:	5001
course name:	Disabilities
Disability:	Hearing Impairment
student_id:	735

course id	course name	Disability	Student id
500	Disabilities	visual impairment	737
5001	Disabilities	Hearing Impairment	735

After deleting:



Course id: 5001

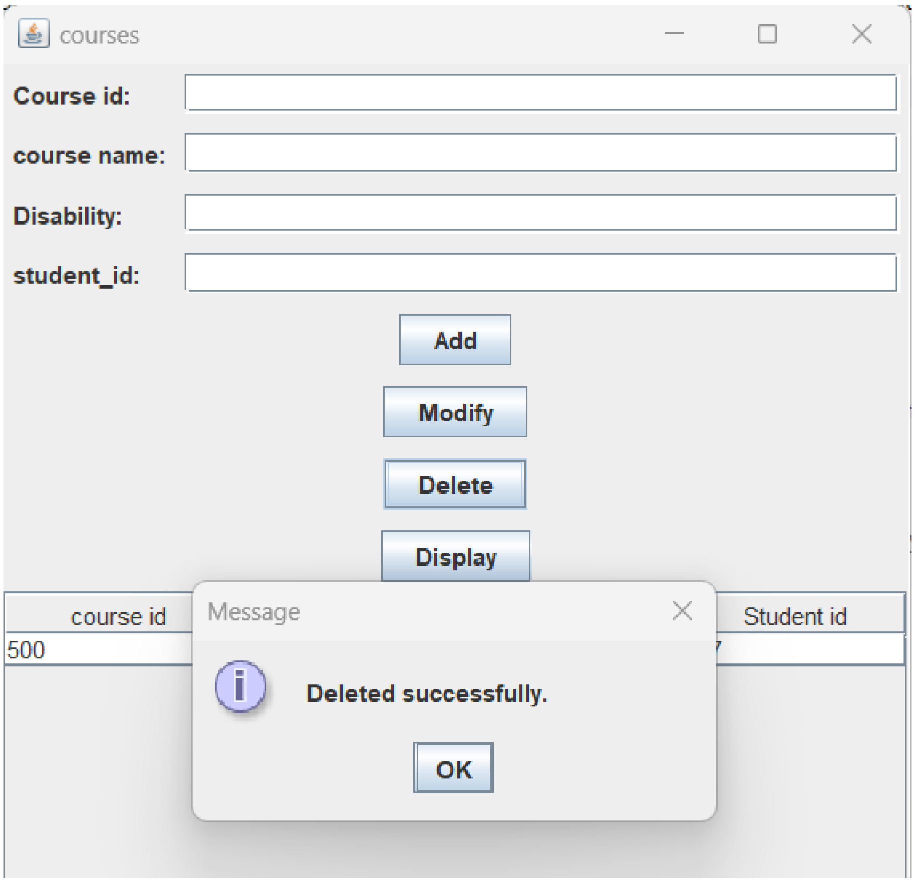
course name: Disabilities

Disability: Hearing Impairment

student_id: 735

course	udent id
500	
5001	

Are you sure you want to delete this course?



Course id:

course name:

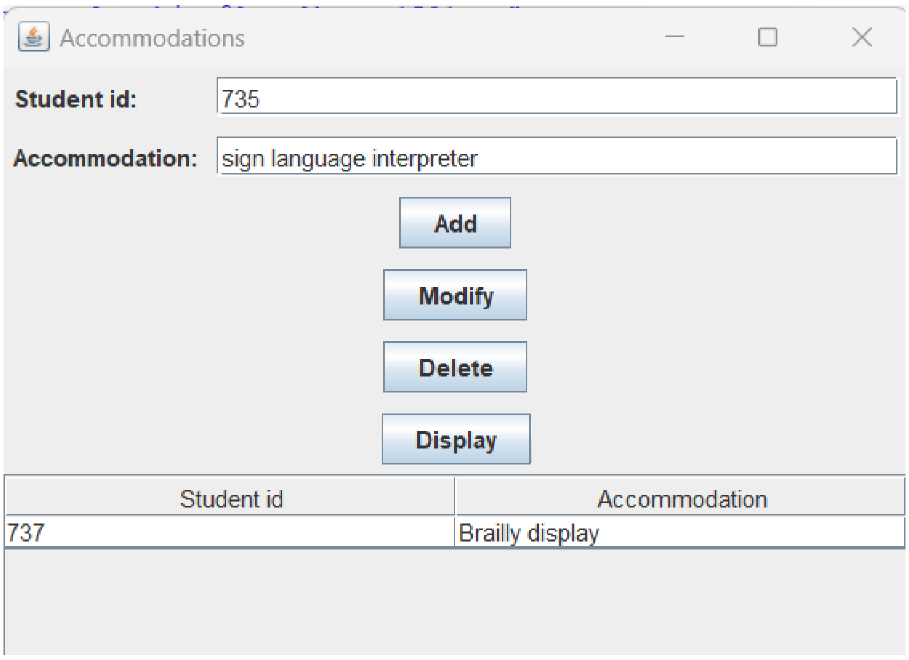
Disability:

student_id:

course id	Student id
500	7

Deleted successfully.

Accommodations Table:

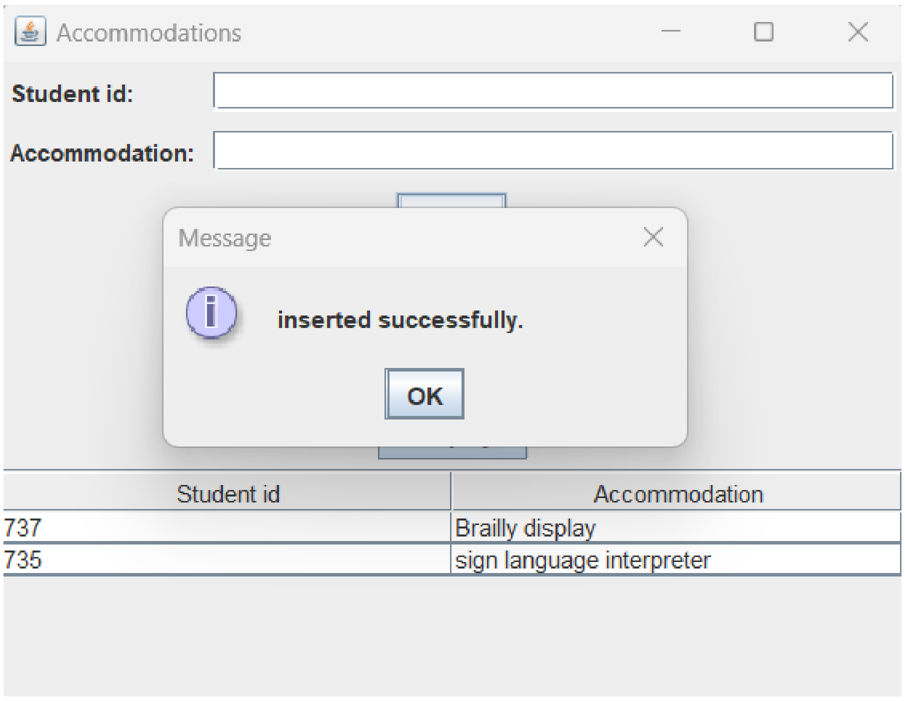


Student id: 735

Accommodation: sign language interpreter

Student id	Accommodation
737	Brailly display

After insertion:

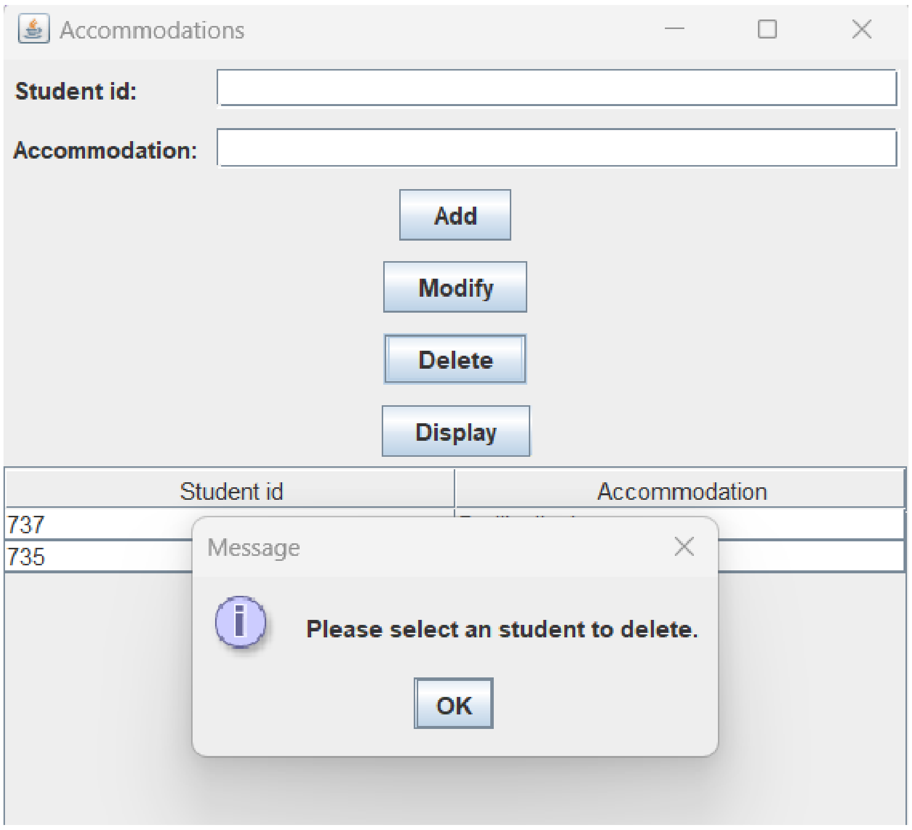


Student id:

Accommodation:

inserted successfully.

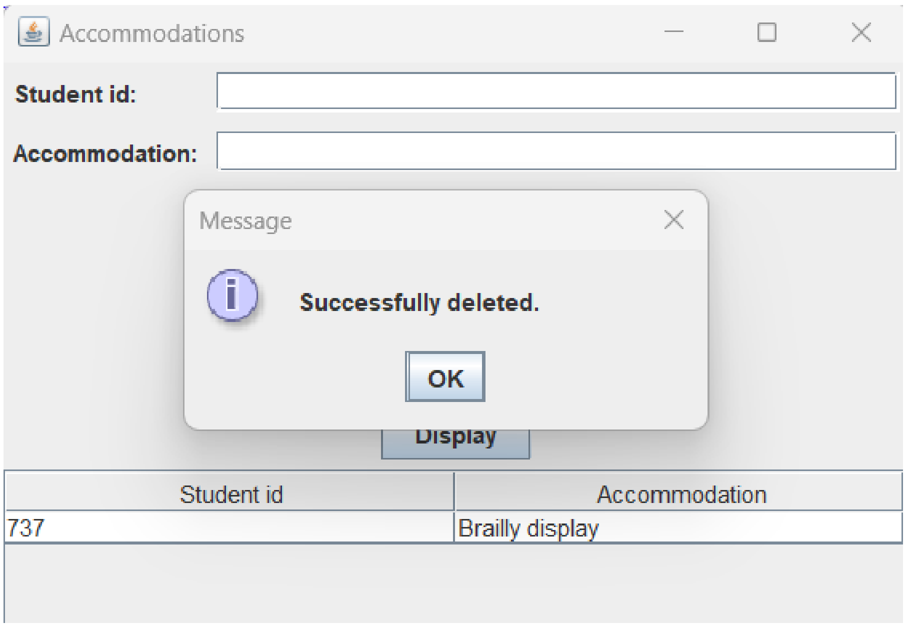
Student id	Accommodation
737	Brailly display
735	sign language interpreter



Student id	Accommodation
737	
735	

Please select an student to delete.

After deletion:

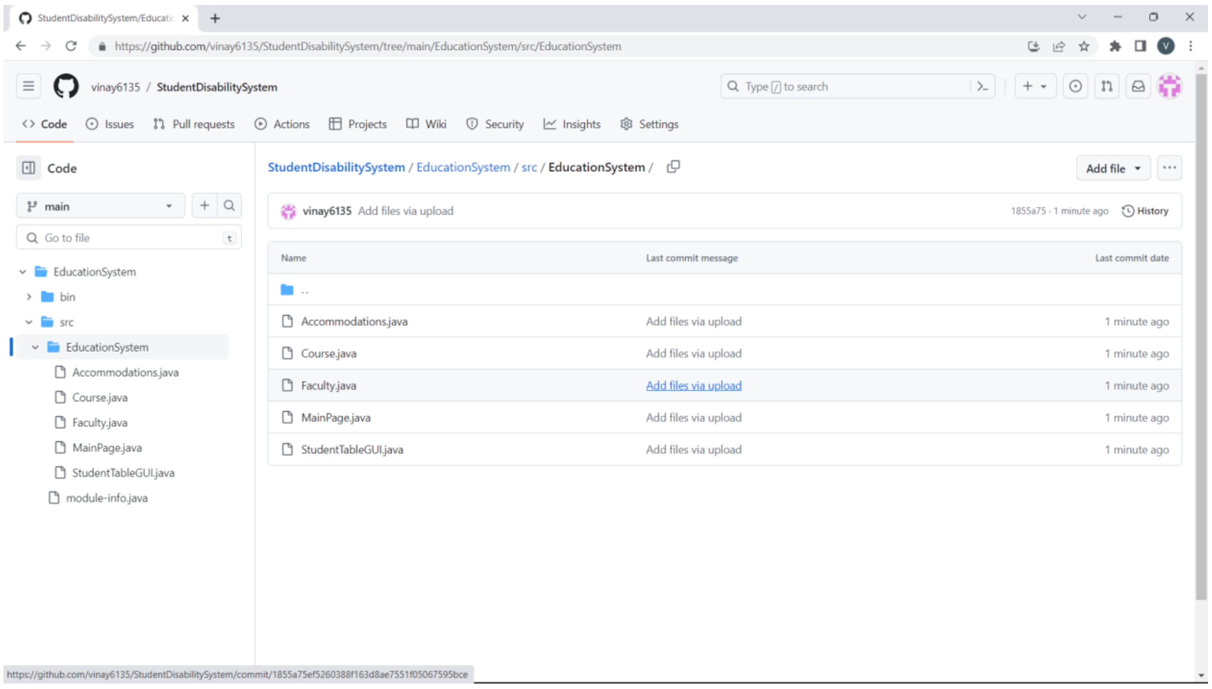


Successfully deleted.

Student id	Accommodation
737	Brailly display

GITHUB LINKS AND FOLDER STRUCTURE:

https://github.com/vinay6135/StudentDisabilitySystem


StudentDisabilitySystem/Educati
https://github.com/vinay6135/StudentDisabilitySystem/tree/main/EducationSystem/src/EducationSystem
vinay6135 / StudentDisabilitySystem
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Code
main
Go to file
Add file
History
https://github.com/vinay6135/StudentDisabilitySystem/commit/1855a75ef5260388f163d8ae7551f05067595bce

StudentDisabilitySystem / EducationSystem / src / EducationSystem /

vinay6135 Add files via upload 1855a75 · 1 minute ago

Name	Last commit message	Last commit date
..		
Accommodations.java	Add files via upload	1 minute ago
Course.java	Add files via upload	1 minute ago
Faculty.java	Add files via upload	1 minute ago
MainPage.java	Add files via upload	1 minute ago
StudentTableGUI.java	Add files via upload	1 minute ago

```
EducationSystem
  bin
  src
    EducationSystem
      Accommodations.java
      Course.java
      Faculty.java
      MainPage.java
      StudentTableGUI.java
    module-info.java
```

Result:

I have successfully completed my DBMS Project “Education Ecosystem for Specially abled students”.

DISCUSSION AND FUTURE WORK:

This project stores the details of the specially abled students ,based on the data provided by the them, government will provide appropriate assistive technologies based on individual students needs.

References:

- https://docs.oracle.com/javase/7/docs/api/
- https://www.javatpoint.com/java-swing
- https://stackoverflow.com/