

JAVA SWINGS BASED- STUDENT ASSIGNMENTS TRACKER
- SQL CONNECTIVITY USING JDBC

A

Report

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

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

Ande Varun Kumar Reddy<1602-19-737-121>

Under the guidance of Ms B. Leelavathy



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

BONAFIDE CERTIFICATE

This is to certify that this project report titled '**STUDENT ASSIGNMENTS TRACKER**' is a project work of Mr. Ande Varun Kumar Reddy bearing roll no. 1602-19-737-121 who carried out the project under my supervision in the IV semester for the academic year 2020- 2021.

Signature

Internal Examiner

Signature

External Examiner

ABSTRACT:

Student Assignments Tracker is a console based application which allows students to view all their assignments, their due dates, any remarks, whether due date is finished or not for particular assignment and their scores also faculty can assign assignments to particular section students with points, instructions and due dates and also give scores and feedback.

REQUIREMENT ANALYSIS:

List of Tables:

- Login
- Students
- Faculty
- Assignments
- Views
- Manages

List of attributes with their domain types:

Login

- user_id varchar2(50)
- Password varchar2(50)
- Usercategory varchar2(50)

Students

- Name varchar2(40)
- Rollnumber number(5)
- Division varchar2(4)
- Dept_id number(5)
- User_id varchar2(50)

Faculty

- Staff_id number(5)
- Name varchar2(50)

Student Assignments tracker

- Subject varchar2(20)
- Dept_id number(5)

Assignments

- Assign_name varchar2(30)
- Points number(5)
- Feedback varchar2(100)
- Rollnumber number(5)

Views

- Rollnumber number(5)
- Due_date date
- Division varchar2(5)
- Assign_name varchar2(50)

Manages

- Section varchar2(10)
- Staff_id number(20)
- Assign_name varchar2(50)
- Subject varchar2(50)

AIM AND PRIORITY OF THE PROJECT

To create a Java GUI based student assignments tracker which takes the values like: user id, password for login and stores details of all students, faculty and student assignments their marks in those and staff can assign assignments to students. These values are to be updated in the database using JDBC connectivity.

ARCHITECTURE AND TECHNOLOGY

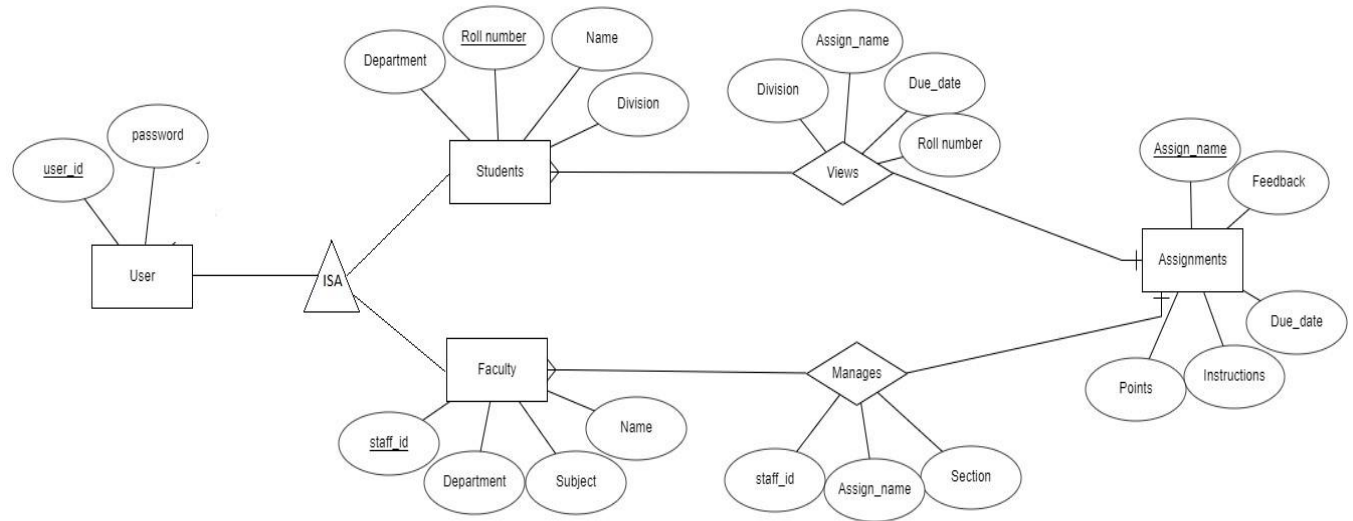
Software used: Java Eclipse, Oracle 11g Database, Java SE version 13, SQL*Plus.

Java SWINGS:

Java SWINGS is an API to develop GUI or window-based applications in java. Java SWING components are platform-independent. It is lightweight. The javax.swing package provides classes for SWING API such as JTextField, JLabel, JTextArea, JRadioButton, JCheckBox, JChoice, JList etc.

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:****Mapping Cardinality and participation constraints:**

Every user who login into application need user id and password. So, login has full participation to users. A user can only be either student or faculty so it is one to one relationship. One student can view any number of assignments so it is one to many relationship. One faculty can give any number of assignments so it is also one to many relationship.

DDL COMMANDS:

```
SQL> create table Login(user_id varchar2(50) PRIMARY KEY,
  2 password varchar2(50) NOT NULL);
```

Table created.

```
SQL> desc Login;
```

Name	Null?	Type
USER_ID	NOT NULL	VARCHAR2(50)
PASSWORD	NOT NULL	VARCHAR2(50)

```
SQL> alter table login add(usercategory varchar2(50));
```

Table altered.

```
SQL> desc login;
```

Name	Null?	Type
USER_ID	NOT NULL	VARCHAR2(50)
PASSWORD	NOT NULL	VARCHAR2(50)
USERCATEGORY		VARCHAR2(50)

```
SQL> create table students(name varchar2(40) NOT NULL,  
2 rollnumber number(5) PRIMARY KEY,  
3 division varchar2(4) NOT NULL,  
4 dept_id number(5) NOT NULL);
```

Table created.

```
SQL> alter table students add FOREIGN KEY(user_id) references Login;
```

Table altered.

```
SQL> desc students;
```

Name	Null?	Type
NAME	NOT NULL	VARCHAR2(40)
ROLLNUMBER	NOT NULL	NUMBER(5)
DIVISION	NOT NULL	VARCHAR2(4)
DEPT_ID	NOT NULL	NUMBER(5)
USER_ID		VARCHAR2(50)

```
SQL> desc students;
```

Name	Null?	Type
NAME	NOT NULL	VARCHAR2(40)
ROLLNUMBER	NOT NULL	NUMBER(5)
DIVISION	NOT NULL	VARCHAR2(4)
DEPT_ID	NOT NULL	NUMBER(5)

```
SQL> create table faculty(staff_id number(5) PRIMARY KEY,  
2 name varchar2(50),  
3  
SQL> create table faculty(staff_id number(5) PRIMARY KEY,  
2 name varchar2(50) NOT NULL,  
3 subject varchar2(20) NOT NULL,  
4 dept_id number(5) NOT NULL);
```

Table created.

```
SQL> desc faculty;
Name                                         Null?    Type
-----
STAFF_ID                                     NOT NULL NUMBER(5)
NAME                                         NOT NULL VARCHAR2(50)
SUBJECT                                     NOT NULL VARCHAR2(20)
DEPT_ID                                     NOT NULL NUMBER(5)
```

```
SQL> create table Assignments(assign_name varchar2(30) PRIMARY KEY,
2  points number(5) NOT NULL,
3  due_date date,
4  feedback varchar2(100),
5  instructions varchar2(500));

Table created.
```

```
SQL> alter table assignments drop column due_date;

Table altered.

SQL> alter table assignments drop column instructions;

Table altered.

SQL> alter table assignments add(rollnumber number(5));

Table altered.

SQL> alter table assignments add FOREIGN KEY(rollnumber) references students;

Table altered.

SQL> desc assignments;
Name                                         Null?    Type
-----
ASSIGN_NAME                                 NOT NULL VARCHAR2(50)
POINTS                                     NOT NULL NUMBER(5)
FEEDBACK                                   VARCHAR2(100)
ROLLNUMBER                                 NUMBER(5)
```

```
SQL> create table manages(section varchar2(3) NOT NULL,
2  staff_id number(5));

Table created.

SQL> alter table manages add FOREIGN KEY(staff_id) references faculty;

Table altered.
```



```
SQL> alter table manages add assign_name varchar2(20);
Table altered.

SQL> alter table manages add FOREIGN KEY(assign_name) references Assignments;
Table altered.

SQL> desc manages;
  Name                                         Null?     Type
-----
SECTION                                     NOT NULL  VARCHAR2(3)
STAFF_ID                                              NUMBER(5)
ASSIGN_NAME                                          VARCHAR2(20)
```

```
SQL> alter table manages add(assign_name varchar2(50));
Table altered.

SQL> alter table manages add(subject varchar2(20));
Table altered.

SQL> desc manages;
  Name                                         Null?     Type
-----
SECTION                                     NOT NULL  VARCHAR2(5)
STAFF_ID                                              NUMBER(6)
ASSIGN_NAME                                          VARCHAR2(50)
SUBJECT                                              VARCHAR2(20)
```

```
SQL> create table views(rollnumber number(5),
 2  due_date date,
 3  division varchar2(5),
 4  assign_name varchar2(50));
Table created.
```

```
SQL> alter table views add FOREIGN KEY(assign_name) references Assignments;
Table altered.
```

```
SQL> desc views;
```

Name	Null?	Type
ROLLNUMBER		NUMBER(5)
DUE_DATE		DATE
DIVISION		VARCHAR2(5)
ASSIGN_NAME		VARCHAR2(50)

```
SQL> select * from tab;
```

TNAME	TABTYPE	CLUSTERID
ASSIGNMENTS	TABLE	
FACULTY	TABLE	
LOGIN	TABLE	
MANAGES	TABLE	
STUDENTS	TABLE	
VIEWS	TABLE	

```
6 rows selected.
```

```
SQL> alter table views add FOREIGN KEY(rollnumber) references students;
Table altered.
```

DML COMMANDS:

```
SQL> insert into login values('akshith1098','ghij001','student');  
  
1 row created.  
  
SQL> insert into login values('vikram199','a9bnjk','staff');  
  
1 row created.
```

```
SQL> insert into students values('&name','&rollnumber','&division','&dept_id','&userid');  
Enter value for name: varun  
Enter value for rollnumber: 121  
Enter value for division: B  
Enter value for dept_id: 737  
Enter value for userid: 121_varun  
old 1: insert into students values('&name','&rollnumber','&division','&dept_id','&userid')  
new 1: insert into students values('varun','121','B','737','121_varun')  
  
1 row created.  
  
SQL> /  
Enter value for name: charan  
Enter value for rollnumber: 006  
Enter value for division: A  
Enter value for dept_id: 733  
Enter value for userid: 006_charan  
old 1: insert into students values('&name','&rollnumber','&division','&dept_id','&userid')  
new 1: insert into students values('charan','006','A','733','006_charan')  
  
1 row created.
```

```
SQL> select * from Login;
```

```
USER_ID
```

```
PASSWORD
```

```
121_varun  
abc3456
```

```
006_charan  
uhg0912
```

```
pps_51  
jik1786
```

```
USER_ID
```

```
PASSWORD
```

```
dbms_21  
fdghnb875
```

```
SQL> select * from students;
```

NAME	ROLLNUMBER	DIVI	DEPT_ID
varun	121	B	737
121_varun			
charan	6	A	733
006_charan			

```
SQL> insert into faculty values('&staff_id','&name','&subject','&dept_id');
Enter value for staff_id: 73751
Enter value for name: Ravi
Enter value for subject: PPS
Enter value for dept_id: 737
old 1: insert into faculty values('&staff_id','&name','&subject','&dept_id')
new 1: insert into faculty values('73751','Ravi','PPS','737')

1 row created.
```

```
SQL> /
Enter value for staff_id: 73321
Enter value for name: Vijay
Enter value for subject: DBMS
Enter value for dept_id: 733
old 1: insert into faculty values('&staff_id','&name','&subject','&dept_id')
new 1: insert into faculty values('73321','Vijay','DBMS','733')

1 row created.
```

```
SQL> select * from faculty;
```

STAFF_ID	NAME	SUBJECT	DEPT_ID
73751	Ravi	PPS	737
73321	Vijay	DBMS	733

```
SQL> insert into assignments values('&assign_name',&points,'&feedback','&rollnumber');
Enter value for assign_name: ppsassignment
Enter value for points: 4
Enter value for feedback: good
Enter value for rollnumber: 102
old 1: insert into assignments values('&assign_name',&points,'&feedback','&rollnumber')
new 1: insert into assignments values('ppsassignment',4,'good','102')

1 row created.
```

```
SQL> insert into views values(121,TO_DATE('2021/05/20 23:59:59','yyyy/mm/dd hh24:mi:ss'),'B','PPS_ASSIGNMENT1');
1 row created.

SQL> insert into views values(006,TO_DATE('2021/05/15 21:02:44','yyyy/mm/dd hh24:mi:ss'),'A','DBMS_ASSIGNMENT1');
1 row created.

SQL> select * from views;
```

ROLLNUMBER	DUE_DATE	DIVIS	ASSIGN_NAME
121	20-MAY-21	B	PPS_ASSIGNMENT1
6	15-MAY-21	A	DBMS_ASSIGNMENT1

```
SQL> insert into manages values('&section',&staff_id,&assign_name','&subject');
Enter value for section: B
Enter value for staff_id: 73701
Enter value for assign_name: dbmsassignment1
Enter value for subject: dbms
old 1: insert into manages values('&section',&staff_id,&assign_name','&subject')
new 1: insert into manages values('B',73701,'dbmsassignment1','dbms')

1 row created.
```

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
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","it19737121","vasavi");
        statement=con.createStatement();
        statement.executeUpdate("commit");
    }
    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.

LOGINPAGE (MAIN METHOD):

```
package Sat;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.EventQueue;
import Sat.*;
import java.sql.*;
import javax.sql.*;
import javax.swing.*;
import javax.swing.border.EmptyBorder;
import javax.swing.event.MenuEvent;
import javax.swing.event.MenuListener;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
public class loginpage extends JFrame
{
    private static final long serialVersionUID = 1L;
    private JFrame StudentAssignmentsTracker;
    private JTextField txtuserid;
    private JPasswordField txtpwd;
    private JPanel contentPane;
    static String s;
    static String str1;
    public static void main(String[] args)
    {
        EventQueue.invokeLater(new Runnable()
        {
            public void run()
            {
                try
                {
                    loginpage window = new loginpage();
                    window.StudentAssignmentsTracker.setVisible(true);
                }
                catch (Exception e)
                {
                    e.printStackTrace();
                }
            }
        });
    }
    public loginpage() {
        initialize();
    }
    JButton btnLogin;
    private void initialize()
    {
        StudentAssignmentsTracker = new JFrame();
        StudentAssignmentsTracker.setTitle("Student Assignments Tracker");
        StudentAssignmentsTracker.setBounds(150, 150, 400, 300);
    }
}
```



```

StudentAssignmentsTracker.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
StudentAssignmentsTracker.getContentPane().setLayout(null);
JMenuBar menuBar = new JMenuBar();
StudentAssignmentsTracker.setJMenuBar(menuBar);
JMenu mnNewMenu = new JMenu("Login");
mnNewMenu.addMenuListener(new MenuListener() {
    public void menuCanceled(MenuEvent e) {
    }
    public void menuDeselected(MenuEvent e) {
    }
    public void menuSelected(MenuEvent e) {
        btnLogin.setVisible(true);
    }
});
menuBar.add(mnNewMenu);
txtuserid = new JTextField();
txtuserid.setBounds(198, 47, 96, 20);
StudentAssignmentsTracker.getContentPane().add(txtuserid);
txtuserid.setColumns(10);

JLabel lblNewLabel = new JLabel("User ID");
lblNewLabel.setBounds(76, 50, 84, 14);
StudentAssignmentsTracker.getContentPane().add(lblNewLabel);

JLabel lblNewLabel_1 = new JLabel("Password");
lblNewLabel_1.setBounds(76, 88, 84, 14);
StudentAssignmentsTracker.getContentPane().add(lblNewLabel_1);

btnLogin = new JButton("Login");
//contentPane.setBackground(Color.cyan);
btnLogin.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e) {
        boolean isValidUser=false;
        try
        {
            DBconnection connect = new DBconnection();
            Connection con = connect.getConnection();
            Statement stmt=con.createStatement();
            str1=txtuserid.getText();
            String txt = "select usercategory from login where user_id =
            '"+txtuserid.getText()+"' and password = '"+new String(txtpwd.getPassword())+"'";
            ResultSet rs=stmt.executeQuery(txt);
            if(rs.next())
            {
                s=rs.getString("usercategory");
                isValidUser = true;
            }
            rs.close();
        }
        catch(Exception ex) {

```

Student Assignments tracker

```
        System.out.println(ex);
    }
    if(isValidUser) {
        UserInterface h=new UserInterface();

    }
    else
    {
        JOptionPane.showMessageDialog(null,"INVALID LOGIN");
    }

    }

});
btnLogin.setBounds(150, 160, 89, 23);
StudentAssignmentsTracker.getContentPane().add(btnLogin);
txtpwd = new JPasswordField();
txtpwd.setBounds(198, 85, 96, 20);
StudentAssignmentsTracker.getContentPane().add(txtpwd);
}
}
```

LOGIN:

```
package Sat;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.sql.*;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;
public class login
{
    private static final long serialVersionUID = 1L;
```

```

    private JButton insertButton,deleteButton,viewButton;
    private JPanel p1,p2,p3,p;
    private JLabel lbluser_id,lblpassword,lblusertype;
    private JTextField txtuser_id,txtpassword,txtusertype;
    private List loginIDList;
    Connection con;
    ResultSet rs;
    Statement statement;
    private JFrame frame;
    private JMenuItem insert,delete,update,view;
    public login(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem
update,JMenuItem view)
    {
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectToDB();
        this.frame=frame;
        this.insert=insert;
        this.delete=delete;
        this.update=update;
        this.view=view;
        lbluser_id=new JLabel("User_id");
        lblpassword=new JLabel("Password");
        lblusertype=new JLabel("User Category");
        txtuser_id=new JTextField(50);
        txtpassword=new JPasswordField(50);
        txtusertype=new JTextField(50);
        this.p=p;
    }
    public void connectToDB()
    {
        try {
            Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","it19737121","vasavi");
            statement=con.createStatement();
            statement.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(p, "\nSQLException: " + e.getMessage() +
"\n"+"SQLState:      " + e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() + "\n");
    }
    public void loadLogins() {
        try {
            loginIDList.removeAll();
            rs=statement.executeQuery("select user_id from login");
            while(rs.next()) {
                loginIDList.add(rs.getString("user_id"));
            }
        }
        catch(SQLException e) {
            displaySQLErrors(e);
        }
    }
    public void buildGUI() {
        insertButton=new JButton("Submit");

        insert.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent arg0) {
                // TODO Auto-generated method stub

                txtuser_id.setText(null);
                txtpassword.setText(null);
                txtusertype.setText(null);
                p.removeAll();
                frame.invalidate();
                frame.validate();
                frame.repaint();

                p1=new JPanel();

                p1.setLayout(new GridLayout(3,2));
                p1.add(lbluser_id);
                p1.add(txtuser_id);
                p1.add(lblpassword);
                p1.add(txtpassword);
                p1.add(lblusertype);
                p1.add(txtusertype);
                p3=new JPanel(new FlowLayout());
                p3.add(insertButton);
                p3.setBackground(Color.blue);
                p3.setBounds(200,280,75,35);
                p1.setBackground(Color.orange) ;

                p1.setBounds(115,80,300,200);

                p2 = new JPanel(new FlowLayout());
                // p2.add(txtf1);
                loginIDList=new List(10);
                loadLogins();
            }
        });
    }
}

```

```
        p2.add(loginIDList);
        p2.setBackground(Color.cyan) ;
        p2.setBounds(125,320,300,180);

        p.add(p1);
        p.add(p3);
        p.add(p2);

        p.setLayout(new BorderLayout());

        frame.add(p);
        frame.setSize(800,800);
        frame.validate();
        insertButton.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub

            try {
                String query="INSERT INTO login
VALUES('"+txtuser_id.getText()+"', '"+txtpassword.getText()+"', '"+txtusertype.getText()+"')";
                int i=statement.executeUpdate(query);
                JOptionPane.showMessageDialog(p,"\\nInserted "+i+" rows
succesfully");

                loadLogins();

                System.out.println("done");
            }
            catch(SQLException insertException){
                displaySQLErrors(insertException);
            }

        }

    });

    delete.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent arg0) {
            // TODO Auto-generated method stub
            deleteButton=new JButton("Delete");

            txtuser_id.setText(null);
            txtpassword.setText(null);
            txtusertype.setText(null);
            p.removeAll();
            frame.invalidate();
```

```

frame.validate();
frame.repaint();

p1=new JPanel();

p1.setLayout(new GridLayout(3,2));
p1.add(lbluser_id);
p1.add(txtuser_id);
p1.add(lblpassword);
p1.add(txtpassword);
p1.add(lblusertype);
p1.add(txtusertype);
p3=new JPanel(new FlowLayout());
p3.add(deleteButton);
p3.setBackground(Color.blue);
p3.setBounds(200,280,75,35);
p1.setBackground(Color.orange) ;

p1.setBounds(115,80,300,200);

p2 = new JPanel(new FlowLayout());
    loginIDList=new List(10);
    loadLogins();
    p2.add(loginIDList);
    p2.setBackground(Color.cyan) ;
    p2.setBounds(125,320,300,180);

p.add(p1);
p.add(p3);
p.add(p2);

p.setLayout(new BorderLayout());

    frame.add(p);
    frame.setSize(800,800);
    frame.validate();
loginIDList.addItemListener(new ItemListener()
{
    public void itemStateChanged(ItemEvent e)
    {
        try
        {
            rs=statement.executeQuery("select * from
login");

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

```

```

        {

txtuser_id.setText(rs.getString("user_id"));

txtpassword.setText(rs.getString("password"));

txtusertype.setText(rs.getString("usercategory"));

        }
    }
    catch (SQLException selectException)
    {
        displaySQLErrors(selectException);
    }
}

});

deleteButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        try {

            String password=JOptionPane.showInputDialog(p,"Enter the
password");

            txtpassword.setText(password);
            int a=JOptionPane.showConfirmDialog(p,"Are you sure want to
delete:");

            if(a==JOptionPane.YES_OPTION){
                String query="DELETE FROM LOGIN WHERE
USER_ID='"+txtuser_id.getText()+"'AND PASSWORD='"+txtpassword.getText()+"'";

                int i=statement.executeUpdate(query);
                JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows
succesfully");

                loadLogins();
            }

        }
        catch(SQLException deleteException){
            displaySQLErrors(deleteException);
        }

    }

});

update.addActionListener(new ActionListener() {

```

```

@Override
public void actionPerformed(ActionEvent arg0) {
    // TODO Auto-generated method stub
    JButton updateButton = new JButton("Modify");
    txtuser_id.setText(null);
    txtpassword.setText(null);
    txtusertype.setText(null);
    p.removeAll();
    frame.invalidate();
    frame.validate();
    frame.repaint();

    p1=new JPanel();

    p1.setLayout(new GridLayout(3,2));
    p1.add(lbluser_id);
    p1.add(txtuser_id);
    p1.add(lblpassword);
    p1.add(txtpassword);
    p1.add(lblusertype);
    p1.add(txtusertype);

    p3=new JPanel(new FlowLayout());
    p3.add(updateButton);
    //p1.add(txtf1);
    p3.setBackground(Color.blue);
    p3.setBounds(200,280,75,35);
    p1.setBackground(Color.orange) ;

    p1.setBounds(115,80,300,200);

    p2 = new JPanel(new FlowLayout());
    // p2.add(txtf1);
    loginIDList=new List(10);
    //loadLogins();
    if(loginpage.s.equals("student") ||
loginpage.s.equals("staff"))
    {
        loginIDList.removeAll();
        loginIDList.add(loginpage.str1);
    }
    if(loginpage.s.equals("admin"))
    {
        loadLogins();
    }
    p2.add(loginIDList);
    p2.setBackground(Color.cyan) ;
    p2.setBounds(125,320,300,180);

    p.add(p1);
    p.add(p3);

```



```

        p.add(p2);

        p.setLayout(new BorderLayout());

        frame.add(p);
        frame.setSize(800,800);
        frame.validate();
        loginIDList.addItemListener(new ItemListener()
        {
            public void itemStateChanged(ItemEvent e)
            {
                try
                {
                    rs=statement.executeQuery("select * from
login");
                    while (rs.next())
                    {
                        if
(rs.getString("user_id").equals(loginIDList.getSelectedItem()))
                        break;
                    }
                    if (!rs.isAfterLast())
                    {

                        txtuser_id.setText(rs.getString("user_id"));

                        txtpassword.setText(rs.getString("password"));

                        txtusertype.setText(rs.getString("usercategory"));
                    }
                }
                catch (SQLException selectException)
                {
                    displaySQLErrors(selectException);
                }
            }
        });

        updateButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
            }
        });

        try
        {
            if(loginpage.s.equals("admin"))
            {
                loadLogins();
                String
password1=JOptionPane.showInputDialog(p,"Enter the new password");

                txtpassword.setText(password1);
            }
        }
        catch (Exception ex)
        {
            displayErrors(ex);
        }
    }
}

```

```
String query="update login set
password='"+password1+"' where user_id='"+txtuser_id.getText()+"' ";

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nUpdated
"+i+" rows succesfully");

loadLogins();
}
if(loginpage.s.equals("student") ||
loginpage.s.equals("staff"))
{
    loginIDList.removeAll();
    loginIDList.add(loginpage.str1);
    String
password2=JOptionPane.showInputDialog(p,"Enter the new password");

    txtpassword.setText(password2);

    String query="update login set
password='"+password2+"' where user_id='"+txtuser_id.getText()+"' ";

    int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nUpdated "+i+" rows succesfully");
}

}
catch(SQLException updateException){

    displaySQLErrors(updateException);
}

}

});

view.addActionListener(new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub

        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();

        Label view1=new Label("Login View");
        //view1.setAlignment(Label.CENTER);
        Font myFont = new Font("Serif",Font.BOLD,50);
```

```

view1.setFont(myFont));
viewButton=new JButton("View");
p1=new JPanel();
p2=new JPanel();
p1.add(view1);
p2.add(viewButton);
p1.setBackground(Color.cyan) ;
p2.setBackground(Color.blue) ;
p.add(p1);p.add(p2);
p.setLayout(new FlowLayout());

p.setBounds(500,800,300,300);
frame.add(p);
frame.setSize(800,800);
frame.validate();
viewButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        JFrame f;

        JTable j;

        f = new JFrame();

        f.setTitle("Login details");

        DefaultTableModel model = new
DefaultTableModel();

        j = new JTable(model);
        model.addColumn("user_id");

        model.addColumn("password");
        model.addColumn("usercategory");
        try {

            rs=statement.executeQuery("select
* from login");

            while(rs.next()) {
                model.addRow(new
Object[]{rs.getString("user_id"), rs.getString("password"),rs.getString("usercategory")});
            }
        }
        catch(SQLException viewException) {
            displaySQLErrors(viewException);
        }
        j.setEnabled(false);
        j.setBounds(30, 40, 150, 150);

        JScrollPane sp = new JScrollPane(j);
    }
});

```

```
f.add(sp);

f.setSize(400, 400);

f.setVisible(true);

}

});

}

});

}
```

STUDENTS:

```
package Sat;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.sql.*;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;
public class Students
```

```

{
    private static final long serialVersionUID = 1L;
    private JButton insertButton,deleteButton,viewButton;
    private JPanel p1,p2,p3,p;
    private JLabel lbluser_id,lblname,lblrollno,lbldiv,lbldept_id;
    private JTextField txtuser_id,txtname,txtrollno,txtdiv,txtdept_id;
    private List RollnoList;
    Connection con;
    ResultSet rs;
    Statement statement;
    private JFrame frame;
    private JMenuItem insert,delete,update,view;
    public Students(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem
update,JMenuItem view)
    {
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectToDB();
        this.frame=frame;
        this.insert=insert;
        this.delete=delete;
        this.update=update;
        this.view=view;
        lbluser_id=new JLabel("User_id");
        lblname=new JLabel("Name");
        lblrollno=new JLabel("Roll Number");
        lbldiv=new JLabel("Division");
        lbldept_id=new JLabel("Dept_id");
        txtuser_id=new JTextField(50);
        txtname=new JTextField(50);
        txtrollno=new JTextField(10);
        txtdiv=new JTextField(10);
        txtdept_id=new JTextField(10);
        this.p=p;
    }
    public void connectToDB()
    {
        try {
            Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","it19737121","vasavi");
            statement=con.createStatement();
            statement.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(p, "\nSQLException: " + e.getMessage() +
"\n"+ "SQLState: " + e.getSQLState() + "\n"+ "VendorError: " + e.getErrorCode() + "\n");
}

public void loadrollno() {
    try {
        RollnoList.removeAll();
        rs=statement.executeQuery("select rollnumber from students");
        while(rs.next()) {
            RollnoList.add(rs.getString("rollnumber"));
        }
    }
    catch(SQLException e) {
        displaySQLErrors(e);
    }
}

public void buildGUI()
{
    //if(loginpage.s=="staff" || loginpage.s=="admin")
    //{
    insertButton=new JButton("Submit");

    insert.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent arg0) {
            // TODO Auto-generated method stub
            txtuser_id.setText(null);
            txtname.setText(null);
            txtrollno.setText(null);
            txtdiv.setText(null);
            txtdept_id.setText(null);
            p.removeAll();
            frame.invalidate();
            frame.validate();
            frame.repaint();
            p1=new JPanel();
            p1.setLayout(new GridLayout(3,2));
            p1.add(lbluser_id);
            p1.add(txtuser_id);
            p1.add(lblname);
            p1.add(txtname);
            p1.add(lblrollno);
            p1.add(txtrollno);
            p1.add(lbldiv);
            p1.add(txtdiv);
            p1.add(lbldept_id);
            p1.add(txtdept_id);
        }
    });
}

```

```

        p3=new JPanel(new FlowLayout());
        p3.add(insertButton);
        p3.setBackground(Color.blue);
        p3.setBounds(200,280,75,35);
        p1.setBackground(Color.orange) ;

        p1.setBounds(115,80,300,200);

        p2 = new JPanel(new FlowLayout());
        // p2.add(txtf1);
        RollnoList=new List(10);
        loadrollno();
        p2.add(RollnoList);
        p2.setBackground(Color.cyan) ;
        p2.setBounds(125,320,300,180);

        p.add(p1);
        p.add(p3);
        p.add(p2);

        p.setLayout(new BorderLayout());

        frame.add(p);
        frame.setSize(800,800);
        frame.validate();
        insertButton.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub

                try {
                    String query="INSERT INTO STUDENTS
VALUES('"+txtname.getText()+"','"+txtrollno.getText()+"','"+txtdiv.getText()+"','"+txtdept_id.ge
tText()+"','"+txtuser_id.getText()+"')";
                    int i=statement.executeUpdate(query);
                    JOptionPane.showMessageDialog(p,"\nInserted "+i+" rows
succesfully");

                    loadrollno();
                    System.out.println("Done");
                }
                catch(SQLException insertException){
                    displaySQLErrors(insertException);
                }

            }

        });

    });
});

```

```

delete.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        deleteButton=new JButton("Delete");

        txtuser_id.setText(null);
        txtname.setText(null);
        txtrollno.setText(null);
        txtdiv.setText(null);
        txtdept_id.setText(null);
        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
        p1=new JPanel();
        p1.setLayout(new GridLayout(3,2));
        p1.add(lbluser_id);
        p1.add(txtuser_id);
        p1.add(lblname);
        p1.add(txtname);
        p1.add(lblrollno);
        p1.add(txtrollno);
        p1.add(lbldiv);
        p1.add(txtdiv);
        p1.add(lbldept_id);
        p1.add(txtdept_id);
        p3=new JPanel(new FlowLayout());
        p3.add(deleteButton);
        p3.setBackground(Color.blue);
        p3.setBounds(200,280,75,35);
        p1.setBackground(Color.orange) ;
        p1.setBounds(115,80,300,200);

        p2 = new JPanel(new FlowLayout());
        RollnoList=new List(10);
        loadrollno();
        p2.add(RollnoList);
        p2.setBackground(Color.cyan) ;
        p2.setBounds(125,320,300,180);

        p.add(p1);
        p.add(p3);
        p.add(p2);

        p.setLayout(new BorderLayout());

        frame.add(p);
        frame.setSize(800,800);
    }
});

```



```
        frame.validate();
        RollnoList.addItemListener(new ItemListener()
        {
            public void itemStateChanged(ItemEvent e)
            {
                try
                {
                    rs=statement.executeQuery("select * from
students");

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

                        txtuser_id.setText(rs.getString("user_id"));

                        txtname.setText(rs.getString("name"));

                        txtrollno.setText(rs.getString("rollnumber"));

                        txtdiv.setText(rs.getString("division"));

                        txtdept_id.setText(rs.getString("dept_id"));

                    }
                }
                catch (SQLException selectException)
                {
                    displaySQLErrors(selectException);
                }
            }
        });

        deleteButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub

                try {

                    int a=JOptionPane.showConfirmDialog(p,"Are you sure want to
delete:");

                    if(a==JOptionPane.YES_OPTION){
                        String query="DELETE FROM STUDENTS WHERE
ROLLNUMBER="+RollnoList.getSelectedItem();
                        int i=statement.executeUpdate(query);
                        JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows
succesfully");

                        loadrollno();
                    }
                }
            }
        });
```

```
        }
        catch(SQLException deleteException){
            displaySQLErrors(deleteException);
        }
    }

    });

}
});

update.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        JButton updateButton = new JButton("Modify");

        txtname.setText(null);
        txtrollno.setText(null);
        txtdiv.setText(null);
        txtdept_id.setText(null);
        txtuser_id.setText(null);
        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();

        p1=new JPanel();

        p1.setLayout(new GridLayout(3,2));

        p1.add(lblname);
        p1.add(txtname);
        p1.add(lblrollno);
        p1.add(txtrollno);
        p1.add(lbldiv);
        p1.add(txtdiv);
        p1.add(lbldept_id);
        p1.add(txtdept_id);
        p1.add(lbluser_id);
        p1.add(txtuser_id);
        p3=new JPanel(new FlowLayout());
        p3.add(updateButton);
        p3.setBackground(Color.blue);
        p3.setBounds(200,280,75,35);
        p1.setBackground(Color.orange) ;

        p1.setBounds(115,80,300,200);
```

```
p2 = new JPanel(new FlowLayout());
// p2.add(txtf1);
RollnoList=new List(10);
loadrollno();
p2.add(RollnoList);
p2.setBackground(Color.cyan) ;
p2.setBounds(125,320,300,180);

p.add(p1);
p.add(p3);
p.add(p2);

p.setLayout(new BorderLayout());

frame.add(p);
frame.setSize(800,800);
frame.validate();
RollnoList.addItemListener(new ItemListener()
{
    public void itemStateChanged(ItemEvent e)
    {
        try
        {
            rs=statement.executeQuery("select * from
students");

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

                txtuser_id.setText(rs.getString("user_id"));

                txtname.setText(rs.getString("name"));

                txtrollno.setText(rs.getString("rollnumber"));

                txtdiv.setText(rs.getString("division"));

                txtdept_id.setText(rs.getString("dept_id"));

            }
        }
        catch (SQLException selectException)
        {
            displaySQLErrors(selectException);
        }
    }
});
```

```
        updateButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
                try {
                    loadrollno();
                    String query="update students set
name='"+txtname.getText()+"',division='"+txtdiv.getText()+"',dept_id='"+txtdept_id.getText()+"
WHERE rollnumber='"+txtrollno.getText()+" ";
                    int i=statement.executeUpdate(query);
                    JOptionPane.showMessageDialog(p,"\nUpdated "+i+" rows
succesfully");
                    loadrollno();

                }
                catch(SQLException updateException){

                    displaySQLErrors(updateException);
                }

            }

        });
    }
});
//}
view.addActionListener(new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub

        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();

        Label view1=new Label("Students view");
        //view1.setAlignment(Label.CENTER);
        Font myFont = new Font("Serif",Font.BOLD,50);
        view1.setFont((myFont));
        viewButton=new JButton("View");
        p1=new JPanel();
        p2=new JPanel();
        p1.add(view1);
        p2.add(viewButton);p1.setBackground(Color.cyan)
;p2.setBackground(Color.blue) ;
        p.add(p1);p.add(p2);
        p.setLayout(new FlowLayout());
```

```
p.setBounds(500,800,300,300);
frame.add(p);
frame.setSize(800,800);
frame.validate();
viewButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        JFrame f;

        JTable j;

        f = new JFrame();

        f.setTitle("Students details");

        DefaultTableModel model = new
DefaultTableModel();

        j = new JTable(model);

        model.addColumn("name");
        model.addColumn("rollnumber");
        model.addColumn("division");
        model.addColumn("dept_id");
        model.addColumn("user_id");

        try {

            rs=statement.executeQuery("select
* from students");

            while(rs.next()) {
                model.addRow(new
Object[]{rs.getString("name"),rs.getString("rollnumber"),rs.getStrin
g("dept_id"),rs.getString("user_id")});
            }
        }
        catch(SQLException viewException) {
            displaySQLerrors(viewException);
        }
        j.setEnabled(false);
        j.setBounds(30, 40, 150, 150);

        JScrollPane sp = new JScrollPane(j);
        f.add(sp);

        f.setSize(400, 400);

        f.setVisible(true);
```

```
        }  
  
        });  
  
    }  
  
});  
  
}
```

FACULTY:

```
package Sat;  
import java.awt.BorderLayout;  
import java.awt.Color;  
import java.awt.FlowLayout;  
import java.awt.Font;  
import java.awt.GridLayout;  
import java.awt.Label;  
import java.awt.List;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.awt.event.ItemEvent;  
import java.awt.event.ItemListener;  
import java.sql.*;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JMenuItem;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.JPasswordField;  
import javax.swing.JScrollPane;  
import javax.swing.JTable;  
import javax.swing.JTextArea;  
import javax.swing.JTextField;  
import javax.swing.table.DefaultTableModel;  
public class Faculty {  
    private static final long serialVersionUID = 1L;  
    private JButton insertButton,deleteButton,viewButton;  
    private JPanel p1,p2,p3,p;  
    private JLabel lbluser_id,lblname,lblstaff_id,lblsubject,lbldept_id;  
    private JTextField txtuser_id,txtname,txtstaff_id,txtsubject,txtdept_id;  
    private List staffidList;
```

```

        Connection con;
        ResultSet rs;
        Statement statement;
        private JFrame frame;
        private JMenuItem insert,delete,update,view;
        public Faculty(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem
update,JMenuItem view)
        {
            try
            {
                Class.forName("oracle.jdbc.driver.OracleDriver");
            }
            catch (Exception e)
            {
                System.err.println("Unable to find and load driver");
                System.exit(1);
            }
            connectToDB();
            this.frame=frame;
            this.insert=insert;
            this.delete=delete;
            this.update=update;
            this.view=view;
            lbluser_id=new JLabel("User_id");
            lblname=new JLabel("Name");
            lblstaff_id=new JLabel("Staff ID");
            lblsubject=new JLabel("Subject");
            lbldept_id=new JLabel("Dept_id");
            txtuser_id=new JTextField(50);
            txtname=new JTextField(50);
            txtstaff_id=new JTextField(10);
            txtsubject=new JTextField(10);
            txtdept_id=new JTextField(10);
            this.p=p;
        }
        public void connectToDB()
        {
            try {
                Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","it19737121","vasavi");
                statement=con.createStatement();
                statement.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(p, "\nSQLException: " + e.getMessage() +
"\n"+"SQLState:      " + e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() + "\n");
    }
    public void loadstaffid() {
        try {
            staffidList.removeAll();
            rs=statement.executeQuery("select staff_id from faculty");
            while(rs.next()) {
                staffidList.add(rs.getString("staff_id"));
            }
        }
        catch(SQLException e) {
            displaySQLErrors(e);
        }
    }

    public void buildGUI() {
        insertButton=new JButton("Submit");

        insert.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent arg0) {
                // TODO Auto-generated method stub
                txtuser_id.setText(null);
                txtname.setText(null);
                txtstaff_id.setText(null);
                txtsubject.setText(null);
                txtdept_id.setText(null);
                p.removeAll();
                frame.invalidate();
                frame.validate();
                frame.repaint();
                p1=new JPanel();
                p1.setLayout(new GridLayout(3,2));
                p1.add(lbluser_id);
                p1.add(txtuser_id);
                p1.add(lblname);
                p1.add(txtname);
                p1.add(lblstaff_id);
                p1.add(txtstaff_id);
                p1.add(lblsubject);
                p1.add(txtsubject);
                p1.add(lbldept_id);
                p1.add(txtdept_id);
                p3=new JPanel(new FlowLayout());
                p3.add(insertButton);
                p3.setBackground(Color.blue);
                p3.setBounds(200,280,75,35);
                p1.setBackground(Color.orange) ;

                p1.setBounds(115,80,300,200);

                p2 = new JPanel(new FlowLayout());
            }
        });
    }

```



```
// p2.add(txtf1);
staffidList=new List(10);
loadstaffid();
p2.add(staffidList);
p2.setBackground(Color.cyan) ;
p2.setBounds(125,320,300,180);

p.add(p1);
p.add(p3);
p.add(p2);

p.setLayout(new BorderLayout());

frame.add(p);
frame.setSize(800,800);
frame.validate();
insertButton.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub

        try {
            String query="INSERT INTO FACULTY
VALUES("+txtstaff_id.getText()+", '"+txtname.getText()+"', '"+txtsubject.getText()+"', '"+txtdept
_id.getText()+", '"+txtuser_id.getText()+"')";
            int i=statement.executeUpdate(query);
            JOptionPane.showMessageDialog(p, "\nInserted "+i+" rows
succesfully");

            loadstaffid();
            System.out.println("Done");
        }
        catch(SQLException insertException){
            displaySQLErrors(insertException);
        }

    }

});

delete.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        deleteButton=new JButton("Delete");

        txtuser_id.setText(null);
        txtname.setText(null);
```

```

txtstaff_id.setText(null);
txtsubject.setText(null);
txtdept_id.setText(null);
p.removeAll();
frame.invalidate();
frame.validate();
frame.repaint();
p1=new JPanel();
p1.setLayout(new GridLayout(3,2));
p1.add(lbluser_id);
p1.add(txtuser_id);
p1.add(lblname);
p1.add(txtname);
p1.add(lblstaff_id);
p1.add(txtstaff_id);
p1.add(lblsubject);
p1.add(txtsubject);
p1.add(lbldept_id);
p1.add(txtdept_id);
p3=new JPanel(new FlowLayout());
p3.add(deleteButton);
p3.setBackground(Color.blue);
p3.setBounds(200,280,75,35);
p1.setBackground(Color.orange) ;
p1.setBounds(115,80,300,200);

p2 = new JPanel(new FlowLayout());
    staffidList=new List(10);
    loadstaffid();
    p2.add(staffidList);
    p2.setBackground(Color.cyan) ;
    p2.setBounds(125,320,300,180);

p.add(p1);
p.add(p3);
p.add(p2);

p.setLayout(new BorderLayout());

    frame.add(p);
    frame.setSize(800,800);
    frame.validate();
    staffidList.addItemListener(new ItemListener()
    {
        public void itemStateChanged(ItemEvent e)
        {
            try
            {
                rs=statement.executeQuery("select * from
faculty");
                while (rs.next())
                {

```

```

                                if
(rs.getString("STAFF_ID").equals(staffidList.getSelectedItem()))
                                break;
                                }
                                if (!rs.isAfterLast())
                                {

txtuser_id.setText(rs.getString("user_id"));

txtname.setText(rs.getString("name"));

txtsubject.setText(rs.getString("subject"));

txtstaff_id.setText(rs.getString("staff_id"));

txtdept_id.setText(rs.getString("dept_id"));

                                }
                                }
                                catch (SQLException selectException)
                                {
displaySQLErrors(selectException);
                                }
                                }
});

deleteButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
// TODO Auto-generated method stub
try {

String staffid=JOptionPane.showInputDialog(p,"Enter the
staff id");

txtstaff_id.setText(staffid);
int a=JOptionPane.showConfirmDialog(p,"Are you sure want to
delete:");
if(a==JOptionPane.YES_OPTION){
String query="DELETE FROM FACULTY WHERE
STAFF_ID="+txtstaff_id.getText()+" ";
int i=statement.executeUpdate(query);
JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows
succesfully");

loadstaffid();
}

}
catch(SQLException deleteException){
displaySQLErrors(deleteException);
}
}

```

```

        }

        });

    }
});

update.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        JButton updateButton = new JButton("Modify");
        txtuser_id.setText(null);
        txtname.setText(null);
        txtstaff_id.setText(null);
        txtsubject.setText(null);
        txtdept_id.setText(null);
        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();

        p1=new JPanel();

        p1.setLayout(new GridLayout(3,2));
        p1.add(lbluser_id);
        p1.add(txtuser_id);
        p1.add(lblname);
        p1.add(txtname);
        p1.add(lblstaff_id);
        p1.add(txtstaff_id);
        p1.add(lblsubject);
        p1.add(txtsubject);
        p1.add(lbldept_id);
        p1.add(txtdept_id);
        p3=new JPanel(new FlowLayout());
        p3.add(updateButton);
        //p1.add(txtf1);
        p3.setBackground(Color.blue);
        p3.setBounds(200,280,75,35);
        p1.setBackground(Color.orange) ;
        p1.setBounds(115,80,300,200);

        p2 = new JPanel(new FlowLayout());
        // p2.add(txtf1);
        staffidList=new List(10);
        loadstaffid();
        p2.add(staffidList);
        p2.setBackground(Color.cyan) ;
        p2.setBounds(125,320,300,180);
    }
});

```

```
p.add(p1);
p.add(p3);
p.add(p2);

p.setLayout(new BorderLayout());

frame.add(p);
frame.setSize(800,800);
frame.validate();
staffidList.addItemListener(new ItemListener()
{
    public void itemStateChanged(ItemEvent e)
    {
        try
        {
            rs=statement.executeQuery("select * from
faculty");

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

                txtuser_id.setText(rs.getString("user_id"));

                txtname.setText(rs.getString("name"));

                txtsubject.setText(rs.getString("subject"));

                txtstaff_id.setText(rs.getString("staff_id"));

                txtdept_id.setText(rs.getString("dept_id"));

            }
        }
        catch (SQLException selectException)
        {
            displaySQLErrors(selectException);
        }
    }
});

updateButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
    }
});

try {
    loadstaffid();
}
```

Student Assignments tracker

```
String query="update faculty set
name='"+txtname.getText()+"',subject='"+txtsubject.getText()+"',dept_id='"+txtdept_id.getText(
)+" where staff_id='"+txtstaff_id.getText()+" ";

int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nUpdated "+i+" rows
succesfully");

loadstaffid();

}
catch(SQLException updateException){

    displaySQLErrors(updateException);
}

}

});

view.addActionListener(new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub

        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();

        Label view1=new Label("Faculty view");
        //view1.setAlignment(Label.CENTER);
        Font myFont = new Font("Serif",Font.BOLD,50);
        view1.setFont((myFont));
        viewButton=new JButton("View");
        p1=new JPanel();
        p2=new JPanel();
        p1.add(view1);
        p2.add(viewButton);p1.setBackground(Color.cyan)
; p2.setBackground(Color.blue) ;
        p.add(p1);p.add(p2);
        p.setLayout(new FlowLayout());

        p.setBounds(500,800,300,300);
        frame.add(p);
        frame.setSize(800,800);
        frame.validate();
        viewButton.addActionListener(new ActionListener() {
            @Override
```

```
public void actionPerformed(ActionEvent e) {
    // TODO Auto-generated method stub
    JFrame f;

    JTable j;

    f = new JFrame();

    f.setTitle("Faculty details");

    DefaultTableModel model = new
DefaultTableModel();

    j = new JTable(model);
    model.addColumn("staff_id");

    model.addColumn("name");
    model.addColumn("subject");

    model.addColumn("dept_id");
    model.addColumn("user_id");

    try {

        rs=statement.executeQuery("select
* from faculty");

        while(rs.next()) {
            model.addRow(new
Object[]{rs.getString("staff_id"),rs.getString("name"),rs.getString("
dept_id"),rs.getString("user_id")}));
        }
    }
    catch(SQLException viewException) {
        displaySQLErrors(viewException);
    }
    j.setEnabled(false);
    j.setBounds(30, 40, 150, 150);

    JScrollPane sp = new JScrollPane(j);
    f.add(sp);

    f.setSize(400, 400);

    f.setVisible(true);

}
```

```
        });  
    }  
});  
}
```

ASSIGNMENTS:

```
package Sat;  
import java.awt.BorderLayout;  
import java.awt.Color;  
import java.awt.FlowLayout;  
import java.awt.Font;  
import java.awt.GridLayout;  
import java.awt.Label;  
import java.awt.List;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.awt.event.ItemEvent;  
import java.awt.event.ItemListener;  
import java.sql.*;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JMenuItem;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.JPasswordField;  
import javax.swing.JScrollPane;  
import javax.swing.JTable;  
import javax.swing.JTextArea;  
import javax.swing.JTextField;  
import javax.swing.table.DefaultTableModel;  
public class Assignments {  
    private static final long serialVersionUID = 1L;  
    private JButton insertButton,deleteButton,viewButton;  
    private JPanel p1,p2,p3,p;  
    private JLabel lblaname,lblpts,lblfeedback,lblrno;  
    private JTextField txtaname,txtpts,txtfeedback,txtrno;  
    private List anameList;  
    Connection con;  
    ResultSet rs;  
    Statement statement;  
    private JFrame frame;  
    private JMenuItem insert,delete,update,view;  
    public Assignments(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem  
update,JMenuItem view)
```



```

    {
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectToDB();
        this.frame=frame;
        this.insert=insert;
        this.delete=delete;
        this.update=update;
        this.view=view;
        lblrno=new JLabel("Roll Number");
        lblaname=new JLabel("Assignment Name");
        lblpts=new JLabel("Points");
        lblfeedback=new JLabel("Feedback");
        txttrno=new JTextField(10);
        txtaname=new JTextField(50);
        txtpts=new JTextField(50);
        txtfeedback=new JTextField(50);
        txttrno=new JTextField(10);
        this.p=p;
    }
    public void connectToDB()
    {
        try {
            Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","it19737121","vasavi");
            statement=con.createStatement();
            statement.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 displaySQLExceptions(SQLException e)
    {
        JOptionPane.showMessageDialog(p,"\nSQLException: " + e.getMessage() +
"\n"+"SQLState:      " + e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() + "\n");
    }
    public void loadaname() {
        try {
            anameList.removeAll();
            rs=statement.executeQuery("select assign_name from assignments");
            while(rs.next()) {
                anameList.add(rs.getString("assign_name"));
            }
        }
    }

```

```

    }
    }
    catch(SQLException e) {
        displaySQLErrors(e);
    }
}

public void buildGUI() {
    insertButton=new JButton("Submit");

    insert.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent arg0) {
            // TODO Auto-generated method stub
            txtaname.setText(null);
            txtpts.setText(null);
            txtfeedback.setText(null);
            txtrno.setText(null);
            p.removeAll();
            frame.invalidate();
            frame.validate();
            frame.repaint();
            p1=new JPanel();
            p1.setLayout(new GridLayout(4,1));
            p1.add(lblaname);
            p1.add(txtaname);
            p1.add(lblpts);
            p1.add(txtpts);
            p1.add(lblfeedback);
            p1.add(txtfeedback);
            p1.add(lblrno);
            p1.add(txtrno);
            p3=new JPanel(new FlowLayout());
            p3.add(insertButton);
            p3.setBackground(Color.blue);
            p3.setBounds(200,280,75,35);
            p1.setBackground(Color.orange) ;

            p1.setBounds(115,80,300,200);

            p2 = new JPanel(new FlowLayout());
            // p2.add(txtf1);
            anameList=new List(10);
            loadaname();
            p2.add(anameList);
            p2.setBackground(Color.cyan) ;
            p2.setBounds(125,320,300,180);

            p.add(p1);
            p.add(p3);
            p.add(p2);
        }
    });
}

```

```

        p.setLayout(new BorderLayout());

        frame.add(p);
        frame.setSize(800,800);
        frame.validate();
        insertButton.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub

                try {
                    String query="INSERT INTO ASSIGNMENTS
VALUES('"+txtaname.getText()+"', '"+txtpts.getText()+"', '"+txtfeedback.getText()+"', '"+txtrno.get
Text()+"')";

                    int i=statement.executeUpdate(query);
                    JOptionPane.showMessageDialog(p, "\nInserted "+i+" rows
succesfully");

                    loadaname();
                    System.out.println("Done");
                }
                catch(SQLException insertException){
                    displaySQLErrors(insertException);
                }

            }

        });

    }
});

delete.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        deleteButton=new JButton("Delete");

        txtaname.setText(null);
        txtpts.setText(null);
        txtfeedback.setText(null);
        txtrno.setText(null);
        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
        p1=new JPanel();
        p1.setLayout(new GridLayout(4,1));
        p1.add(lblaname);
        p1.add(txtaname);
        p1.add(lblpts);
        p1.add(txtpts);
    }
});

```

```

        p1.add(lblfeedback);
        p1.add(txtfeedback);
        p1.add(lblrno);
        p1.add(txtrno);
        p3=new JPanel(new FlowLayout());
        p3.add(deleteButton);
        p3.setBackground(Color.blue);
        p3.setBounds(200,280,75,35);
        p1.setBackground(Color.orange) ;

        p1.setBounds(115,80,300,200);

        p2 = new JPanel(new FlowLayout());
        anameList=new List(10);
        loadaname();
        p2.add(anameList);
        p2.setBackground(Color.cyan) ;
        p2.setBounds(125,320,300,180);

        p.add(p1);
        p.add(p3);
        p.add(p2);

        p.setLayout(new BorderLayout());

        frame.add(p);
        frame.setSize(800,800);
        frame.validate();
        anameList.addItemListener(new ItemListener()
        {
            public void itemStateChanged(ItemEvent e)
            {
                try
                {
                    rs=statement.executeQuery("select * from
assignments");
                    while (rs.next())
                    {
                        if
(rs.getString("assign_name").equals(anameList.getSelectedItem()))
                            break;
                    }
                    if (!rs.isAfterLast())
                    {
                        txtaname.setText(rs.getString("assign_name"));

                        txtpts.setText(rs.getString("points"));

                        txtfeedback.setText(rs.getString("feedback"));

                        txtrno.setText(rs.getString("rollnumber"));

```

```

    }
    }
    catch (SQLException selectException)
    {
        displaySQLErrors(selectException);
    }
    });

deleteButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
    try {

        String aname=JOptionPane.showInputDialog(p,"Enter the
assignment name");

        txtaname.setText(aname);
        int a=JOptionPane.showConfirmDialog(p,"Are you sure want to
delete:");

        if(a==JOptionPane.YES_OPTION){
            String query="DELETE FROM ASSIGNMENTS WHERE
assign_name='"+txtaname.getText()+"' ";
            int i=statement.executeUpdate(query);
            JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows
succesfully");

            loadaname();
        }

        catch(SQLException deleteException){
            displaySQLErrors(deleteException);
        }

    }

    });

update.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        JButton updateButton = new JButton("Modify");
        txtaname.setText(null);
        txtpts.setText(null);
        txtfeedback.setText(null);
    }
});

```

```
        ttxttrno.setText(null);
        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();

        p1=new JPanel();

        p1.setLayout(new GridLayout(4,1));
        p1.add(lblaname);
        p1.add(txtaname);
        p1.add(lblpts);
        p1.add(txtpts);
        p1.add(lblfeedback);
        p1.add(txtfeedback);
        p1.add(lblrno);
        p1.add(txttrno);
        p3=new JPanel(new FlowLayout());
        p3.add(updateButton);
        //p1.add(txtf1);
        p3.setBackground(Color.blue);
        p3.setBounds(200,280,75,35);
        p1.setBackground(Color.orange );

        p1.setBounds(115,80,300,200);

        p2 = new JPanel(new FlowLayout());
        // p2.add(txtf1);
        anameList=new List(10);
        loadaname();
        p2.add(anameList);
        p2.setBackground(Color.cyan) ;
        p2.setBounds(125,320,300,180);

        p.add(p1);
        p.add(p3);
        p.add(p2);

        p.setLayout(new BorderLayout());

        frame.add(p);
        frame.setSize(800,800);
        frame.validate();
        anameList.addItemListener(new ItemListener()
        {
            public void itemStateChanged(ItemEvent e)
            {
                try
                {
                    rs=statement.executeQuery("select * from
assignments");
```

```

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

txtaname.setText(rs.getString("assign_name"));

txtpts.setText(rs.getString("points"));

txtfeedback.setText(rs.getString("feedback"));

txtrno.setText(rs.getString("rollnumber"));

                                }
                                }
                                catch (SQLException selectException)
                                {
                                    displaySQLErrors(selectException);
                                }
                            });

p.setLayout(new BorderLayout());

frame.add(p);
frame.setSize(800,800);
frame.validate();
updateButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        try {
            loadaname();
            String query="update assignments set
points="+txtpts.getText()+",feedback='"+txtfeedback.getText()+"',rollnumber="+txtrno.getText(
)+" WHERE assign_name='"+txtaname.getText()+"' ";

            int i=statement.executeUpdate(query);

JOptionPane.showMessageDialog(p,"\nUpdated "+i+" rows
succesfully");

            loadaname();

        }
        catch(SQLException updateException){

            displaySQLErrors(updateException);
        }
    }
});
```

```
        }
    }

    });

    view.addActionListener(new ActionListener(){

        @Override
        public void actionPerformed(ActionEvent arg0) {
            // TODO Auto-generated method stub

            p.removeAll();
            frame.invalidate();
            frame.validate();
            frame.repaint();

            Label view1=new Label("Assignments view");
            //view1.setAlignment(Label.CENTER);
            Font myFont = new Font("Serif",Font.BOLD,50);
            view1.setFont(myFont);
            viewButton=new JButton("View");
            p1=new JPanel();
            p2=new JPanel();
            p1.add(view1);
            p2.add(viewButton);p1.setBackground(Color.cyan)
; p2.setBackground(Color.blue) ;
            p.add(p1);p.add(p2);
            p.setLayout(new FlowLayout());

            p.setBounds(500,800,300,300);
            frame.add(p);
            frame.setSize(800,800);
            frame.validate();
            viewButton.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent e) {
                    // TODO Auto-generated method stub
                    JFrame f;

                    JTable j;

                    f = new JFrame();

                    f.setTitle("Assignment details");

                    DefaultTableModel model = new
DefaultTableModel();

                    j = new JTable(model);
```



```
model.addColumn("assign_name");

model.addColumn("points");

model.addColumn("feedback");
model.addColumn("rollnumber");

try {

    rs=statement.executeQuery("select
* from assignments");

    while(rs.next()) {
        model.addRow(new
Object[]{rs.getString("assign_name"),rs.getString("points"),rs.getString("feedback"),rs.getSt
ring("rollnumber")});
    }
} catch(SQLException viewException) {
    displaySQLErrors(viewException);
}
j.setEnabled(false);
j.setBounds(30, 40, 150, 150);

JScrollPane sp = new JScrollPane(j);
f.add(sp);

f.setSize(400, 400);

f.setVisible(true);

}

});

}

}
```

MANAGES:

```
package Sat;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.sql.*;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;
public class manages{
    private static final long serialVersionUID = 1L;
    private JButton insertButton,deleteButton,viewButton;
    private JPanel p1,p2,p3,p;
    private JLabel lblsection,lblstaff_id,lblaname,lblsub;
    private JTextField txtsection,txtstaff_id,txtaname,txtsub;
    private List staff_idList;
    Connection con;
    ResultSet rs;
    Statement statement;
    private JFrame frame;
    private JMenuItem insert,delete,update,view;
    public manages(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem
update,JMenuItem view)
    {
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectToDB();
        this.frame=frame;
        this.insert=insert;
```

```

        this.delete=delete;
        this.update=update;
        this.view=view;
        lblsection=new JLabel("Section");
        lblstaff_id=new JLabel("Staff ID");
        lblsub=new JLabel("Subject");
        lblaname=new JLabel("Assignment Name");
        txtsection=new JTextField(50);
        txtstaff_id=new JTextField(50);
        txtsub=new JTextField(50);
        txtaname=new JTextField(50);
        this.p=p;
    }
    public void connectToDB()
    {
        try {
            Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","it19737121","vasavi");
            statement=con.createStatement();
            statement.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(p,"\nSQLException: " + e.getMessage() +
"\n"+"SQLState:      " + e.getSQLState() + "\n"+"VendorError:  " + e.getErrorCode() + "\n");
    }
    public void loadstaff_id() {
        try {
            staff_idList.removeAll();
            rs=statement.executeQuery("select staff_id from manages");
            while(rs.next()) {
                staff_idList.add(rs.getString("staff_id"));
            }
        }
        catch(SQLException e) {
            displaySQLErrors(e);
        }
    }

    public void buildGUI() {
        insertButton=new JButton("Submit");

        insert.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent arg0) {

```

```

// TODO Auto-generated method stub
txtsection.setText(null);
txtstaff_id.setText(null);
txtsub.setText(null);
txtaname.setText(null);
p.removeAll();
frame.invalidate();
frame.validate();
frame.repaint();
p1=new JPanel();
p1.setLayout(new GridLayout(4,2));
p1.add(lblsection);
p1.add(txtsection);
p1.add(lblstaff_id);
p1.add(txtstaff_id);
p1.add(lblsub);
p1.add(txtsub);
p1.add(lblaname);
p1.add(txtaname);
p3=new JPanel(new FlowLayout());
p3.add(insertButton);
p3.setBackground(Color.blue);
p3.setBounds(200,280,75,35);
p1.setBackground(Color.orange) ;

p1.setBounds(115,80,300,200);

p2 = new JPanel(new FlowLayout());
// p2.add(txtf1);
staff_idList=new List(10);
loadstaff_id();
p2.add(staff_idList);
p2.setBackground(Color.cyan) ;
p2.setBounds(125,320,300,180);

p.add(p1);
p.add(p3);
p.add(p2);

p.setLayout(new BorderLayout());

frame.add(p);
frame.setSize(800,800);
frame.validate();
insertButton.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
    }
}

```

```

        String query="INSERT INTO MANAGES
VALUES('"+txtsection.getText()+"', '"+txtstaff_id.getText()+"', '"+txtaname.getText()+"', '"+txtsub
b.getText()+"')";

        int i=statement.executeUpdate(query);
        JOptionPane.showMessageDialog(p, "\nInserted "+i+" rows
succesfully");

        loadstaff_id();
        System.out.println("Done");
    }
    catch(SQLException insertException){
        displaySQLErrors(insertException);
    }

    }

    });

}
});

delete.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        deleteButton=new JButton("Delete");

        txtsection.setText(null);
        txtstaff_id.setText(null);
        txtsub.setText(null);
        txtaname.setText(null);
        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();
        p1=new JPanel();
        p1.setLayout(new GridLayout(4,2));
        p1.add(lblsection);
        p1.add(txtsection);
        p1.add(lblstaff_id);
        p1.add(txtstaff_id);
        p1.add(lblsub);
        p1.add(txtsub);
        p1.add(lblaname);
        p1.add(txtaname);
        p3=new JPanel(new FlowLayout());
        p3.add(deleteButton);
        p3.setBackground(Color.blue);
        p3.setBounds(200,280,75,35);
        p1.setBackground(Color.orange) ;

        p1.setBounds(115,80,300,200);

        p2 = new JPanel(new FlowLayout());
    }
});

```

```
        staff_idList=new List(10);
        loadstaff_id();
        p2.add(staff_idList);
        p2.setBackground(Color.cyan) ;
        p2.setBounds(125,320,300,180);

        p.add(p1);
        p.add(p3);
        p.add(p2);

        p.setLayout(new BorderLayout());

        frame.add(p);
        frame.setSize(800,800);
        frame.validate();
        staff_idList.addItemListener(new ItemListener()
        {
            public void itemStateChanged(ItemEvent e)
            {
                try
                {
                    rs=statement.executeQuery("select * from
manages");

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

                        txtaname.setText(rs.getString("section"));

                        txtstaff_id.setText(rs.getString("staff_id"));

                        txtsection.setText(rs.getString("assign_name"));

                        txtsub.setText(rs.getString("subject"));

                    }
                }
                catch (SQLException selectException)
                {
                    displaySQLErrors(selectException);
                }
            }
        });

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

```

        @Override
        public void actionPerformed(ActionEvent e) {
            // TODO Auto-generated method stub
            try {
                delete:");
                int a=JOptionPane.showConfirmDialog(p,"Are you sure want to
                if(a==JOptionPane.YES_OPTION){
                    String query="DELETE FROM MANAGES WHERE
                    staff_id="+staff_idList.getSelectedItemAt();
                    int i=statement.executeUpdate(query);
                    JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows
                    successfully");
                    loadstaff_id();
                }
            }
            catch(SQLException deleteException){
                displaySQLErrors(deleteException);
            }
        }

    });

}
});

update.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        JButton updateButton = new JButton("Modify");
        txtsection.setText(null);
        txtstaff_id.setText(null);
        txtsub.setText(null);
        txtaname.setText(null);
        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();

        p1=new JPanel();

        p1.setLayout(new GridLayout(4,2));
        p1.add(lblsection);
        p1.add(txtsection);
        p1.add(lblstaff_id);
        p1.add(txtstaff_id);
        p1.add(lblsub);
        p1.add(txtsub);
        p1.add(lblaname);
        p1.add(txtaname);
        p3=new JPanel(new FlowLayout());
    }
});

```

```

        p3.add(updateButton);
        //p1.add(txtf1);
        p3.setBackground(Color.blue);
        p3.setBounds(200,280,75,35);
        p1.setBackground(Color.orange) ;

        p1.setBounds(115,80,300,200);

        p2 = new JPanel(new FlowLayout());
        // p2.add(txtf1);
        staff_idList=new List(10);
        loadstaff_id();
        p2.add(staff_idList);
        p2.setBackground(Color.cyan) ;
        p2.setBounds(125,320,300,180);

        p.add(p1);
        p.add(p3);
        p.add(p2);

        staff_idList.addItemListener(new ItemListener()
        {
            public void itemStateChanged(ItemEvent e)
            {
                try
                {
                    rs=statement.executeQuery("select * from
manages");

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

                        txtsection.setText(rs.getString("section"));

                        txtstaff_id.setText(rs.getString("staff_id"));

                        txtaname.setText(rs.getString("assign_name"));

                        txtsub.setText(rs.getString("subject"));

                    }
                }
                catch (SQLException selectException)
                {
                    displaySQLErrors(selectException);
                }
            }
        }
    }

```



```

    });

    p.setLayout(new BorderLayout());

    frame.add(p);
    frame.setSize(800,800);
    frame.validate();
    updateButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // TODO Auto-generated method stub

            try {
                loadstaff_id();
                String query="update manages set
section='"+txtsection.getText()+"',assign_name='"+txtaname.getText()+"',subject='"+txtsub.get
Text()+"' WHERE staff_id='"+txtstaff_id.getText()+"' ";

                int i=statement.executeUpdate(query);

                JOptionPane.showMessageDialog(p,"\nUpdated "+i+" rows
succesfully");

                loadstaff_id();

            }
            catch(SQLException updateException){

                displaySQLErrors(updateException);
            }

        }

    });

}
});
view.addActionListener(new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub

        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();

        Label view1=new Label("Manages view");
        //view1.setAlignment(Label.CENTER);
        Font myFont = new Font("Serif",Font.BOLD,50);
        view1.setFont((myFont));
        viewButton=new JButton("View");
        p1=new JPanel();
        p2=new JPanel();
        p1.add(view1);

```

Student Assignments tracker

```
p2.add(viewButton);p1.setBackground(Color.cyan)
;p2.setBackground(Color.blue) ;
p.add(p1);p.add(p2);
p.setLayout(new FlowLayout());

p.setBounds(500,800,300,300);
frame.add(p);
frame.setSize(800,800);
frame.validate();
viewButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        JFrame f;

        JTable j;

        f = new JFrame();

        f.setTitle("Assignments");

        DefaultTableModel model = new
DefaultTableModel();

        j = new JTable(model);

        model.addColumn("section");
        model.addColumn("staff_id");
        model.addColumn("assign_name");
        model.addColumn("subject");
        try {

            rs=statement.executeQuery("select
* from manages");

            while(rs.next()) {
                model.addRow(new
Object[]{rs.getString("section"),rs.getString("staff_id"),rs.getSt
tring("subject")}));
            }
        }
        catch(SQLException viewException) {
            displaySQLErrors(viewException);
        }
        j.setEnabled(false);
        j.setBounds(30, 40, 150, 150);

        JScrollPane sp = new JScrollPane(j);
        f.add(sp);

        f.setSize(400, 400);
```

```
        f.setVisible(true);

    }

});

}

});

}
```

VIEWS:

```
package Sat;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.List;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.sql.*;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;
public class Views {
    private static final long serialVersionUID = 1L;
    private JButton insertButton,deleteButton,viewButton;
    private JPanel p1,p2,p3,p;
```

```

    private JLabel lblrno,lbldue,lbldiv,blaname;
    private JTextField txtrno,txtdue,txtdiv,txtaname;
    private List rnoList;
    Connection con;
    ResultSet rs;
    Statement statement;
    private JFrame frame;
    private JMenuItem insert,delete,update,view;
    public Views(JPanel p,JFrame frame,JMenuItem insert,JMenuItem delete,JMenuItem
update,JMenuItem view)
    {
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e)
        {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        connectToDB();
        this.frame=frame;
        this.insert=insert;
        this.delete=delete;
        this.update=update;
        this.view=view;
        lblrno=new JLabel("Roll Number");
        lbldue=new JLabel("Due Date");
        lbldiv=new JLabel("Division");
        blaname=new JLabel("Assignment Name");
        txtrno=new JTextField(5);
        txtdue=new JTextField(12);
        txtdiv=new JTextField(3);
        txtaname=new JTextField(20);
        this.p=p;
    }
    public void connectToDB()
    {
        try {
            Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","it19737121","vasavi");
            statement=con.createStatement();
            statement.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(p, "\nSQLException: " + e.getMessage() +
"\n" + "SQLState: " + e.getSQLState() + "\n" + "VendorError: " + e.getErrorCode() + "\n");
}
public void loadrno()
{
    try
    {
        rnoList.removeAll();
        rs=statement.executeQuery("select rollnumber from views");
        while(rs.next())
        {
            rnoList.add(rs.getString("rollnumber"));
        }
    }
    catch(SQLException e) {
        displaySQLErrors(e);
    }
}

public void buildGUI() {
    insertButton=new JButton("Submit");

    insert.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent arg0) {
            // TODO Auto-generated method stub
            txttrno.setText(null);
            txtdue.setText(null);
            txtdiv.setText(null);
            txtaname.setText(null);
            p.removeAll();
            frame.invalidate();
            frame.validate();
            frame.repaint();
            p1=new JPanel();
            p1.setLayout(new GridLayout(4,2));
            p1.add(lblrno);
            p1.add(txttrno);
            p1.add(lbldue);
            p1.add(txtdue);
            p1.add(lbldiv);
            p1.add(txtdiv);
            p1.add(lblaname);
            p1.add(txtaname);
            p3=new JPanel(new FlowLayout());
            p3.add(insertButton);
            p3.setBackground(Color.blue);
            p3.setBounds(200,280,75,35);
            p1.setBackground(Color.orange) ;

            p1.setBounds(115,80,300,200);
```

```
p2 = new JPanel(new FlowLayout());
// p2.add(txtf1);
rnoList=new List(10);
loadrno();
p2.add(rnoList);
p2.setBackground(Color.cyan) ;
p2.setBounds(125,320,300,180);

p.add(p1);
p.add(p3);
p.add(p2);

p.setLayout(new BorderLayout());

frame.add(p);
frame.setSize(800,800);
frame.validate();
insertButton.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            String query="INSERT INTO VIEWS
VALUES("+txtrno.getText()+", '"+txtdue.getText()+"', '"+txtdiv.getText()+"', '"+txtaname.getText
()+"'");

            int i=statement.executeUpdate(query);
            JOptionPane.showMessageDialog(p,"\\nInserted "+i+" rows
succesfully");

            loadrno();
            System.out.println("Done");
        }
        catch(SQLException insertException){
            displaySQLErrors(insertException);
        }

    }

});

delete.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        deleteButton=new JButton("Delete");

        txtrno.setText(null);
```

```

txtdue.setText(null);
txtdiv.setText(null);
txtaname.setText(null);
p.removeAll();
frame.invalidate();
frame.validate();
frame.repaint();
p1=new JPanel();
p1.setLayout(new GridLayout(4,2));

p1.add(lblrno);
p1.add(txtrno);
p1.add(lbldue);
p1.add(txtdue);
p1.add(lbldiv);
p1.add(txtdiv);
p1.add(lblaname);
p1.add(txtaname);
p3=new JPanel(new FlowLayout());
p3.add(deleteButton);
p3.setBackground(Color.blue);
p3.setBounds(200,280,75,35);
p1.setBackground(Color.orange) ;

p1.setBounds(115,80,300,200);

p2 = new JPanel(new FlowLayout());
    rnoList=new List(10);
    loadrno();
    p2.add(rnoList);
    p2.setBackground(Color.cyan) ;
    p2.setBounds(125,320,300,180);

p.add(p1);
p.add(p3);
p.add(p2);

p.setLayout(new BorderLayout());

    frame.add(p);
    frame.setSize(800,800);
    frame.validate();
    rnoList.addItemListener(new ItemListener()
    {
        public void itemStateChanged(ItemEvent e)
        {
            try
            {
                rs=statement.executeQuery("select * from
views");

                while (rs.next())
                {

```

```

                                if
(rs.getString("rollnumber").equals(rnoList.getSelectedItem()))
                                break;
                                }
                                if (!rs.isAfterLast())
                                {

txttrno.setText(rs.getString("rollnumber"));

txtdue.setText(rs.getString("due_date"));

txtdiv.setText(rs.getString("division"));

txtaname.setText(rs.getString("assign_name"));

                                }
                                }
                                catch (SQLException selectException)
                                {
                                    displaySQLErrors(selectException);
                                }
                            }
                        });

deleteButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        try {

            // aname=JOptionPane.showInputDialog(p,"Enter the assignment
name");

            //txtaname.setText(aname);
            int a=JOptionPane.showConfirmDialog(p,"Are you sure want to
delete:");

            if(a==JOptionPane.YES_OPTION){
                String query="DELETE FROM VIEWS WHERE
rollnumber="+rnoList.getSelectedItem();
                int i=statement.executeUpdate(query);
                JOptionPane.showMessageDialog(p,"\nDeleted "+i+" rows
succesfully");

                loadrno();
            }

        }
        catch(SQLException deleteException){
            displaySQLErrors(deleteException);
        }
    }
});

```



```

        });
    }
});

update.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        JButton updateButton = new JButton("Modify");
        txttrno.setText(null);
        txtdue.setText(null);
        txtdiv.setText(null);
        txtaname.setText(null);
        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();

        p1=new JPanel();

        p1.setLayout(new GridLayout(4,2));
        p1.add(lblrno);
        p1.add(txttrno);
        p1.add(lbldue);
        p1.add(txtdue);
        p1.add(lbldiv);
        p1.add(txtdiv);
        p1.add(lblaname);
        p1.add(txtaname);
        p3=new JPanel(new FlowLayout());
        p3.add(updateButton);
        p3.setBackground(Color.blue);
        p3.setBounds(200,280,75,35);
        p1.setBackground(Color.orange) ;

        p1.setBounds(115,80,300,200);

        p2 = new JPanel(new FlowLayout());
        // p2.add(txtf1);
        rnoList=new List(10);
        loadrno();
        p2.add(rnoList);
        p2.setBackground(Color.cyan) ;
        p2.setBounds(125,320,300,180);

        p.add(p1);
        p.add(p3);
        p.add(p2);
    }
});

```

```
p.setLayout(new BorderLayout());

frame.add(p);
frame.setSize(800,800);
frame.validate();
rnoList.addItemListener(new ItemListener()
{
    public void itemStateChanged(ItemEvent e)
    {
        try
        {
            rs=statement.executeQuery("select * from
views");

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

                txtrno.setText(rs.getString("rollnumber"));

                txtdue.setText(rs.getString("due_date"));
                txtdiv.setText(rs.getString("division"));
                txtaname.setText(rs.getString("assign_name"));

            }
        }
        catch (SQLException selectException)
        {
            displaySQLErrors(selectException);
        }
    }
});

updateButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
try {
    loadrno();

    String query="update views set
due_date='"+txtdue.getText()+"',division='"+txtdiv.getText()+"',assign_name='"+txtaname.getTe
xt()+"' where rollnumber='"+txtrno.getText()+" ";

    int i=statement.executeUpdate(query);
```

```

        JOptionPane.showMessageDialog(p, "\nUpdated "+i+" rows
        successfully");

        loadrno();

    }
    catch(SQLException updateException){

        displaySQLErrors(updateException);

    }

    }

    });

}
});
view.addActionListener(new ActionListener(){

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub

        p.removeAll();
        frame.invalidate();
        frame.validate();
        frame.repaint();

        Label view1=new Label("Student assignments view");
        //view1.setAlignment(Label.CENTER);
        Font myFont = new Font("Serif",Font.BOLD,50);
        view1.setFont((myFont));
        viewButton=new JButton("View");
        p1=new JPanel();
        p2=new JPanel();
        p1.add(view1);
        p2.add(viewButton);
        p1.setBackground(Color.cyan) ;
        p2.setBackground(Color.blue) ;
        p.add(p1);p.add(p2);
        p.setLayout(new FlowLayout());

        p.setBounds(500,800,300,300);
        frame.add(p);
        frame.setSize(800,800);
        frame.validate();
        viewButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
                JFrame f;

```

Student Assignments tracker

```
        jTable j;  
        f = new JFrame();  
  
        f.setTitle("Student Assignments details");  
  
        DefaultTableModel model = new  
DefaultTableModel();  
  
        j = new JTable(model);  
        model.addColumn("rollnumber");  
  
        model.addColumn("due_date");  
  
        model.addColumn("division");  
        model.addColumn("assign_name");  
  
        try {  
            rs=statement.executeQuery("select  
* from views");  
            while(rs.next()) {  
                model.addRow(new  
Object[]{rs.getString("rollnumber"),rs.getString("due_date"),rs.getString("division"),rs.getS  
tring("assign_name")});  
            }  
        } catch(SQLException viewException) {  
            displaySQLErrors(viewException);  
        }  
        j.setEnabled(false);  
        j.setBounds(30, 40, 150, 150);  
  
        JScrollPane sp = new JScrollPane(j);  
        f.add(sp);  
  
        f.setSize(400, 400);  
  
        f.setVisible(true);  
  
    }  
  
});
```

```
        }  
    });  
}  
}
```

Github links and folder structure:

Link:

Folder structure:

This PC > OS (C:) > javaprograms > Assignment












Name	Date modified	Type	Size
.settings	15-06-2021 17:31	File folder	
bin	15-06-2021 17:41	File folder	
src	15-06-2021 17:32	File folder	
.classpath	15-06-2021 17:41	CLASSPATH File	1 KB
.project	15-06-2021 17:31	PROJECT File	1 KB

This PC > OS (C:) > javaprograms > Assignment > src

Name	Date modified	Type	Size
Sat	17-06-2021 19:09	File folder	

Student Assignments tracker

This PC > OS (C:) > javaprograms > Assignment > src > Sat

Name	Date modified	Type	Size
 Assignments	18-06-2021 18:37	JAVA File	13 KB
 connecttoDB	16-06-2021 21:45	JAVA File	1 KB
 DBconnection	17-06-2021 19:11	JAVA File	1 KB
 Faculty	18-06-2021 18:34	JAVA File	14 KB
 login	18-06-2021 18:31	JAVA File	14 KB
 loginpage	18-06-2021 18:45	JAVA File	4 KB
 Main	15-06-2021 17:36	JAVA File	1 KB
 manages	18-06-2021 19:33	JAVA File	13 KB
 Students	18-06-2021 18:33	JAVA File	14 KB
 UserInterface	18-06-2021 19:27	JAVA File	7 KB
 Views	18-06-2021 18:39	JAVA File	13 KB

TESTING

OUTPUT SCREENSHOTS:



Student Assignments Tracker

Login

User ID

Password

Student Assignments tracker

Student Assignments Tracker

Login Faculty Manages Students Assignments Views

STUDENT ASSIGNMENTS TRACKER

User_id	abhiram_312
Password
User Category	student

Submit

Message

Inserted 1 rows succesfully

OK

akshith1
charan_
kittu987
nithin
ravi214
varun_121
vijay576
vikram199

Student Assignments tracker

User_id	abhiram_312
Password
User Category	student

Delete

abhiram_312
akshith100
charan_001
kittu987
nithin
ravi214
varun_123
vijay576
vikram199

Message

Deleted 1 rows succesfully

OK

Student Assignments tracker



Login details



user_id	password	usercategory
charan_006	def1465	student
ravi214	pqr9852	staff
vijay576	kzyw77	staff
kittu987	mnp771	student
akshith1098	ghij001	student
nithin	ijk9130	admin
varun_121	abc789	student
vikram199	a9bnjk	staff

iew

View

Student Assignments tracker

 Student Assignments Tracker

Login

User ID

Password

Login

Message



INVALID LOGIN

OK

 Student Assignments Tracker


Login

User ID

Password


Login

Student Assignments tracker

 Student Assignments Tracker

Login Students Assignments Views

STUDENT ASSIGNMENTS TRACKER

 Students details

name	rollnumber	division	dept_id	user_id
akshith	102	B	737	akshith1098
varun	121	b	737	varun_121
charan	6	a	733	charan_006

view

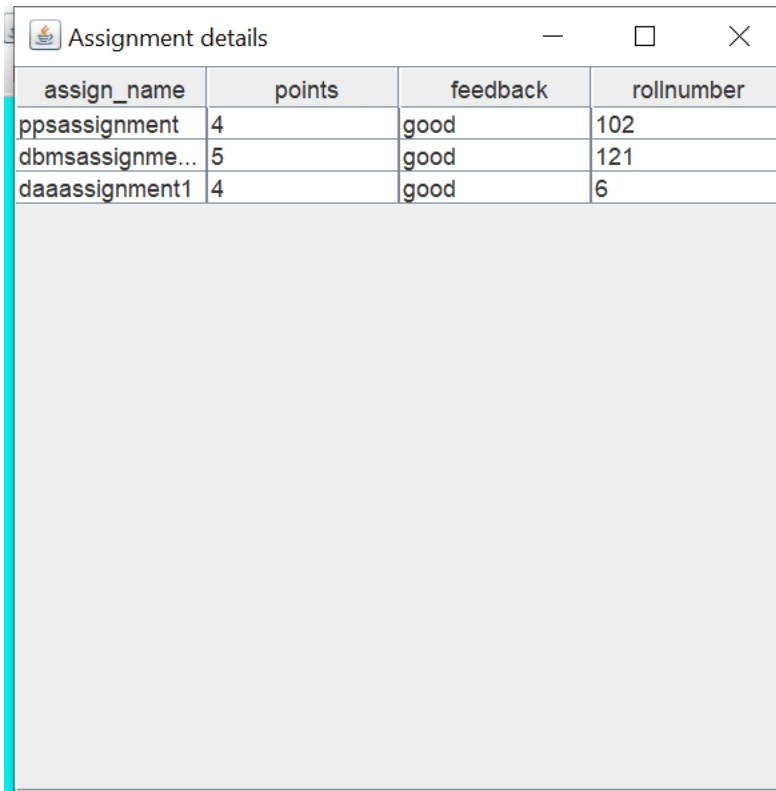
View

Login Students Assignments Views

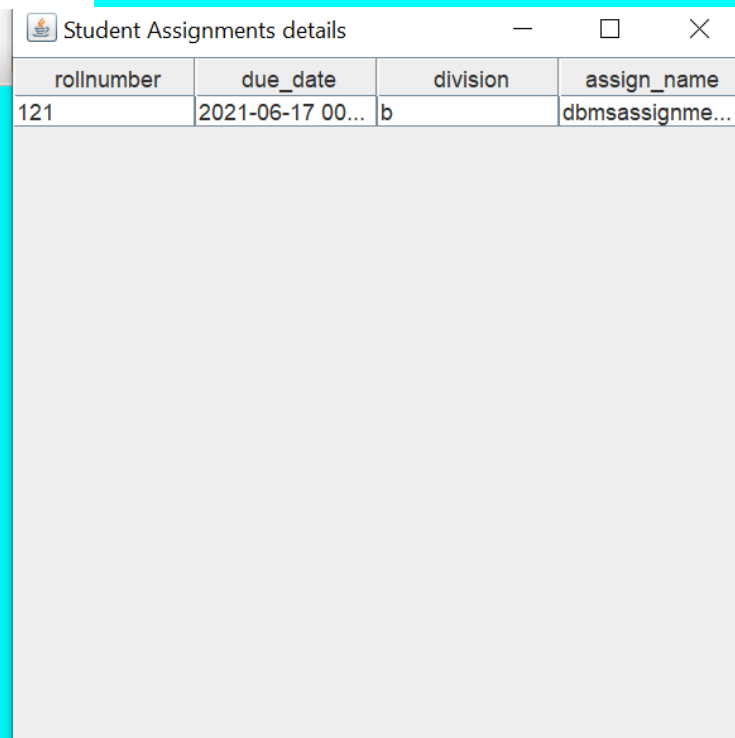
Assignments view

View

Student Assignments tracker

A screenshot of a web application window titled "Assignment details". It contains a table with four columns: assign_name, points, feedback, and rollnumber. The table lists three assignments: ppsassignment (4 points, good feedback, rollnumber 102), dbmsassignme... (5 points, good feedback, rollnumber 121), and daaassignment1 (4 points, good feedback, rollnumber 6).

assign_name	points	feedback	rollnumber
ppsassignment	4	good	102
dbmsassignme...	5	good	121
daaassignment1	4	good	6

A screenshot of a web application window titled "Student Assignments details". It contains a table with four columns: rollnumber, due_date, division, and assign_name. The table lists one assignment for rollnumber 121: due_date 2021-06-17 00..., division b, and assign_name dbmsassignme...

rollnumber	due_date	division	assign_name
121	2021-06-17 00...	b	dbmsassignme...

ments view

View

Login Faculty Manages Students Assignments Views

User_id

ravi214

Password

.....

User Category

staff

Modify

Input



Enter the new password

abc7891

OK

Cancel

Student Assignments tracker

User_id	ravi214
Password
User Category	staff

Modify

ravi214

Message

Updated 1 rows succesfully

OK


Student Assignments tracker

Section	A
Staff ID	73352
Subject	daa
Assignment Name	daaassignment1

Submit

73701

Message

 Inserted 1 rows succesfully

OK

Student Assignments tracker

section	staff_id	assign_name	subject
B	73701	dbmsassignme...	dbms
A	73352	daaassignment1	daa

Manages view

[View](#)

name	rollnumber	division	dept_id	user_id
akshith	102	B	737	akshith1098
varun	121	b	737	varun_121
charan	6	a	733	charan_006

Students view

[View](#)

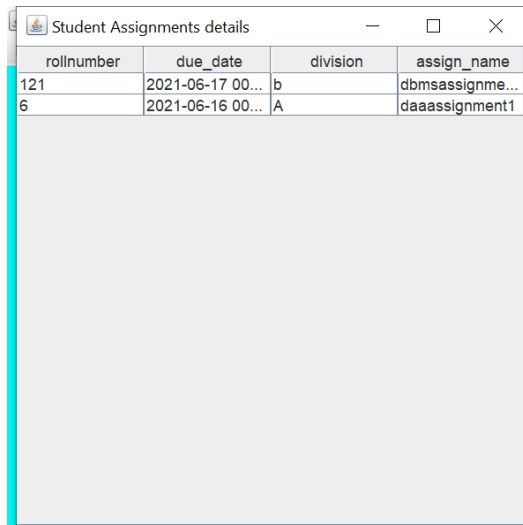
Student Assignments tracker

Assignment details			
assign_name	points	feedback	rollnumber
ppsassignment	4	good	102
dbmsassignme...	5	good	121
daaassignment1	4	good	6

ts view

View

Student Assignments tracker



rollnumber	due_date	division	assign_name
121	2021-06-17 00...	b	dbmsassignme...
6	2021-06-16 00...	A	daaassignment1

Student assignments view

[View](#)

Roll Number	006
Due Date	16-JUN-2021
Division	A
Assignment Name	daaassignment1

[Submit](#)

121

Message



Inserted 1 rows succesfully

[OK](#)

Student Assignments tracker

Assignment Name	ppsassignment
Points	4.5
Feedback	good
Roll Number	102

Modify

daaassignment
dbmsassignment
ppsassignment

Message

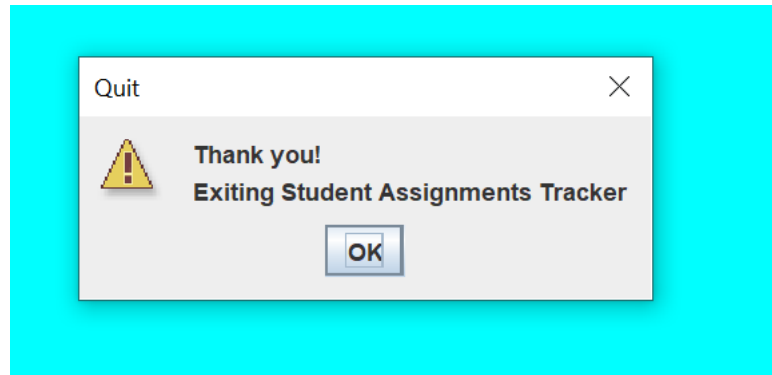
Updated 1 rows succesfully

OK

Select an Option

Are you sure to Quit Student Assignments Tracker:

Yes No Cancel



DATABASE OUTPUTS:

BEFORE

```
SQL> select * from manages;

SECTI   STAFF_ID ASSIGN_NAME
-----
SUBJECT
-----
B          73701 dbmsassignment1
dbms
```

AFTER

```
SQL> select * from manages;

SECTI   STAFF_ID ASSIGN_NAME
-----
SUBJECT
-----
B          73701 dbmsassignment1
dbms

A          73352 daaassignment1
daa
```

DISCUSSION AND FUTURE WORK

So far this project has helped us assign assignments to students by staff and students will be able to view assignments their marks, due dates and feedback

In future we try to include an option which allows students to know their mistakes in their assignments and staff can highlight the mistakes by students in their respective submissions.

REFERENCES:

1. Abraham Silberschatz, Henry F Korth, S. Sudarshan, Database System Concepts, 6th Edition, McGraw-Hill International Edition, 2010.
2. https://docs.oracle.com/cd/E11882_01/server.112/e40540/intro.htm#CNCPT001