HOBBIES DATABASE

JAVA SWING BASED – HOBBIES DATABASE – SQL CONNECTIVITY USING JDBC

A Report
Submitted in partial fulfillment of the Requirements

for the COURSE

DATABASE MANAGEMENT SYSTEMS

By

AKHIL **1602-21-737-005 Under the guidance of Ms B. Leelavathy**

Department of Information Technology Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University) Ibrahimbagh, Hyderabad-31 2023-2024

BONAFIDE CERTIFICATE

This is to certify that this project report titled

'HOBBIES DATABASE'

is a project work of **AKHIL** bearing roll no. 1602-21-737-005 who carried out this project under my supervision in the IV semester for

the academic year 2022- 2023

Signature External Examiner

Signature Internal Examiner

1602-20-737-005 Akhil

HOBBIES DATABASE

AIM AND PRIORITY OF THE PROJECT

To create a **Java GUI-based** desktop application that connects students looking for career choices with skills and Interest. It takes values like student name, username, Age, Skills, etc through forms which are then updated in the database using JDBC connectivity.

ARCHITECTURE AND TECHNOLOGY

Software used:

Java, Oracle 11g Database, Java SE version 14, Run SQL.

Java SWING:

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

Swing was developed to provide a more sophisticated set of GUI components than the earlier AWT. Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

HOBBIES DATABASE

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.

1602-21-737-005

Akhil

Abstract:

The goal of this project is to develop an ER model for a database that helps students become successful software developers by providing them with suitable hobbies. The model will include entities such as students, hobbies, skills, projects and will use associative entities to connect them. The resulting database will allow students to explore and select hobbies that align with their interests and will help them develop skills that are relevant to software development. The database will also provide information on projects that can further enhance their skills and knowledge in the field. The project will involve designing the ER model, creating the necessary tables, and writing DML queries to insert and retrieve data from the database.

1602-21-737-005

Akhil

HOBBIES DATABASE

Requirements:

Tables that I have identified

are :Student,Hobby,Skill,Project,Student_Hobby,Student_Skill,Student_project,skill_hobby.

1.Student Table:

ATTRIBUTE	DOMAIN	CONSTRAINT
Student_id	NUMBER	Primary Key
Student_name	VARCHAR	Not Null
Student_email	VARCHAR	
Student_phone	VARCHAR	

2.Hobby Table:

ATTRIBUTE	DOMAIN	CONSTRAINT
Hobby_id	NUMBER	Primary Key
Hobby_name	VARCHAR	
Hobby_description	VARCHAR	

3.Skill_hobby Table:

ATTRIBUTE	DOMAIN	CONSTRAINT
Skill_id	NUMBER	Foreign key
hobby_id	NUMBER	foreign key

4.Skill Table:

ATTRIBUTE	DOMAIN	CONSTRAINT
Skill_id	NUMBER	Primary Key
Skill_Name	VARCHAR	
Skill_Description	VARCHAR	

5.Project Table:

ATTRIBUTE	DOMAIN	CONSTRAINT
Project_ID	NUMBER	Primary Key
Project_Description	VARCHAR	
Start_Date	DATE	
End_Date	DATE	
Status	VARCHAR	
Project_Name	VARCHAR	

6.Student_Hobby Table:

ATTRIBUTE	DOMAIN	CONSTRAINT
Student_id	NUMBER	Foreign Key
Hobby_id	NUMBER	Foreign Key

7.Student_Skill Table:

ATTRIBUTE	DOMAIN	CONSTRAINT
Student_id	NUMBER	Foreign Key
Skill_id	NUMBER	Foreign Key
Proficiency	VARCHAR	

8.Student_Project Table:

ATTRIBUTE	DOMAIN	CONSTRAINT
Student_id	NUMBER	Foreign Key
project_id	NUMBER	Foreign Key
Role	VARCHAR	

HOBBIES DATABASE

The relations are many to many relations —ER DIAGRAM

DDL COMMANDS :

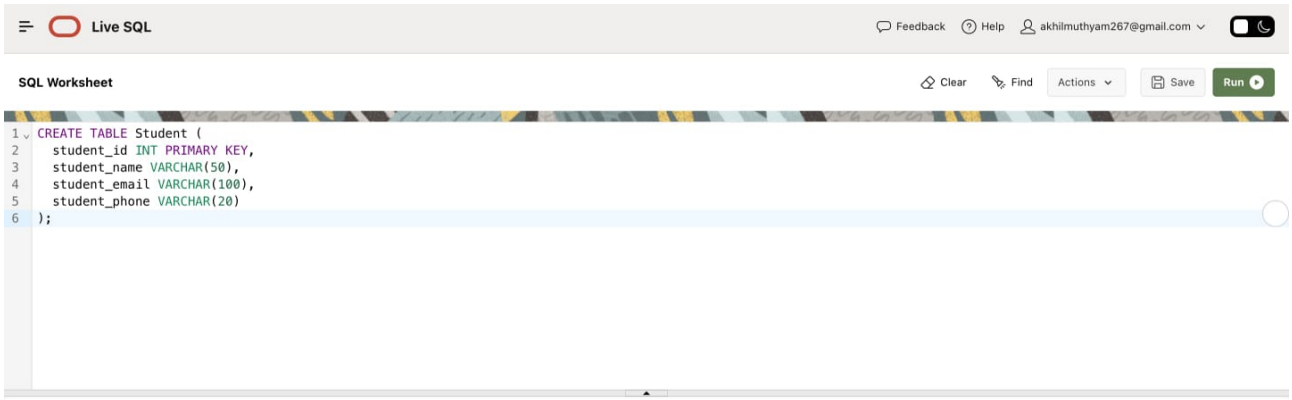
1.creating table for **student** with constraints :

QUERY : create table student(
2 student_id int PRIMARY KEY ,
3 student_name VARCHAR(50) ,
4 student_email VARCHAR(100),
5 student_phone VARCHAR(20));

1602-21-737-005

Akhil

HOBBIES DATABASE



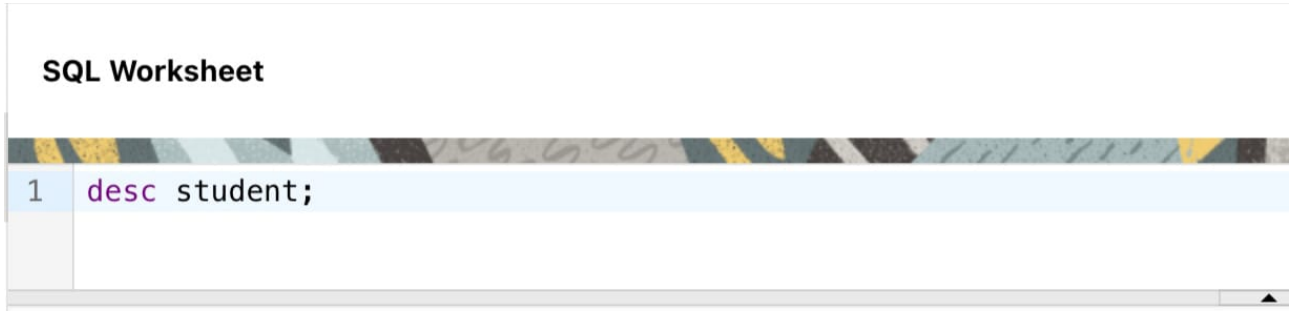


TABLE STUDENT

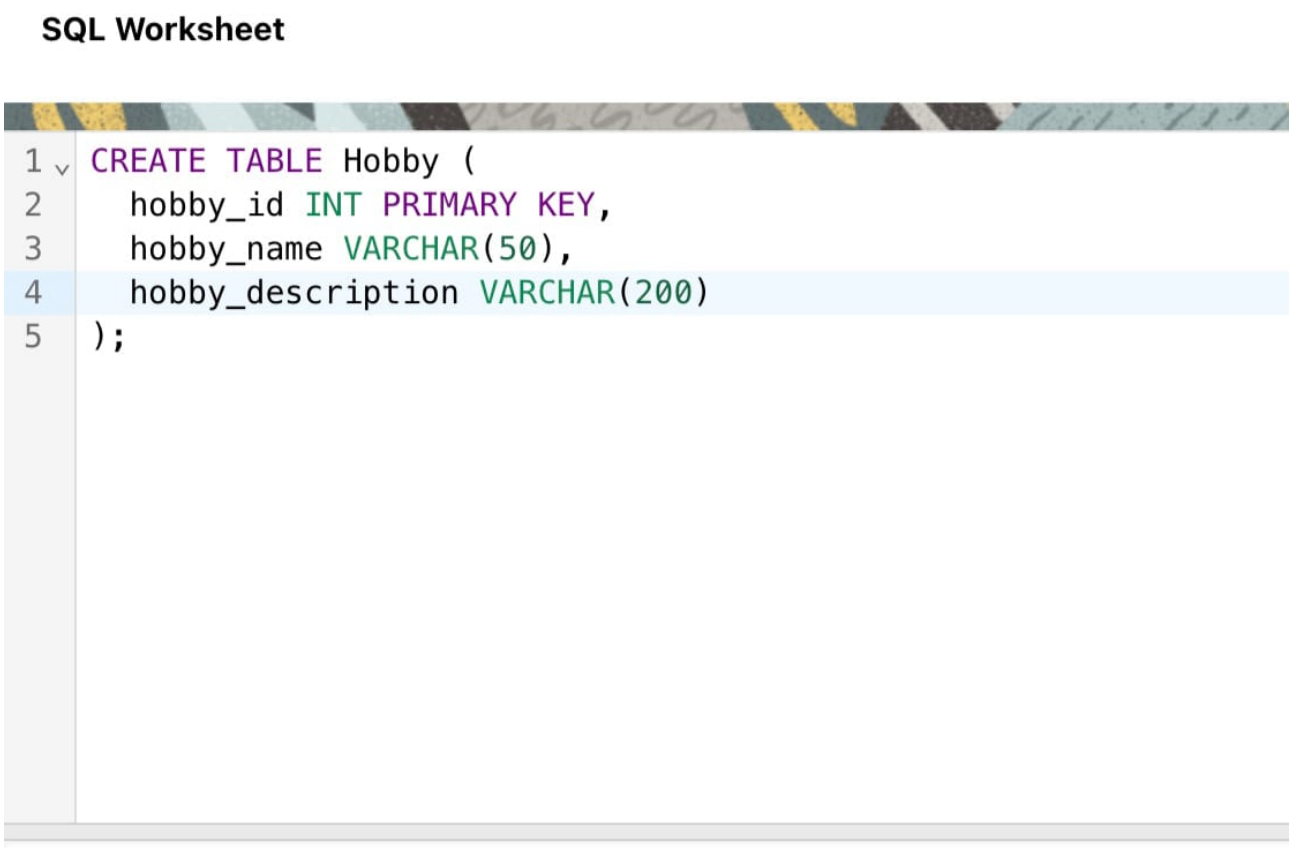
Column	Null?	Type
STUDENT_ID	NOT NULL	NUMBER
STUDENT_NAME	NOT NULL	VARCHAR2(50)
STUDENT_EMAIL	-	VARCHAR2(100)
STUDENT_PHONE	-	VARCHAR2(20)

Download CSV

4 rows selected.

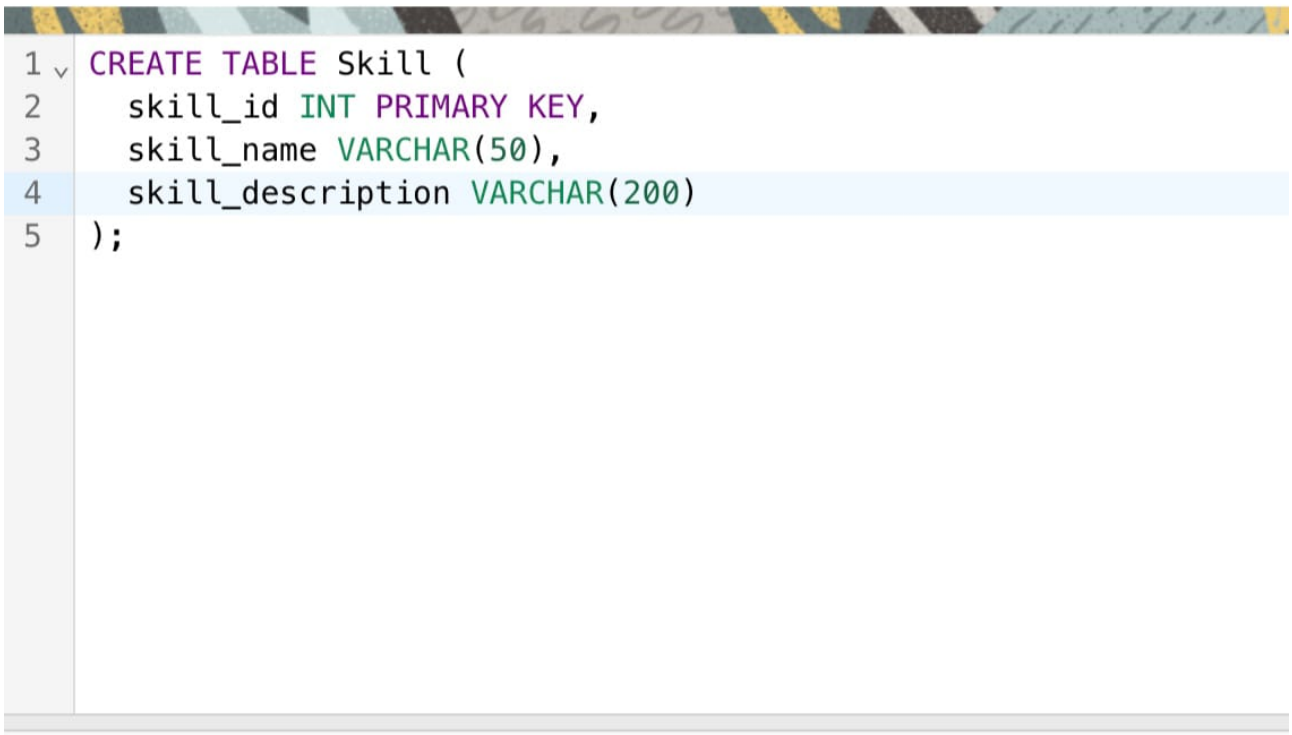
1602-21-737-005 Akhil

2.creating **hobby** table :
QUERY: create table hobby (
2 hobby_id int primary key,
3 hobby_name varchar(50),
4 hobby_description varchar(200));



2.creating **hobby** table :
QUERY: create table hobby (
2 hobby_id int primary key,
3 hobby_name varchar(50),
4 hobby_description varchar(200));

SQL Worksheet



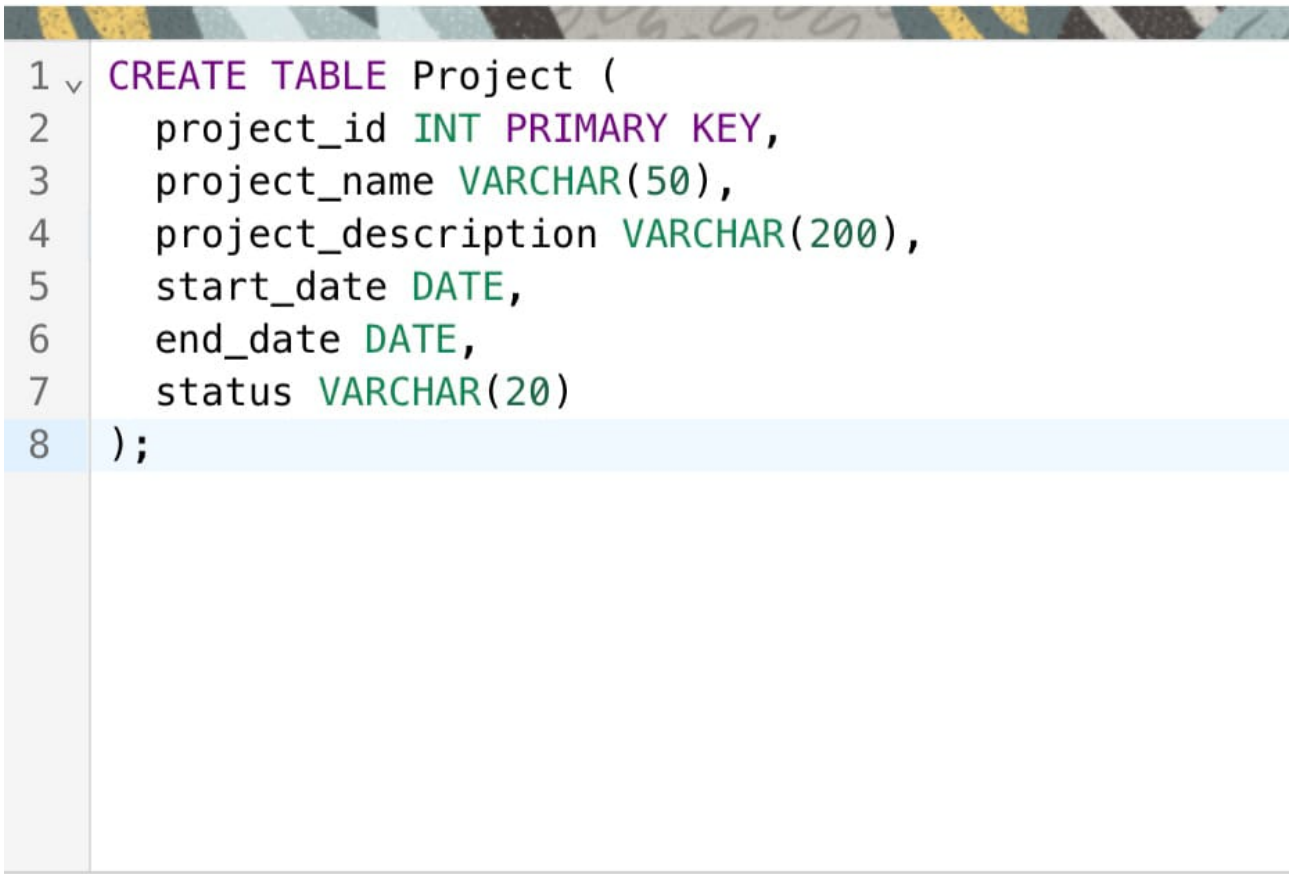
```
CREATE TABLE Skill (
  skill_id INT PRIMARY KEY,
  skill_name VARCHAR(50),
  skill_description VARCHAR(200)
);
```

```
Table created.
```

3.creating **skill** table :
QUERY: create table skill(
2 skill_id int primary key ,
3 skill_name varchar(50),
4 skill_description varchar(200));

4.creating **project** table : **QUERY:** create table project(
2 project_id int primary key ,
3 project_name varchar(50),
4 project_description varchar(20), 5 start_date DATE,
6 end_date DATE,
7 status varchar(20));

SQL Worksheet



```
CREATE TABLE Project (
  project_id INT PRIMARY KEY,
  project_name VARCHAR(50),
  project_description VARCHAR(200),
  start_date DATE,
  end_date DATE,
  status VARCHAR(20)
);
```

```
Table created.
```

5.creating **student_skill** table:
QUERY: create table student_skill (
2 student_id int,
3 skill_id int,
4 proficiency varchar(20),
5 foreign key (student_id) references student(student_id), 6 foreign key (skill_id) references skill(skill_id),
7 primary key (student_id,skill_id));

SQL Worksheet

```sql
CREATE TABLE Student_Skill (
  student_id INT,
  skill_id INT,
  proficiency VARCHAR(20),
  FOREIGN KEY (student_id) REFERENCES Student(student_id),
  FOREIGN KEY (skill_id) REFERENCES Skill(skill_id),
  PRIMARY KEY (student_id, skill_id)
);
```

```
Table created.
```

6.creating **student_hobby** table:

QUERY: create table student_hobby(
2 student_id int,
3 hobby_id int,
4 foreign key (student_id) references

student(student_id),

5 foreign key (hobby_id) references
hobby(hobby_id),

6 primary key (student_id,hobby_id));

SQL Worksheet

```
CREATE TABLE Student_Hobby (
  student_id INT,
  hobby_id INT,
  FOREIGN KEY (student_id) REFERENCES Student(student_id),
  FOREIGN KEY (hobby_id) REFERENCES Hobby(hobby_id),
  PRIMARY KEY (student_id, hobby_id)
);
```

```
Table created.
```

7.creating **student_project** table:
QUERY: create table student_project(
2 student_id int,
3 project_id int,
4 role varchar(50),
5 foreign key (student_id) references student(student_id), 6 foreign key (project_id) references project(project_id),
7 primary key (student_id,project_id));

SQL Worksheet

```
1  CREATE TABLE Student_Project (
2    student_id INT,
3    project_id INT,
4    role VARCHAR(50),
5    FOREIGN KEY (student_id) REFERENCES Student(student_id),
6    FOREIGN KEY (project_id) REFERENCES Project(project_id),
7    PRIMARY KEY (student_id, project_id)
8  );
```

```
Table created.
```

8.creating **skill_hobby** table:

QUERY: create table skill_hobby(
2 skill_id int,
3 hobby_id int,
4 foreign key (skill_id) references skill(skill_id),
5 foreign key (hobby_id) references hobby(hobby_id), 6 primary key (skill_id,project_id));

SQL Worksheet

```
1  create table skill_hobby (
2      skill_id int,
3      hobby_id int,
4      foreign key (skill_id) references skill(skill_id),
5      foreign key (hobby_id) references hobby(hobby_id),
6      primary key(skill_id,hobby_id));
```

```
Table created.
```

DML COMMANDS:

1.insert values into **student**:
QUERY: insert into student(student_id,student_name,student_email,student_phone) 2 values(20,’hemanth’,’**hemanth122@gmail.com**','8312877721');

SQL Worksheet

```
1  select * from student;
2
```

STUDENT_ID	STUDENT_NAME	STUDENT_EMAIL	STUDENT_PHONE
45	sharma	rohit45@gmail.com	9848100223
20	hemanth	hemanth1222@gmail.com	8312877721
5	akhil	akhil267@gmail.com	8712356726
333	gayle	gaylechris@gmail.com	9848123456
18	virat	virat18@gmail.com	9848167890

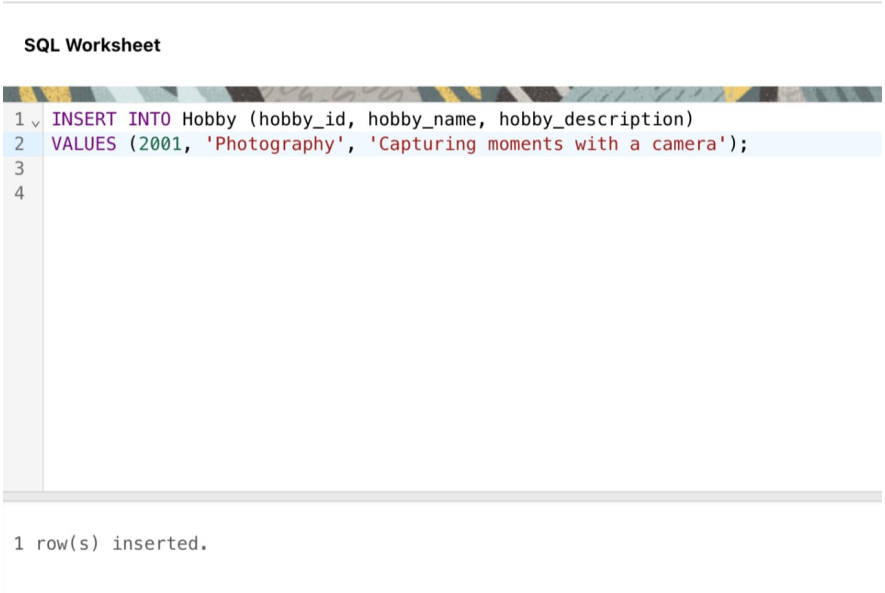
Download CSV

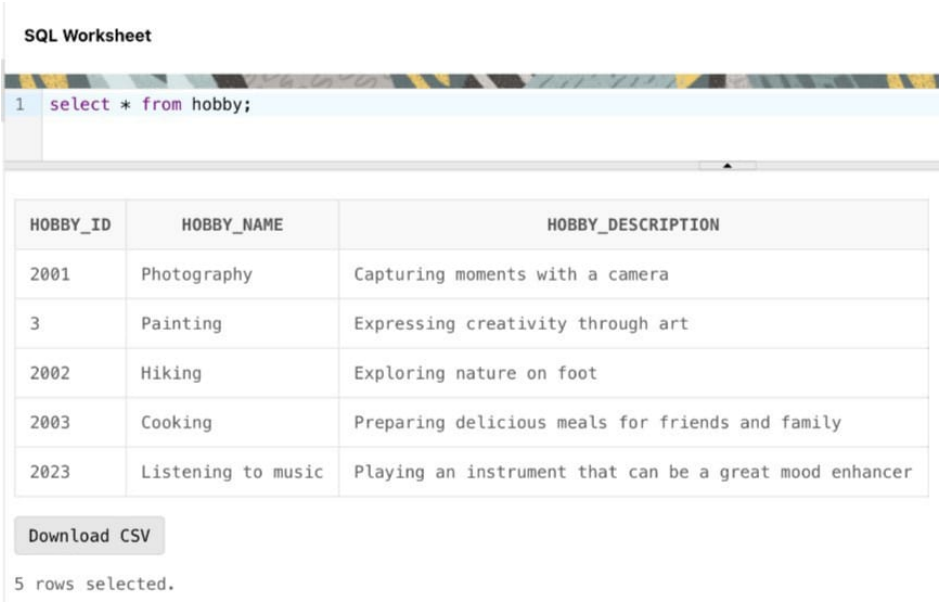
SQL Worksheet

```
1  INSERT INTO Student (student_id, student_name,student_email, student_phone)
2  VALUES (20,'hemanth','hemanth1222@gmail.com','8312877721');
3
```

1 row(s) inserted.

2.insert values into **hobby:**
QUERY:insert into hobby(hobby_id,hobby_name,hobby_description) 2 values (2001,’Photography’,’Capturing moments with a camera’);





3.insert values into **skill**:
QUERY: insert into skill(skill_id,skill_name,skill_description) 2 values (199,'Graphic Design','Creating visual content to communicate information');

SQL Worksheet

```
1  INSERT INTO Skill (skill_id, skill_name, skill_description)
2  VALUES (199, 'Graphic Design', 'Creating visual content to communicate information');
3
4
5
6
7
```

1 row(s) inserted.

SQL Worksheet Clear Find

```
1  select * from skill;
```

SKILL_ID	SKILL_NAME	SKILL_DESCRIPTION
99	Programming	The art of writing computer programs
199	Graphic Design	Creating visual content to communicate information
111	Public Speaking	Effective communication of ideas to an audience
108	Project Management	Planning, organizing, and managing resources to complete a specific project
118	Teamwork	Collaboration and cooperation within a group to achieve a common goal

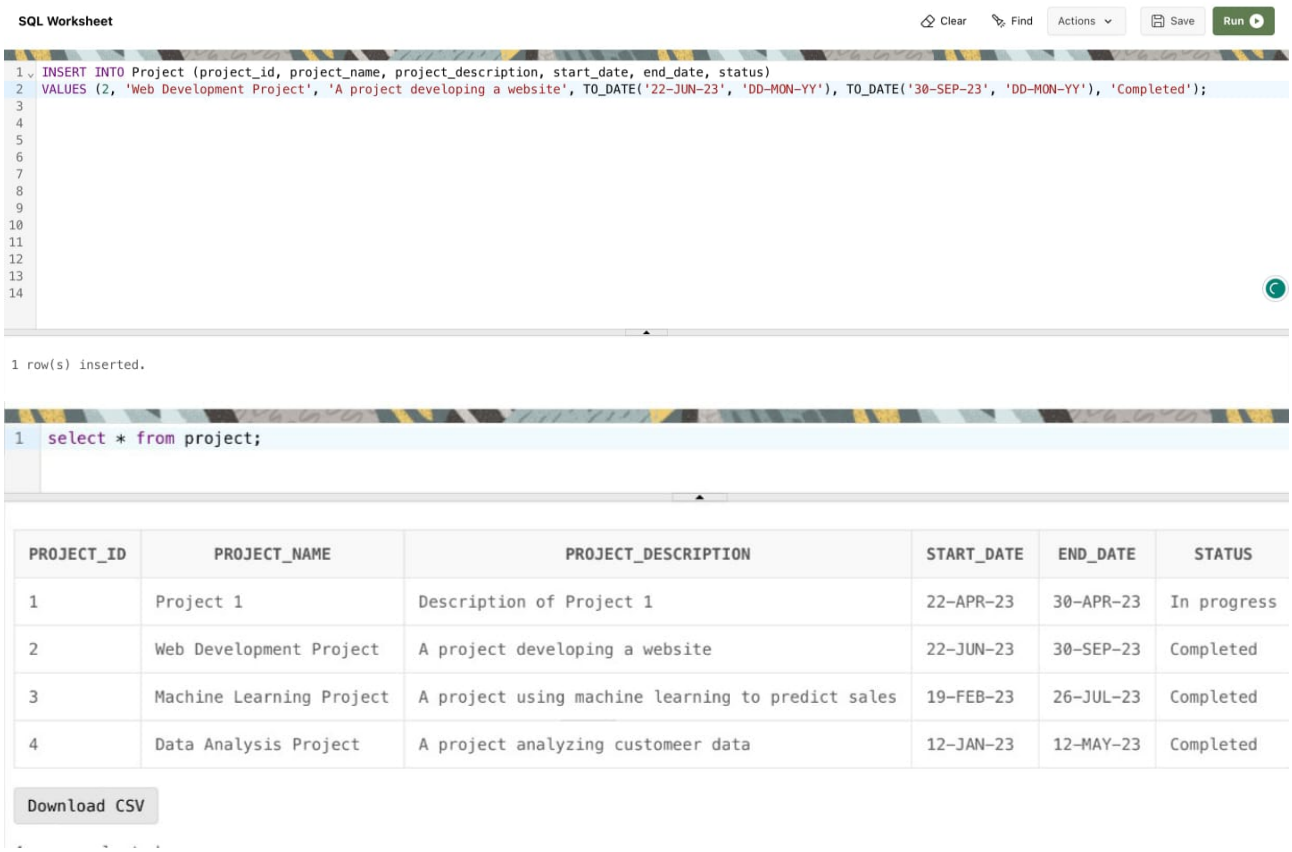
Download CSV

5 rows selected.

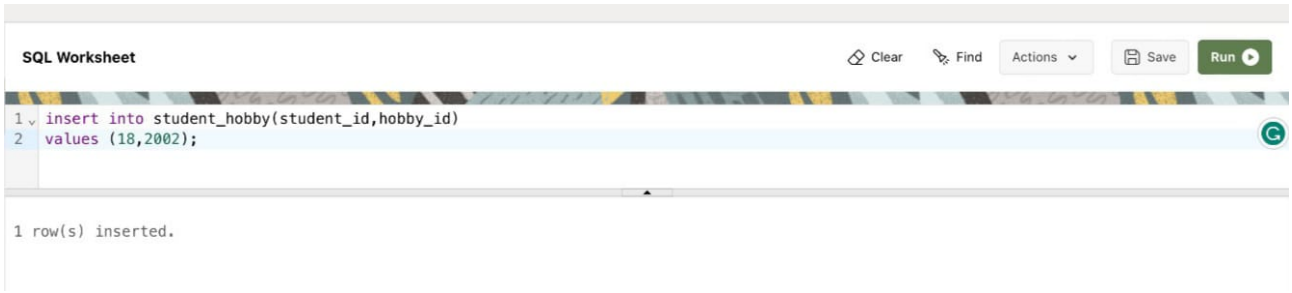
4.insert values into **project:**

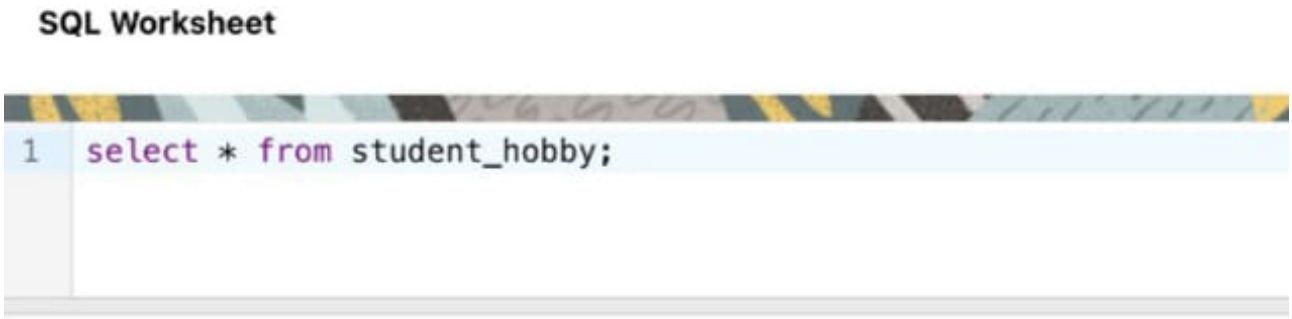
QUERY:

insert into project(project_id,project_name,project_description,start_date,end_date)
2 values (2,'Web Development Project','A project developing a website',TO_DATE('22-JUN-23','DD-MON-YY'),TO_DATE('30-SEP-23','DD-MON-YY'),'Completed');



5.insert values into **student_hobby:**
QUERY: insert into student_hobby(student_id,hobby_id) 2 values (18,2002);



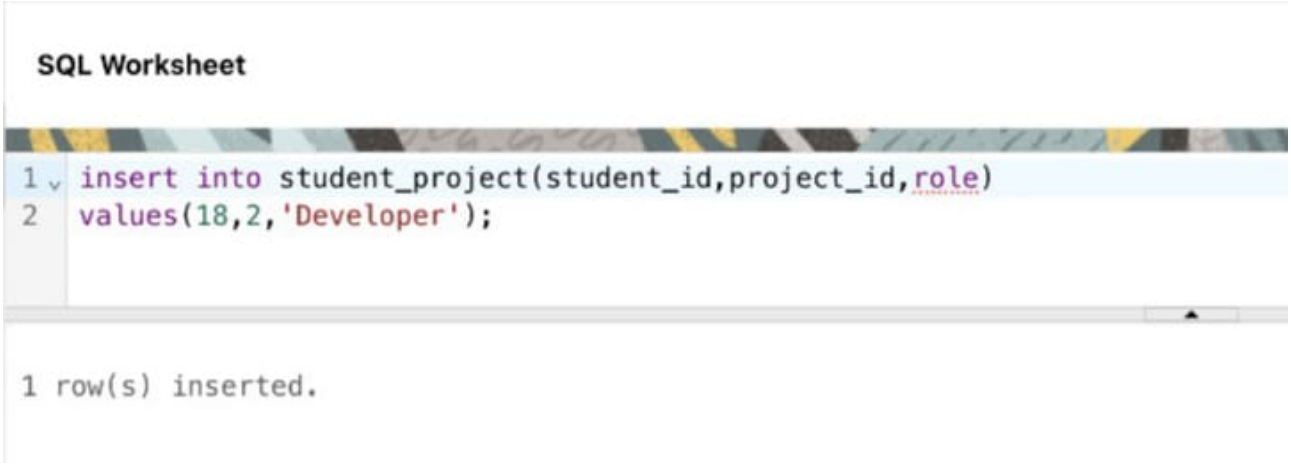


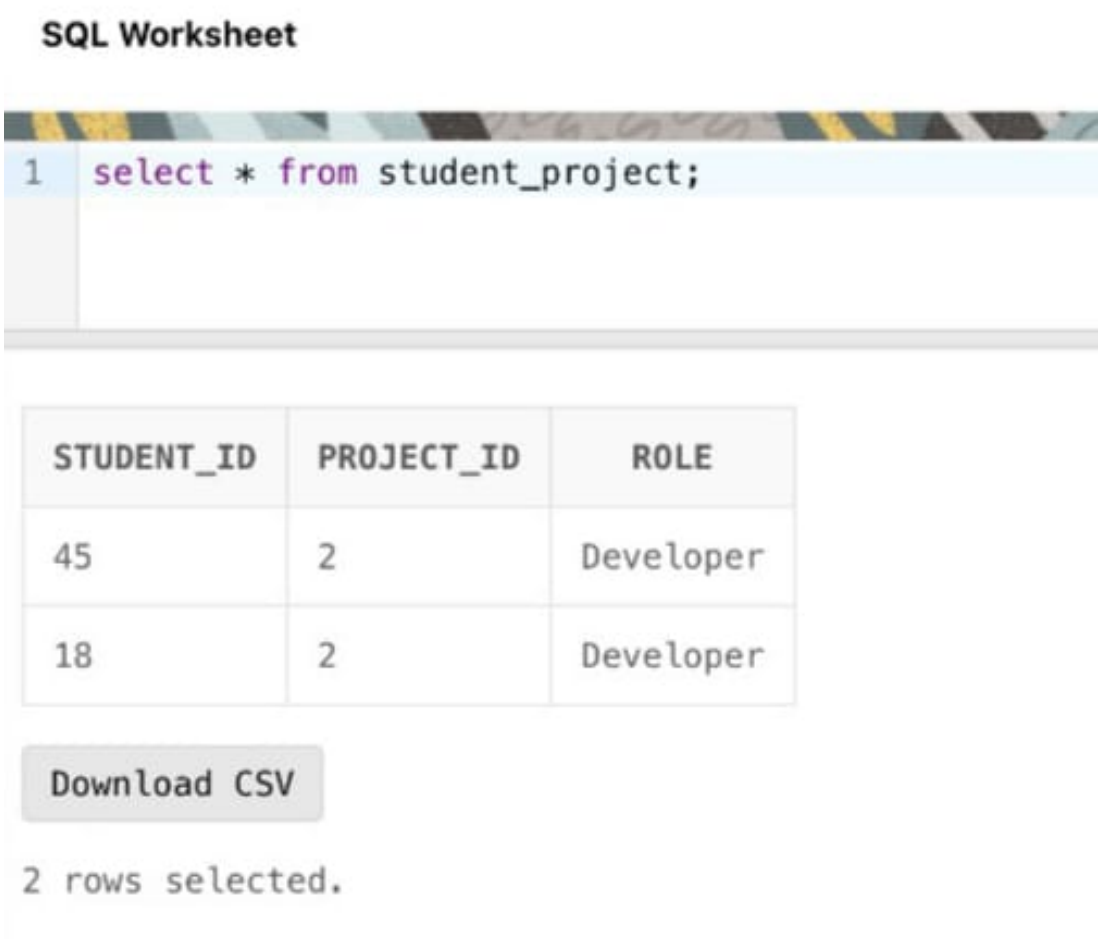
STUDENT_ID	HOBBY_ID
18	2002
20	2001
20	2002
45	2023

Download CSV

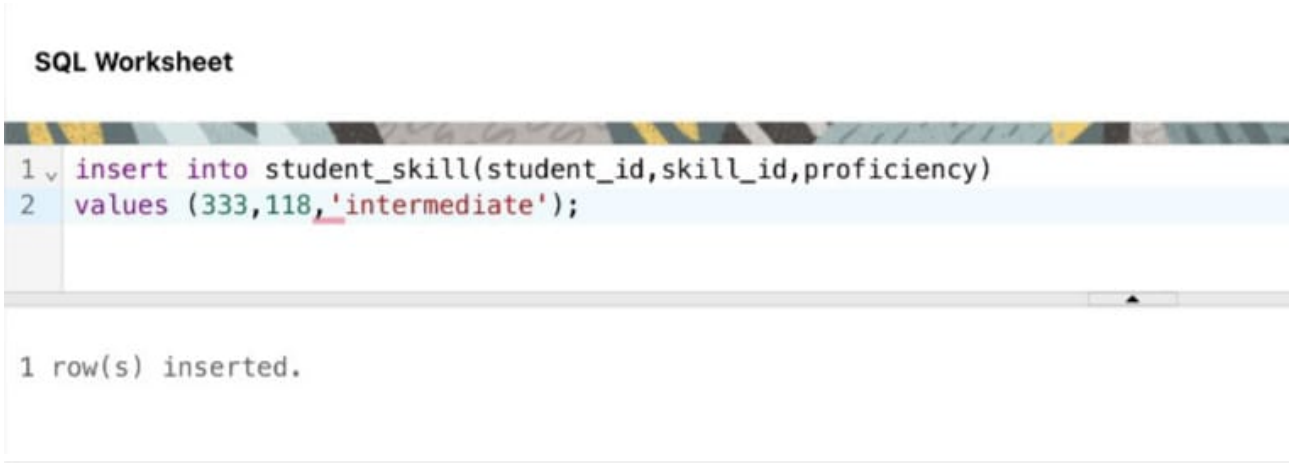
4 rows selected.

6.insert values into **student_project:**
QUERY: insert into student_project(student_id,project_id,role) 2 values (18,2,'Developer');





7.insert values into **student_skill:**
QUERY: insert into student_skill(student_id,skill_id,proficiency) 2 values (333,118,'intermediate');



STUDENT_ID	SKILL_ID	PROFICIENCY
333	118	intermediate
20	111	Beginner

Download CSV

8.insert values into **skill_hobby:**
QUERY: insert into skill_hobby(skill_id,hobby_id) 2 values (99,2003);

SQL Worksheet

```
1  select * from skill_hobby;
```

SKILL_ID	HOBBY_ID
99	2003
111	2002

Download CSV

2 rows selected.

SQL Worksheet

```
1  insert into skill_hobby(skill_id,hobby_id)
2      values (99,2003);
```

1 row(s) inserted.

IMPLEMENTATION JAVA-SQL

Connectivity using JDBC:

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

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

```
{

DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());

// Connect to Oracle Database

Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE" ,"
hr","hr");

Statement statement = con.createStatement()

String query = "UPDATE SKILL SET SKILL_NAME=“ +"'"+
jTextField3.getText() +"',SKILL_DESCRIPTION=" +"'"+ jTextField5.getText()
+"' WHERE SKILL_ID =+" + jTextField4.getText();

ResultSet rs = statement.executeQuery(query);

JOptionPane.showMessageDialog(new JFrame(), "Upadated Successfully",
"INFORMATION", JOptionPane.INFORMATION_MESSAGE);

rs.close();

statement.close(); con.close();  }
```

Front-end Programs (User Interfaces) Login Page:

1.Login Page

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class Login extends JFrame {
    private JTextField usernameField;
    private JPasswordField passwordField;
    Connection con = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    public Login() {
        setTitle("Login Page");
        setSize(400, 250);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(3, 2, 10, 10));

        JLabel usernameLabel = new JLabel("Username:");
        usernameField = new JTextField();
        JLabel passwordLabel = new JLabel("Password:");
        passwordField = new JPasswordField();
        JButton loginButton = new JButton("Login");

        loginButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String username = usernameField.getText();
                String password = new String(passwordField.getPassword());
                try
                {
                    con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
                    String name = usernameField .getText();
                    String pass = passwordField.getText();
                    String sql = "select * from logintbl where username='"+name+"' and
password='"+pass+"' ";
                    pst = con.prepareStatement(sql);
                    rs = pst.executeQuery();

                    if(rs.next())
                    {
                        JOptionPane.showMessageDialog(null, "Success");
                        con.close();
                        MainFrame t  = new MainFrame();
```

```
                t.setVisible(true);



            }
            else
            {
                usernameField.setText("");
                passwordField.setText("");
                JOptionPane.showMessageDialog(Login.this, "Invalid username or password");
            }
        }
        catch(SQLException ex)
        {
            JOptionPane.showMessageDialog(null, ex);


        }


    }
});

panel.add(usernameLabel);
panel.add(usernameField);
panel.add(passwordLabel);
panel.add(passwordField);
panel.add(loginButton);


panel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));

add(panel);
setVisible(true);
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new Login();
        }
    });
}
}
```

1602-21-737-005

Akhil

2.Home Page

```
import javax.swing.*;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import javax.swing.table.AbstractTableModel;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.util.Vector;

public class MainFrame extends JFrame {
    private JDesktopPane desktopPane;
    private JTable table;
    private JTable tableS;
    private JTable tableH;
    private JTable tableP;
    private JTextField idField;
    private JTextField nameField;
    private JTextField emailField;
    private JTextField phoneField;
    private JTextField txthobby;
    private JTextField txthobbydes;
    private JTextField txthobbyname;
    private JTextField txtskill;
    private JTextField txtskillname;
    private JTextField txtskilldes;
    private JTextField txtproj;
    private JTextField txtprojdes;
    private JTextField txtprojname;
    private JTextField txtstat;

    public MainFrame() {
        setTitle("Main Frame");
        setSize(800, 600);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        desktopPane = new JDesktopPane();
        JMenuBar menuBar = new JMenuBar();


        JMenu homeMenu = new JMenu("Home");
        JMenu studentMenu = new JMenu("Student");
        JMenu hobbyMenu = new JMenu("Hobby");
        JMenu skillMenu = new JMenu("Skill");
```

```
JMenu projectMenu = new JMenu("Project");


JMenuItem homeItem = new JMenuItem("Home");
JMenuItem studentItem = new JMenuItem("Student");
JMenuItem hobbyItem = new JMenuItem("Hobby");
JMenuItem skillItem = new JMenuItem("Skill");
JMenuItem projectItem = new JMenuItem("Project");



homeItem .addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
            desktopPane.removeAll();
        try {
                                    showInternalFrame("Home");
                                } catch (SQLException e1) {
                                    // TODO Auto-generated catch block
                                    e1.printStackTrace();
                                }
    }
});

studentItem .addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {

            desktopPane.removeAll();
            try {
                                    showInternalFrame1("Student");
                                } catch (SQLException e1) {

                                    e1.printStackTrace();
                                }
    }
});

hobbyItem .addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
            desktopPane.removeAll();
        try {
                                    showInternalFrame2("Hobby");
                                } catch (SQLException e1) {
                                    // TODO Auto-generated catch block
                                    e1.printStackTrace();
                                }
    }
});

skillItem .addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
            desktopPane.removeAll();
      try {
                                showInternalFrame3("Skill");
                            } catch (SQLException e1) {
                                // TODO Auto-generated catch block
                                e1.printStackTrace();
```

```
                }
            }
        });

        projectItem .addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                    desktopPane.removeAll();
                try {
                                        showInternalFrame4("Project");
                                    } catch (SQLException e1) {

                                        e1.printStackTrace();
                                    }
            }
        });




        menuBar.add(homeMenu);
        menuBar.add(studentMenu);
        menuBar.add(hobbyMenu);
        menuBar.add(skillMenu);
        menuBar.add(projectMenu);
        homeMenu.add(homeItem);
        studentMenu.add(studentItem);
        hobbyMenu.add(hobbyItem);
        skillMenu.add(skillItem);
        projectMenu.add(projectItem);



        setJMenuBar(menuBar);



        add(desktopPane, BorderLayout.CENTER);


        setVisible(true);
        try {
            desktopPane.removeAll();
                        showInternalFrame("Home");
                    } catch (SQLException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                    }

    }

    private void showInternalFrame(String title) throws SQLException {

        JInternalFrame internalFrame = new JInternalFrame(title, true, true, true, true);
        internalFrame.setSize(800, 600);
        internalFrame.setLayout(new GridBagLayout()); // Use GridBagLayout for flexible component
placement

        GridBagConstraints constraints = new GridBagConstraints();
```

```
        constraints.anchor = GridBagConstraints.WEST;
        constraints.insets = new Insets(5, 10, 5, 5); // Add left padding to the components

        JLabel T = new JLabel("HOBBIES DATABASE");


        constraints.gridx = 0;
        constraints.gridy = 1;
        internalFrame.add(T, constraints);



        internalFrame.setVisible(true);
        desktopPane.add(internalFrame);

        // Center the internal frame within the desktop pane
        Dimension desktopSize = desktopPane.getSize();
        Dimension frameSize = internalFrame.getSize();
        internalFrame.setLocation((desktopSize.width - frameSize.width) / 2, (desktopSize.height -
frameSize.height) / 2);



    }
```

3.STUDENT TABLE

```
Connection con1;
  PreparedStatement insert;
  private void table1() throws SQLException
  {
          DefaultTableModel defaultModel = null;
      AbstractTableModel abstractModel = null;

      TableModel model = table.getModel();
      if (model instanceof DefaultTableModel) {
          defaultModel = (DefaultTableModel) model;
      } else if (model instanceof AbstractTableModel) {
          abstractModel = (AbstractTableModel) model;
      } else {
          throw new UnsupportedOperationException("Table model not supported");
      }

      if (defaultModel != null) {
          defaultModel.setRowCount(0);
      }

          try {

          Class.forName("oracle.jdbc.driver.OracleDriver");
          con1 = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
       insert = con1.prepareStatement("select * from student");
       ResultSet rs = insert.executeQuery();
       ResultSetMetaData Rss = rs.getMetaData();

       while(rs.next()) {
          String id = rs.getString("student_id");
       String name = rs.getString("student_name");
       String email = rs.getString("student_email");
       String phone = rs.getString("student_phone");
```

```
        TableModel model1 = table.getModel();



        Object[] rowData = {id, name, email, phone};

            if (model1 instanceof DefaultTableModel) {

           ((DefaultTableModel) model1).addRow(rowData);
        } else {

            Object[][] data = new Object[model1.getRowCount() + 1][model1.getColumnCount()];
            for (int row = 0; row < model1.getRowCount(); row++) {
                for (int col = 0; col < model1.getColumnCount(); col++) {
                    data[row][col] = model1.getValueAt(row, col);
                }
            }
            data[model.getRowCount()] = rowData;


            TableModel newModel = new DefaultTableModel(data, getColumnNames(model));


            table.setModel(newModel);
        }
        }


        idField.setText("");
        nameField.setText("");
        emailField.setText("");
        phoneField.setText("");
    } catch (ClassNotFoundException e) {

            e.printStackTrace();
    }


     }
     int selectedRow1;
     String ef = "";
     private void showInternalFrame1(String title) throws SQLException {

        JInternalFrame internalFrame = new JInternalFrame(title, true, true, true, true);
        internalFrame.setSize(800, 600);
        internalFrame.setLayout(new GridBagLayout());

        GridBagConstraints constraints = new GridBagConstraints();
        constraints.anchor = GridBagConstraints.WEST;
        constraints.insets = new Insets(5, 10, 5, 5);

        JLabel idLabel = new JLabel("Student ID:");
        idField = new JTextField(20);
        JLabel nameLabel = new JLabel("Student Name:");
        nameField = new JTextField(20);
        JLabel emailLabel = new JLabel("Student Email:");
        emailField = new JTextField(20);
        JLabel phoneLabel = new JLabel("Student Phone:");
```

```
    phoneField = new JTextField(20);

    JButton addButton = new JButton("Add");
    JButton editButton = new JButton("Edit");
    JButton deleteButton = new JButton("Delete");



    editButton.addActionListener(new ActionListener() {
       @Override
       public void actionPerformed(ActionEvent e) {
        try {
              int ind = Integer.parseInt(ef);
              String name = nameField.getText();
              String id = idField.getText();
              String email = emailField.getText();
              String phone = phoneField.getText();
              Class.forName("oracle.jdbc.driver.OracleDriver");
              con1 = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
              insert = con1.prepareStatement("update student set
student_name='"+name+"',student_email='"+email+"',student_phone='"+phone+"' where
student_id='"+ind+"' ");

               insert.executeUpdate();

               JOptionPane.showMessageDialog(null, "Record Update","Message",
                                                   JOptionPane.INFORMATION_MESSAGE);
              table1();

              idField.setText("");
              nameField.setText("");
              emailField.setText("");
              phoneField.setText("");
              idField.requestFocus();

          }
          catch (ClassNotFoundException | SQLException ex) {
                JOptionPane.showMessageDialog(null, ex,"WARNING",
                                                   JOptionPane.WARNING_MESSAGE);
          }

       }
    });

    deleteButton.addActionListener(new ActionListener() {
       @Override
       public void actionPerformed(ActionEvent e) {
        try {


              int ele = Integer.parseInt(idField.getText());
              if(selectedRow1==-1)
              {
                 int dialogResult = JOptionPane.showConfirmDialog(null,"Do you what to Delete the
Record","Warning",JOptionPane.YES_NO_OPTION);
              if(dialogResult == JOptionPane.YES_NO_OPTION)
              {
                 Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
                con1 =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
                insert = con1.prepareStatement("delete from student where student_id='"+ele+"' ");
                insert.executeUpdate();

                JOptionPane.showMessageDialog(null, "Record Deleted","Message",
                                              JOptionPane.INFORMATION_MESSAGE);
                table1();

                idField.setText("");
                nameField.setText("");
                emailField.setText("");
                phoneField.setText("");
                idField.requestFocus();


            }
            }
            else{
                int ind = Integer.parseInt(table.getValueAt(selectedRow1,0).toString());
            int dialogResult = JOptionPane.showConfirmDialog(null,"Do you what to Delete the
Record","Warning",JOptionPane.YES_NO_OPTION);
            if(dialogResult == JOptionPane.YES_NO_OPTION)
            {
                Class.forName("oracle.jdbc.driver.OracleDriver");
                con1 =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
                insert = con1.prepareStatement("delete from student where student_id='"+ind+"' ");
                insert.executeUpdate();

                JOptionPane.showMessageDialog(null, "Record Deleted","HELP",
                                              JOptionPane.INFORMATION_MESSAGE);
                table1();

                idField.setText("");
                nameField.setText("");
                emailField.setText("");
                phoneField.setText("");
                idField.requestFocus();

            }
            }




        }
        catch (ClassNotFoundException | SQLException ex) {
                JOptionPane.showMessageDialog(null, ex,"WARNING",
                                              JOptionPane.WARNING_MESSAGE);
        }

    }
});


String[] columnNames = {"Student ID", "Student Name", "Student Email", "Student Phone"};
Object[][] data = {
```

```
};



table = new JTable(data, columnNames);
JScrollPane scrollPane = new JScrollPane(table);



table.getSelectionModel().addListSelectionListener(new ListSelectionListener() {
    @Override
    public void valueChanged(ListSelectionEvent e) {
        if (!e.getValueIsAdjusting()) {
            selectedRow1 = table.getSelectedRow();
            if (selectedRow1 >= 0) {
                Object selectedId = table.getValueAt(selectedRow1, 0);
                Object selectedName = table.getValueAt(selectedRow1, 1);
                Object selectedEmail = table.getValueAt(selectedRow1, 2);
                Object selectedPhone = table.getValueAt(selectedRow1, 3);

                idField.setText(String.valueOf(selectedId));
                ef=String.valueOf(selectedId);
                nameField.setText(String.valueOf(selectedName));
                emailField.setText(String.valueOf(selectedEmail));
                phoneField.setText(String.valueOf(selectedPhone));
            }
        }
    }
});

table1();
constraints.gridx = 0;
constraints.gridy = 0;
internalFrame.add(idLabel, constraints);

constraints.gridy = 1;
internalFrame.add(nameLabel, constraints);

constraints.gridy = 2;
internalFrame.add(emailLabel, constraints);

constraints.gridy = 3;
internalFrame.add(phoneLabel, constraints);

constraints.gridx = 1;
constraints.gridy = 0;
constraints.gridwidth = 2;
constraints.fill = GridBagConstraints.HORIZONTAL;
internalFrame.add(idField, constraints);

constraints.gridy = 1;
internalFrame.add(nameField, constraints);

constraints.gridy = 2;
internalFrame.add(emailField, constraints);

constraints.gridy = 3;
internalFrame.add(phoneField, constraints);
```

```
        constraints.gridx = 0;
        constraints.gridy = 4;
        constraints.gridwidth = 1;
        constraints.fill = GridBagConstraints.NONE;
        internalFrame.add(addButton, constraints);

        constraints.gridx = 1;
        internalFrame.add(editButton, constraints);

        constraints.gridx = 2;
        internalFrame.add(deleteButton, constraints);

        constraints.gridx = 3;
        constraints.gridy = 0;
        constraints.gridheight = GridBagConstraints.REMAINDER;
        constraints.fill = GridBagConstraints.BOTH;
        constraints.weightx = 1.0;
        constraints.weighty = 1.0;
        internalFrame.add(scrollPane, constraints);

        addButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String id = idField.getText();
                String name = nameField.getText();
                String email = emailField.getText();
                String phone = phoneField.getText();


                TableModel model = table.getModel();


                Object[] rowData = {id, name, email, phone};



                try {
                    Class.forName("oracle.jdbc.driver.OracleDriver");
                    con1 = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
                    insert = con1.prepareStatement("insert into
student(student_id,student_name,student_email,student_phone)
values('"+id+"','"+name+"','"+email+"','"+phone+"')");

                    insert.executeUpdate();

                    JOptionPane.showMessageDialog(null, "Record Added","HELP",
                                                  JOptionPane.INFORMATION_MESSAGE);


                    table1();
                    idField.setText("");
                    nameField.setText("");
                    emailField.setText("");
                    phoneField.setText("");
                    idField.requestFocus();

                }
                catch (ClassNotFoundException | SQLException ex) {
                        JOptionPane.showMessageDialog(null, ex,"WARNING",
                                                  JOptionPane.WARNING_MESSAGE);
```

```
            }



            idField.setText("");
            nameField.setText("");
            emailField.setText("");
            phoneField.setText("");
        }
    });




    internalFrame.setVisible(true);
    desktopPane.add(internalFrame);


    Dimension desktopSize = desktopPane.getSize();
    Dimension frameSize = internalFrame.getSize();
    internalFrame.setLocation((desktopSize.width - frameSize.width) / 2, (desktopSize.height -
frameSize.height) / 2);
  }
```

4. HOBBY TABLE

```
private void table2() throws SQLException
  {
        DefaultTableModel defaultModel = null;
    AbstractTableModel abstractModel = null;

    TableModel model = table.getModel();
    if (model instanceof DefaultTableModel) {
      defaultModel = (DefaultTableModel) model;
    } else if (model instanceof AbstractTableModel) {
      abstractModel = (AbstractTableModel) model;
    } else {
      throw new UnsupportedOperationException("Table model not supported");
    }

    if (defaultModel != null) {
      defaultModel.setRowCount(0);
    }

        try {

        Class.forName("oracle.jdbc.driver.OracleDriver");
        con1 = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
     insert = con1.prepareStatement("select * from hobby");
     ResultSet rs = insert.executeQuery();
     ResultSetMetaData Rss = rs.getMetaData();

     while(rs.next()) {
        String id = rs.getString("hobby_id");
     String name = rs.getString("hobby_name");
     String email = rs.getString("hobby_description");
```

```
        TableModel model1 = table.getModel();



        Object[] rowData = {id, name, email};

            if (model1 instanceof DefaultTableModel) {

           ((DefaultTableModel) model1).addRow(rowData);
        } else {

            Object[][] data = new Object[model1.getRowCount() + 1][model1.getColumnCount()];
            for (int row = 0; row < model1.getRowCount(); row++) {
                for (int col = 0; col < model1.getColumnCount(); col++) {
                    data[row][col] = model1.getValueAt(row, col);
                }
            }
            data[model.getRowCount()] = rowData;


            TableModel newModel = new DefaultTableModel(data, getColumnNames(model));


            table.setModel(newModel);
        }
        }


        txthobby.setText("");
        txthobbyname.setText("");
        txthobbydes.setText("");

} catch (ClassNotFoundException e) {

            JOptionPane.showMessageDialog(null, e,"WARNING",
                            JOptionPane.WARNING_MESSAGE);
}


  }
  int selectedRow2;
  String ef1 = "";
  private void showInternalFrame2(String title) throws SQLException {
      JInternalFrame internalFrame = new JInternalFrame(title, true, true, true, true);
      internalFrame.setSize(800, 600);
      internalFrame.setLayout(new GridBagLayout());

      GridBagConstraints constraints = new GridBagConstraints();
      constraints.anchor = GridBagConstraints.WEST;
      constraints.insets = new Insets(5, 10, 5, 5);

      JLabel idLabel = new JLabel("Hobby ID:");
      txthobby = new JTextField(20);
      JLabel nameLabel = new JLabel("Hobby Name:");
      txthobbyname = new JTextField(20);
      JLabel emailLabel = new JLabel("Hobby Description:");
      txthobbydes = new JTextField(20);
```

```
    JButton addButton = new JButton("Add");
    JButton editButton = new JButton("Edit");
    JButton deleteButton = new JButton("Delete");


    addButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
         String hob = txthobby.getText();
           String name = txthobbyname.getText();
           String des = txthobbydes.getText();

           TableModel model = table.getModel();


           Object[] rowData = {hob,name,des};
           try {
               Class.forName("oracle.jdbc.driver.OracleDriver");
               con1 = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");

               insert = con1.prepareStatement("insert into
hobby(hobby_id,hobby_name,hobby_description) values('"+hob+"','"+name+"','"+des+"' )");

               insert.executeUpdate();

               JOptionPane.showMessageDialog(null, "Record Added","HELP",
                                                   JOptionPane.INFORMATION_MESSAGE);
               table2();

               txthobby.setText("");
               txthobbyname.setText("");
               txthobbydes.setText("");

               txthobby.requestFocus();

           }
           catch (ClassNotFoundException | SQLException ex) {
                   JOptionPane.showMessageDialog(null, ex,"WARNING",
                                                   JOptionPane.WARNING_MESSAGE);
           }
       }
    });

    editButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
         try {
         int ind = Integer.parseInt(ef1);
           String name = txthobbyname.getText();
           String hob = txthobby.getText();
           String des = txthobbydes.getText();

           Class.forName("oracle.jdbc.driver.OracleDriver");
           con1 = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
           insert = con1.prepareStatement("update hobby set
hobby_name='"+name+"',hobby_description='"+des+"' where hobby_id='"+ind+"' ");

           insert.executeUpdate();
```

```
            JOptionPane.showMessageDialog(null, "Record Update","Message",
                                    JOptionPane.INFORMATION_MESSAGE);
            table2();

            txthobby.setText("");
            txthobbyname.setText("");
            txthobbydes.setText("");

            txthobby.requestFocus();

        }
        catch (ClassNotFoundException | SQLException ex) {
          JOptionPane.showMessageDialog(null, ex,"WARNING",
                                    JOptionPane.WARNING_MESSAGE);
        }

    }
    });

    deleteButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
          try {


                int ele = Integer.parseInt(txthobby.getText());
                if(selectedRow2==-1)
                {
                    int dialogResult = JOptionPane.showConfirmDialog(null,"Do you what to Delete the
Record","Warning",JOptionPane.YES_NO_OPTION);
                if(dialogResult == JOptionPane.YES_NO_OPTION)
                {
                    Class.forName("oracle.jdbc.driver.OracleDriver");
                    con1 =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
                    insert = con1.prepareStatement("delete from hobby where hobby_id='"+ele+"' ");
                    insert.executeUpdate();

                    JOptionPane.showMessageDialog(null, "Record Deleted","Message",
                                                  JOptionPane.INFORMATION_MESSAGE);
                    table2();


                    txthobby.setText("");
                    txthobbydes.setText("");
                    txthobbyname.setText("");
                    txthobby.requestFocus();


                }
                }
                else{
                    int ind = Integer.parseInt(table.getValueAt(selectedRow2,0).toString());
                int dialogResult = JOptionPane.showConfirmDialog(null,"Do you what to Delete the
Record","Warning",JOptionPane.YES_NO_OPTION);
                if(dialogResult == JOptionPane.YES_NO_OPTION)
                {
                    Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
                con1 =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
                insert = con1.prepareStatement("delete from hobby where hobby_id='"+ind+"' ");
                insert.executeUpdate();

                JOptionPane.showMessageDialog(null, "Record Deleted","Message",
                                              JOptionPane.INFORMATION_MESSAGE);
                table2();

                txthobby.setText("");
                txthobbyname.setText("");
                txthobbydes.setText("");

                txthobby.requestFocus();

            }
            }




        }
        catch (ClassNotFoundException | SQLException ex) {
                JOptionPane.showMessageDialog(null, ex,"WARNING",
                                              JOptionPane.WARNING_MESSAGE);
        }

    }
});


String[] columnNames = {"Hobby ID", "Hobby Name", "Hobby description"};
Object[][] data = {

};




table = new JTable(data, columnNames);
JScrollPane scrollPane = new JScrollPane(table);



table.getSelectionModel().addListSelectionListener(new ListSelectionListener() {
    @Override
    public void valueChanged(ListSelectionEvent e) {
        if (!e.getValueIsAdjusting()) {
            selectedRow2 = table.getSelectedRow();
            if (selectedRow2 >= 0) {
                Object selectedId = table.getValueAt(selectedRow2, 0);
                Object selectedName = table.getValueAt(selectedRow2, 1);
                Object selectedEmail = table.getValueAt(selectedRow2, 2);


                txthobby.setText(String.valueOf(selectedId));
                ef1= String.valueOf(selectedId);
                txthobbyname.setText(String.valueOf(selectedName));
```

```
                txthobbydes.setText(String.valueOf(selectedEmail));

            }
        }
    }
});

table2();
constraints.gridx = 0;
constraints.gridy = 0;
internalFrame.add(idLabel, constraints);

constraints.gridy = 1;
internalFrame.add(nameLabel, constraints);

constraints.gridy = 2;
internalFrame.add(emailLabel, constraints);



constraints.gridx = 1;
constraints.gridy = 0;
constraints.gridwidth = 2;
constraints.fill = GridBagConstraints.HORIZONTAL;
internalFrame.add(txthobby, constraints);

constraints.gridy = 1;
internalFrame.add(txthobbyname, constraints);

constraints.gridy = 2;
internalFrame.add(txthobbydes, constraints);



constraints.gridx = 0;
constraints.gridy = 4;
constraints.gridwidth = 1;
constraints.fill = GridBagConstraints.NONE;
internalFrame.add(addButton, constraints);

constraints.gridx = 1;
internalFrame.add(editButton, constraints);

constraints.gridx = 2;
internalFrame.add(deleteButton, constraints);

constraints.gridx = 3;
constraints.gridy = 0;
constraints.gridheight = GridBagConstraints.REMAINDER;
constraints.fill = GridBagConstraints.BOTH;
constraints.weightx = 1.0;
constraints.weighty = 1.0;
internalFrame.add(scrollPane, constraints);




internalFrame.setVisible(true);
desktopPane.add(internalFrame);
```

```
    Dimension desktopSize = desktopPane.getSize();
    Dimension frameSize = internalFrame.getSize();
    internalFrame.setLocation((desktopSize.width - frameSize.width) / 2, (desktopSize.height -
frameSize.height) / 2);
  }
```

5.SKILL TABLE

```
private void table3() throws SQLException
  {
        DefaultTableModel defaultModel = null;
    AbstractTableModel abstractModel = null;

    TableModel model = table.getModel();
    if (model instanceof DefaultTableModel) {
       defaultModel = (DefaultTableModel) model;
    } else if (model instanceof AbstractTableModel) {
       abstractModel = (AbstractTableModel) model;
    } else {
       throw new UnsupportedOperationException("Table model not supported");
    }

    if (defaultModel != null) {
       defaultModel.setRowCount(0);
    }

        try {

        Class.forName("oracle.jdbc.driver.OracleDriver");
        con1 = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
     insert = con1.prepareStatement("select * from skill");
     ResultSet rs = insert.executeQuery();
     ResultSetMetaData Rss = rs.getMetaData();

     while(rs.next()) {
        String id = rs.getString("skill_id");
     String name = rs.getString("skill_name");
     String email = rs.getString("skill_description");



     TableModel model1 = table.getModel();



     Object[] rowData = {id, name, email};

        if (model1 instanceof DefaultTableModel) {

       ((DefaultTableModel) model1).addRow(rowData);
    } else {

       Object[][] data = new Object[model1.getRowCount() + 1][model1.getColumnCount()];
       for (int row = 0; row < model1.getRowCount(); row++) {
          for (int col = 0; col < model1.getColumnCount(); col++) {
             data[row][col] = model1.getValueAt(row, col);
          }
       }
```

```
            data[model.getRowCount()] = rowData;


            TableModel newModel = new DefaultTableModel(data, getColumnNames(model));


            table.setModel(newModel);
        }
        }


        txtskill.setText("");
        txtskillname.setText("");
        txtskilldes.setText("");

} catch (ClassNotFoundException e) {

            JOptionPane.showMessageDialog(null, e,"WARNING",
                            JOptionPane.WARNING_MESSAGE);
}


  }
  int selectedRow3;
  String ef2 = "";
  private void showInternalFrame3(String title) throws SQLException {
      JInternalFrame internalFrame = new JInternalFrame(title, true, true, true, true);
      internalFrame.setSize(800, 600);
      internalFrame.setLayout(new GridBagLayout());

      GridBagConstraints constraints = new GridBagConstraints();
      constraints.anchor = GridBagConstraints.WEST;
      constraints.insets = new Insets(5, 10, 5, 5);

      JLabel idLabel = new JLabel("Skill ID:");
      txtskill = new JTextField(20);
      JLabel nameLabel = new JLabel("Skill Name:");
      txtskillname = new JTextField(20);
      JLabel emailLabel = new JLabel("Skill Description:");
      txtskilldes = new JTextField(20);


      JButton addButton = new JButton("Add");
      JButton editButton = new JButton("Edit");
      JButton deleteButton = new JButton("Delete");


      addButton.addActionListener(new ActionListener() {
          @Override
          public void actionPerformed(ActionEvent e) {
            String hob = txtskill.getText();
              String name = txtskillname.getText();
              String des = txtskilldes.getText();

              TableModel model = table.getModel();


              Object[] rowData = {hob,name,des};
              try {
                  Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
            con1 = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");

            insert = con1.prepareStatement("insert into skill(skill_id,skill_name,skill_description)
values('"+hob+"','"+name+"','"+des+"' )");

            insert.executeUpdate();

            JOptionPane.showMessageDialog(null, "Record Added","HELP",
                                                JOptionPane.INFORMATION_MESSAGE);
            table3();

            txtskill.setText("");
            txtskillname.setText("");
            txtskilldes.setText("");

            txtskill.requestFocus();

        }
        catch (ClassNotFoundException | SQLException ex) {
              JOptionPane.showMessageDialog(null, ex,"WARNING",
                                                JOptionPane.WARNING_MESSAGE);
        }
    }
});

    editButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
         try {
         int ind = Integer.parseInt(ef2);
           String name = txtskillname.getText();
           String hob = txtskill.getText();
           String des = txtskilldes.getText();

           Class.forName("oracle.jdbc.driver.OracleDriver");
           con1 = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
           insert = con1.prepareStatement("update skill set
skill_name='"+name+"',skill_description='"+des+"' where skill_id='"+ind+"' ");

           insert.executeUpdate();

           JOptionPane.showMessageDialog(null, "Record Update","Message",
                                JOptionPane.INFORMATION_MESSAGE);
           table3();

           txtskill.setText("");
           txtskillname.setText("");
           txtskilldes.setText("");

           txtskill.requestFocus();

        }
        catch (ClassNotFoundException | SQLException ex) {
         JOptionPane.showMessageDialog(null, ex,"WARNING",
                                JOptionPane.WARNING_MESSAGE);
        }

    }
    });
```

```
    deleteButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
          try {


                int ele = Integer.parseInt(txtskill.getText());
                if(selectedRow3==-1)
                {
                    int dialogResult = JOptionPane.showConfirmDialog(null,"Do you what to Delete the
Record","Warning",JOptionPane.YES_NO_OPTION);
                if(dialogResult == JOptionPane.YES_NO_OPTION)
                {
                    Class.forName("oracle.jdbc.driver.OracleDriver");
                    con1 =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
                    insert = con1.prepareStatement("delete from skill where skill_id='"+ele+"' ");
                    insert.executeUpdate();

                    JOptionPane.showMessageDialog(null, "Record Deleted","Message",
                                                  JOptionPane.INFORMATION_MESSAGE);
                    table3();


                    txtskill.setText("");
                    txtskillname.setText("");
                    txtskilldes.setText("");
                    txtskill.requestFocus();


                }
                }
                else{
                    int ind = Integer.parseInt(table.getValueAt(selectedRow3,0).toString());
                int dialogResult = JOptionPane.showConfirmDialog(null,"Do you what to Delete the
Record","Warning",JOptionPane.YES_NO_OPTION);
                if(dialogResult == JOptionPane.YES_NO_OPTION)
                {
                    Class.forName("oracle.jdbc.driver.OracleDriver");
                    con1 =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
                    insert = con1.prepareStatement("delete from skill where skill_id='"+ind+"' ");
                    insert.executeUpdate();

                    JOptionPane.showMessageDialog(null, "Record Deleted","Message",
                                                  JOptionPane.INFORMATION_MESSAGE);
                    table3();

                    txtskill.setText("");
                    txtskillname.setText("");
                    txtskilldes.setText("");

                    txtskill.requestFocus();

                }
                }
```

```
        }
        catch (ClassNotFoundException | SQLException ex) {
                JOptionPane.showMessageDialog(null, ex,"WARNING",
                                                JOptionPane.WARNING_MESSAGE);
        }

    }
});


String[] columnNames = {"Skill ID", "Skill Name", "Skill description"};
Object[][] data = {

};




table = new JTable(data, columnNames);
JScrollPane scrollPane = new JScrollPane(table);



table.getSelectionModel().addListSelectionListener(new ListSelectionListener() {
    @Override
    public void valueChanged(ListSelectionEvent e) {
        if (!e.getValueIsAdjusting()) {
            selectedRow3 = table.getSelectedRow();
            if (selectedRow3 >= 0) {
                Object selectedId = table.getValueAt(selectedRow3, 0);
                Object selectedName = table.getValueAt(selectedRow3, 1);
                Object selectedEmail = table.getValueAt(selectedRow3, 2);


                txtskill.setText(String.valueOf(selectedId));
                ef2= String.valueOf(selectedId);
                txtskillname.setText(String.valueOf(selectedName));
                txtskilldes.setText(String.valueOf(selectedEmail));

            }
        }
    }
});

table3();
constraints.gridx = 0;
constraints.gridy = 0;
internalFrame.add(idLabel, constraints);

constraints.gridy = 1;
internalFrame.add(nameLabel, constraints);

constraints.gridy = 2;
internalFrame.add(emailLabel, constraints);



constraints.gridx = 1;
```

```
        constraints.gridy = 0;
        constraints.gridwidth = 2;
        constraints.fill = GridBagConstraints.HORIZONTAL;
        internalFrame.add(txtskill, constraints);

        constraints.gridy = 1;
        internalFrame.add(txtskillname, constraints);

        constraints.gridy = 2;
        internalFrame.add(txtskilldes, constraints);



        constraints.gridx = 0;
        constraints.gridy = 4;
        constraints.gridwidth = 1;
        constraints.fill = GridBagConstraints.NONE;
        internalFrame.add(addButton, constraints);

        constraints.gridx = 1;
        internalFrame.add(editButton, constraints);

        constraints.gridx = 2;
        internalFrame.add(deleteButton, constraints);

        constraints.gridx = 3;
        constraints.gridy = 0;
        constraints.gridheight = GridBagConstraints.REMAINDER;
        constraints.fill = GridBagConstraints.BOTH;
        constraints.weightx = 1.0;
        constraints.weighty = 1.0;
        internalFrame.add(scrollPane, constraints);




        internalFrame.setVisible(true);
        desktopPane.add(internalFrame);


        Dimension desktopSize = desktopPane.getSize();
        Dimension frameSize = internalFrame.getSize();
        internalFrame.setLocation((desktopSize.width - frameSize.width) / 2, (desktopSize.height -
frameSize.height) / 2);
    }
```

6.PROJECT TABLE

```
private void table4() throws SQLException
    {
            DefaultTableModel defaultModel = null;
        AbstractTableModel abstractModel = null;

        TableModel model = table.getModel();
        if (model instanceof DefaultTableModel) {
            defaultModel = (DefaultTableModel) model;
        } else if (model instanceof AbstractTableModel) {
            abstractModel = (AbstractTableModel) model;
```

```
        } else {
            throw new UnsupportedOperationException("Table model not supported");
        }

        if (defaultModel != null) {
            defaultModel.setRowCount(0);
        }

            try {

                                Class.forName("oracle.jdbc.driver.OracleDriver");
                                con1 =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
                        insert = con1.prepareStatement("select * from project");
                        ResultSet rs = insert.executeQuery();
                        ResultSetMetaData Rss = rs.getMetaData();

                        while(rs.next()) {
                            String id = rs.getString("project_id");
                        String name = rs.getString("project_name");
                        String email = rs.getString("project_description");
                        String stat = rs.getString("status");



                        TableModel model1 = table.getModel();



                        Object[] rowData = {id, name, email,stat};

                            if (model1 instanceof DefaultTableModel) {

                            ((DefaultTableModel) model1).addRow(rowData);
                        } else {

                            Object[][] data = new Object[model1.getRowCount() + 1]
[model1.getColumnCount()];
                            for (int row = 0; row < model1.getRowCount(); row++) {
                                for (int col = 0; col < model1.getColumnCount(); col++) {
                                    data[row][col] = model1.getValueAt(row, col);
                                }
                            }
                            data[model.getRowCount()] = rowData;


                            TableModel newModel = new DefaultTableModel(data, getColumnNames(model));


                            table.setModel(newModel);
                        }
                        }


                        txtproj.setText("");
                        txtprojname.setText("");
                        txtprojdes.setText("");
                        txtstat.setText("");

                            } catch (ClassNotFoundException e) {
```

```
                    JOptionPane.showMessageDialog(null, e,"WARNING",
                                    JOptionPane.WARNING_MESSAGE);
                }


    }
    int selectedRow4;
    String ef3 = "";
    private void showInternalFrame4(String title) throws SQLException {
        JInternalFrame internalFrame = new JInternalFrame(title, true, true, true, true);
        internalFrame.setSize(800, 600);
        internalFrame.setLayout(new GridBagLayout());

        GridBagConstraints constraints = new GridBagConstraints();
        constraints.anchor = GridBagConstraints.WEST;
        constraints.insets = new Insets(5, 10, 5, 5);

        JLabel idLabel = new JLabel("Project ID:");
        txtproj = new JTextField(20);
        JLabel nameLabel = new JLabel("Project Name:");
        txtprojname = new JTextField(20);
        JLabel emailLabel = new JLabel("Project Description:");
        txtprojdes = new JTextField(20);
        JLabel statLabel = new JLabel("Status :");
        txtstat = new JTextField(20);



        JButton addButton = new JButton("Add");
        JButton editButton = new JButton("Edit");
        JButton deleteButton = new JButton("Delete");


        addButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                    String hob = txtproj.getText();
                String name = txtprojname.getText();
                String des = txtprojdes.getText();
                String stat = txtstat.getText();

                TableModel model = table.getModel();


                Object[] rowData = {hob,name,des};
                try {
                    Class.forName("oracle.jdbc.driver.OracleDriver");
                    con1 =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");

                    insert = con1.prepareStatement("insert into
project(project_id,project_name,project_description,status)
values('"+hob+"','"+name+"','"+des+"','"+stat+"')");

                    insert.executeUpdate();

                    JOptionPane.showMessageDialog(null, "Record Added","HELP",
                                                    JOptionPane.INFORMATION_MESSAGE);
                    table4();
```

```
                txtproj.setText("");
                txtprojname.setText("");
                txtprojdes.setText("");
                txtstat.setText("");

                txtproj.requestFocus();

            }
            catch (ClassNotFoundException | SQLException ex) {
                JOptionPane.showMessageDialog(null, ex,"WARNING",
                                                JOptionPane.WARNING_MESSAGE);
            }
        }
    });

    editButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
                try {
                int ind = Integer.parseInt(ef3);
            String name = txtprojname.getText();
            String hob = txtproj.getText();
            String des = txtprojdes.getText();
            String stat = txtstat.getText();

            Class.forName("oracle.jdbc.driver.OracleDriver");
            con1 = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
            insert = con1.prepareStatement("update project set
project_name='"+name+"',project_description='"+des+"',status='"+stat+"' where
project_id='"+ind+"' ");

            insert.executeUpdate();

            JOptionPane.showMessageDialog(null, "Record Update","Message",
                                                JOptionPane.INFORMATION_MESSAGE);
            table4();

            txtproj.setText("");
            txtprojname.setText("");
            txtprojdes.setText("");
            txtstat.setText("");

            txtproj.requestFocus();

        }
        catch (ClassNotFoundException | SQLException ex) {
                JOptionPane.showMessageDialog(null, ex,"WARNING",
                                                JOptionPane.WARNING_MESSAGE);
        }

    }
    });

    deleteButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
                try {
```

```
            int ele = Integer.parseInt(txtproj.getText());
            if(selectedRow4==-1)
            {
                int dialogResult = JOptionPane.showConfirmDialog(null,"Do you what to Delete the
Record","Warning",JOptionPane.YES_NO_OPTION);
            if(dialogResult == JOptionPane.YES_NO_OPTION)
            {
                Class.forName("oracle.jdbc.driver.OracleDriver");
                con1 =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
                insert = con1.prepareStatement("delete from project where project_id='"+ele+"' ");
                insert.executeUpdate();

                JOptionPane.showMessageDialog(null, "Record Deleted","Message",
                                              JOptionPane.INFORMATION_MESSAGE);
                table4();


                txtproj.setText("");
                txtprojname.setText("");
                txtprojdes.setText("");
                txtstat.setText("");
                txtstat.requestFocus();


            }
            }
            else{
                int ind = Integer.parseInt(table.getValueAt(selectedRow4,0).toString());
            int dialogResult = JOptionPane.showConfirmDialog(null,"Do you what to Delete the
Record","Warning",JOptionPane.YES_NO_OPTION);
            if(dialogResult == JOptionPane.YES_NO_OPTION)
            {
                Class.forName("oracle.jdbc.driver.OracleDriver");
                con1 =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
                insert = con1.prepareStatement("delete from project where project_id='"+ind+"' ");
                insert.executeUpdate();

                JOptionPane.showMessageDialog(null, "Record Deleted","Message",
                                              JOptionPane.INFORMATION_MESSAGE);
                table4();

                txtproj.setText("");
                txtprojname.setText("");
                txtprojdes.setText("");
                txtstat.setText("");

                txtproj.requestFocus();

            }
            }




        }
        catch (ClassNotFoundException | SQLException ex) {
            JOptionPane.showMessageDialog(null, ex,"WARNING",
```

```
                                                JOptionPane.WARNING_MESSAGE);
        }

    }
});


String[] columnNames = {"Project ID", "Project Name", "Project description","Status"};
Object[][] data = {

};




table = new JTable(data, columnNames);
JScrollPane scrollPane = new JScrollPane(table);



table.getSelectionModel().addListSelectionListener(new ListSelectionListener() {
    @Override
    public void valueChanged(ListSelectionEvent e) {
        if (!e.getValueIsAdjusting()) {
            selectedRow4 = table.getSelectedRow();
            if (selectedRow4 >= 0) {
                Object selectedId = table.getValueAt(selectedRow4, 0);
                Object selectedName = table.getValueAt(selectedRow4, 1);
                Object selectedEmail = table.getValueAt(selectedRow4, 2);
                Object selectedStat = table.getValueAt(selectedRow4, 3);

                txtproj.setText(String.valueOf(selectedId));
                ef3= String.valueOf(selectedId);
                txtprojname.setText(String.valueOf(selectedName));
                txtprojdes.setText(String.valueOf(selectedEmail));
                txtstat.setText(String.valueOf(selectedStat));

            }
        }
    }
});

table4();
constraints.gridx = 0;
constraints.gridy = 0;
internalFrame.add(idLabel, constraints);

constraints.gridy = 1;
internalFrame.add(nameLabel, constraints);

constraints.gridy = 2;
internalFrame.add(emailLabel, constraints);

constraints.gridy = 3;
internalFrame.add(statLabel, constraints);


constraints.gridx = 1;
constraints.gridy = 0;
```

```
        constraints.gridwidth = 2;
        constraints.fill = GridBagConstraints.HORIZONTAL;
        internalFrame.add(txtproj, constraints);

        constraints.gridy = 1;
        internalFrame.add(txtprojname, constraints);

        constraints.gridy = 2;
        internalFrame.add(txtprojdes, constraints);

        constraints.gridy = 3;
        internalFrame.add(txtstat, constraints);



        constraints.gridx = 0;
        constraints.gridy = 4;
        constraints.gridwidth = 1;
        constraints.fill = GridBagConstraints.NONE;
        internalFrame.add(addButton, constraints);

        constraints.gridx = 1;
        internalFrame.add(editButton, constraints);

        constraints.gridx = 2;
        internalFrame.add(deleteButton, constraints);

        constraints.gridx = 3;
        constraints.gridy = 0;
        constraints.gridheight = GridBagConstraints.REMAINDER;
        constraints.fill = GridBagConstraints.BOTH;
        constraints.weightx = 1.0;
        constraints.weighty = 1.0;
        internalFrame.add(scrollPane, constraints);




        internalFrame.setVisible(true);
        desktopPane.add(internalFrame);


        Dimension desktopSize = desktopPane.getSize();
        Dimension frameSize = internalFrame.getSize();
        internalFrame.setLocation((desktopSize.width - frameSize.width) / 2, (desktopSize.height -
frameSize.height) / 2);
    }
    private String[] getColumnNames(TableModel model) {
        int columnCount = model.getColumnCount();
        String[] columnNames = new String[columnCount];
        for (int col = 0; col < columnCount; col++) {
            columnNames[col] = model.getColumnName(col);
        }
        return columnNames;
    }
```

MAIN FUNCTION FOR RUNNING

```
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new MainFrame();
            }
        });
    }
}
```

1602-21-737-005
AKHIL

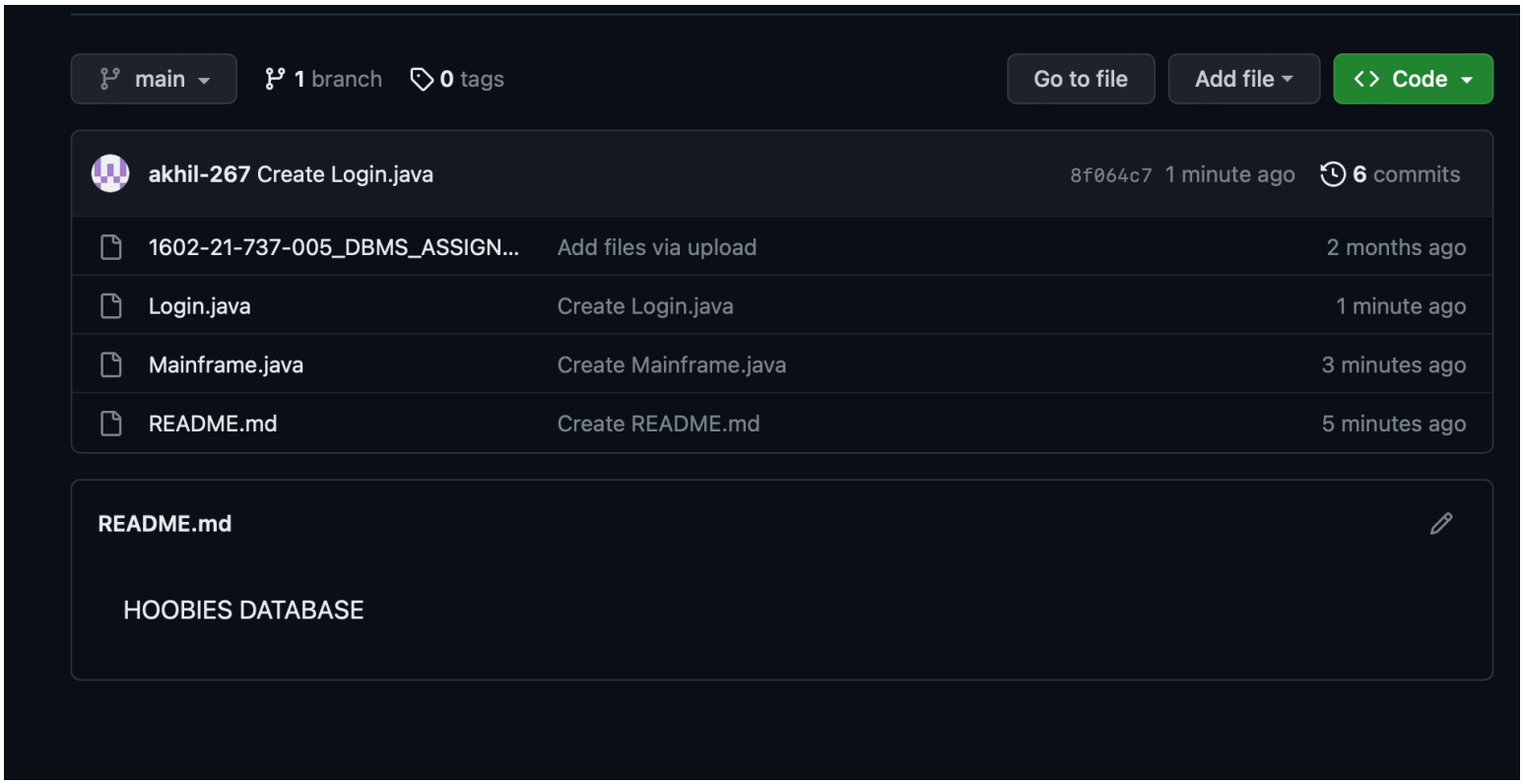
Hobbies Database

GitHub Links and Folder Structure

Link:

https://github.com/akhil-267/DBMS_PROJECT

Folder Structure:



1602-21-737-005

Akhil

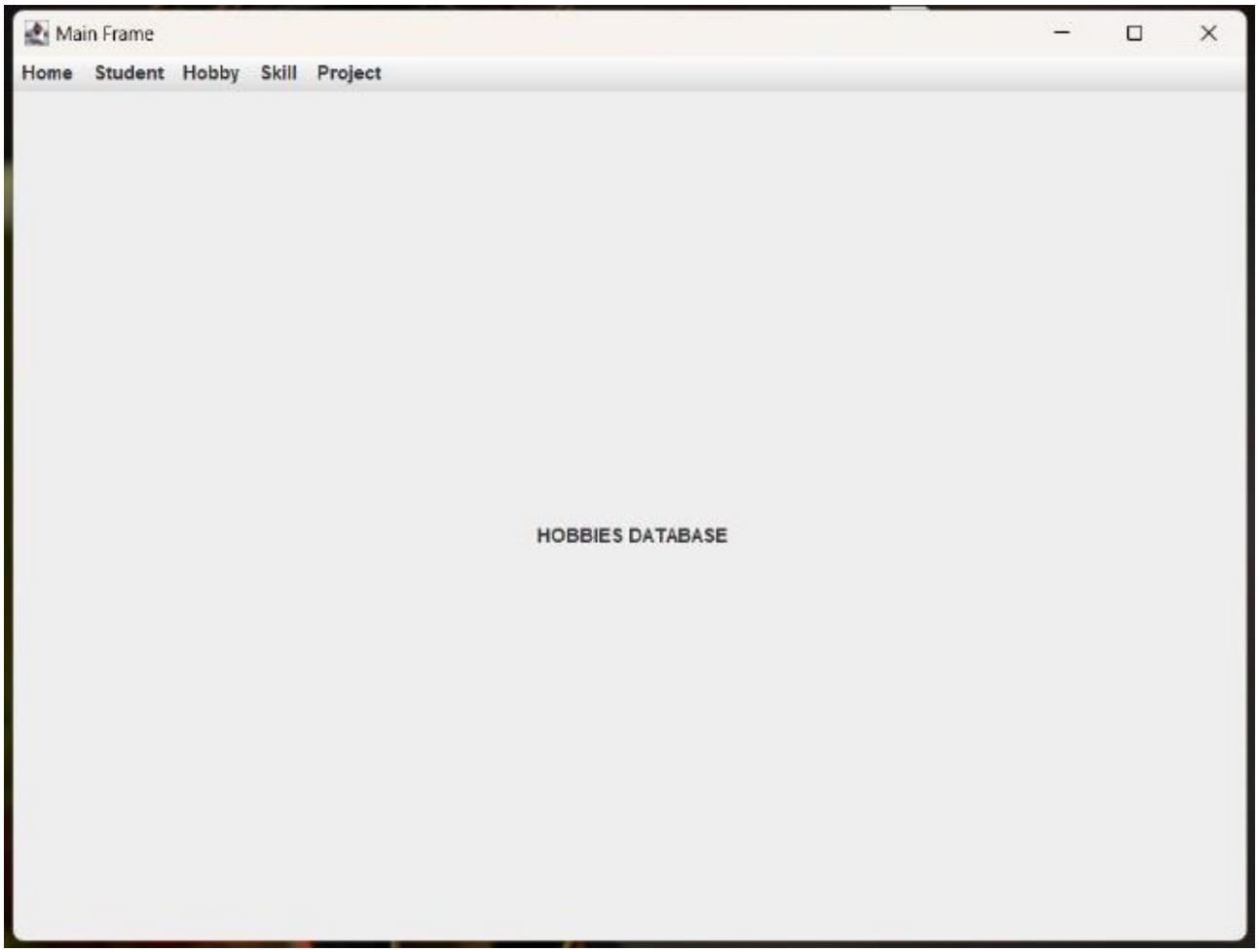
TESTING

HOBBIES DATABASE LOGIN PAGE:

1602-20-737-005

Akhil

HOME PAGE

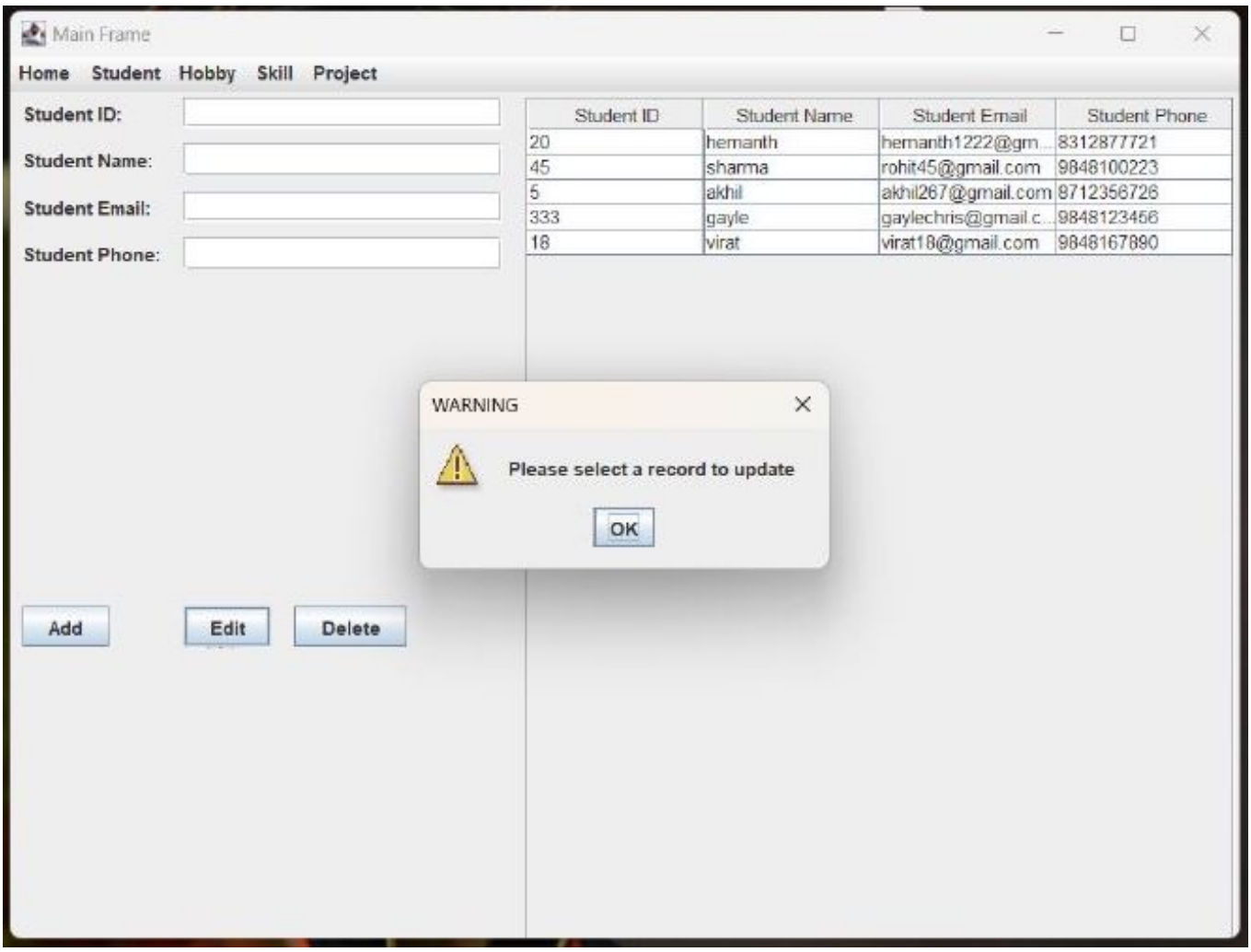


1602-21-737-005

Akhil

STUDENT TABLE

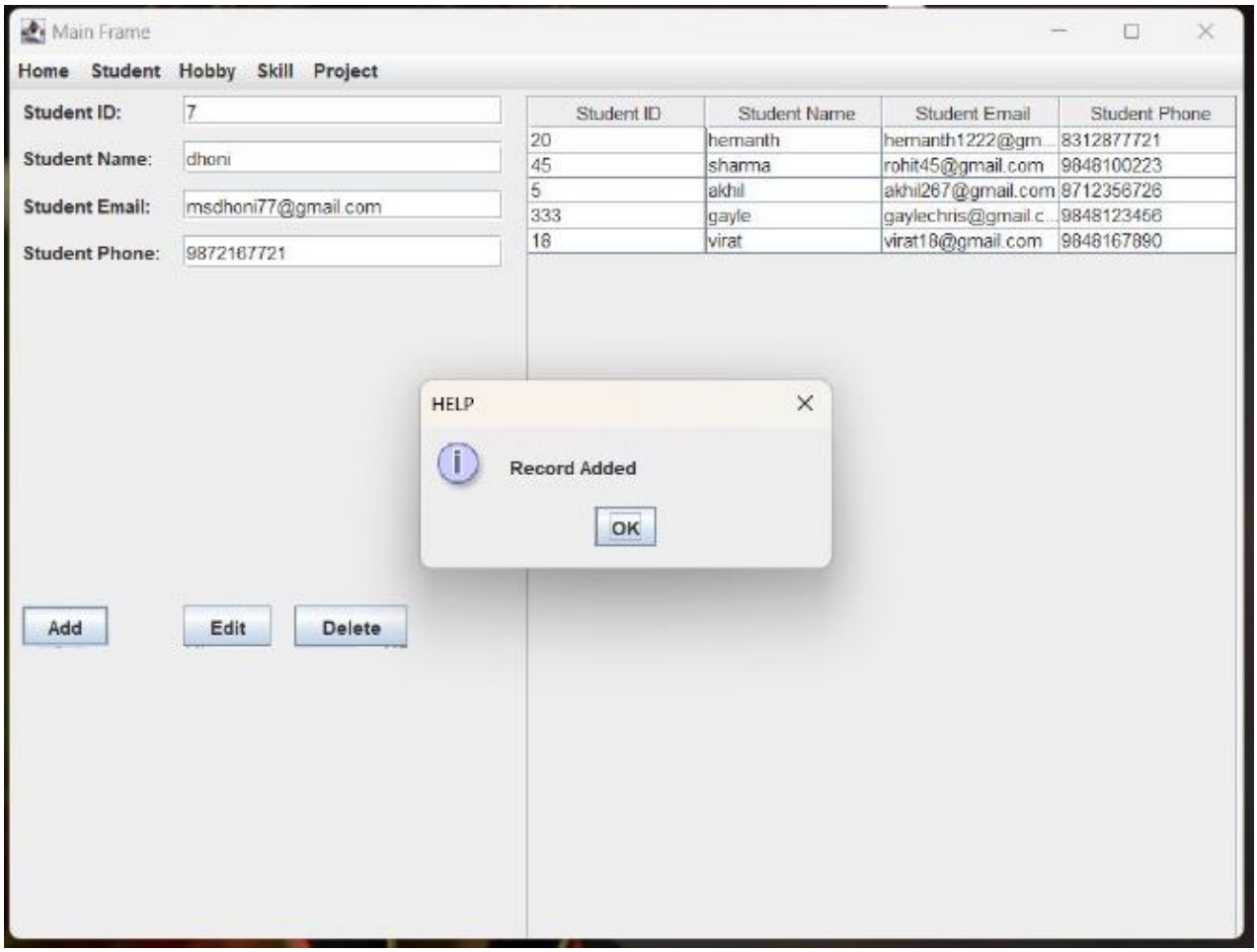
I.UPDATE WITHOUT INFORMATION



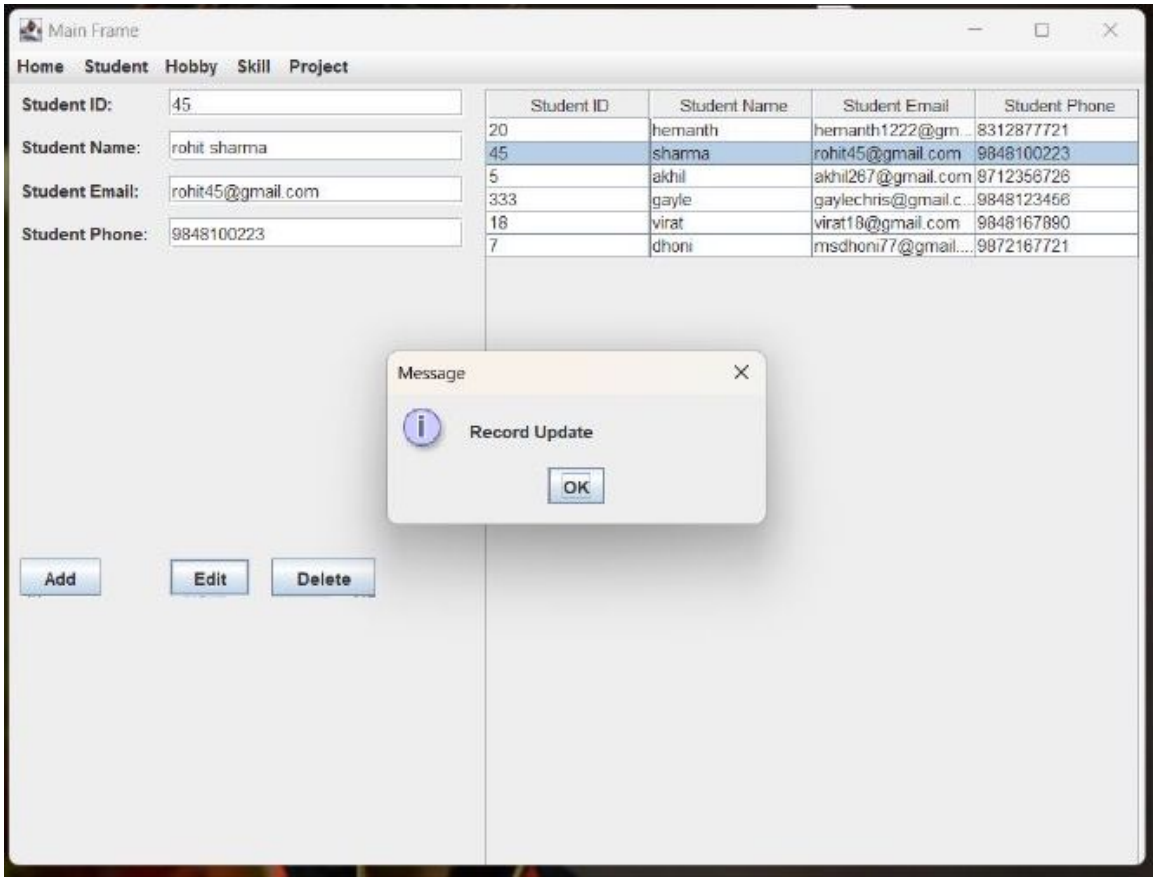
1602-21-737-005

Akhil

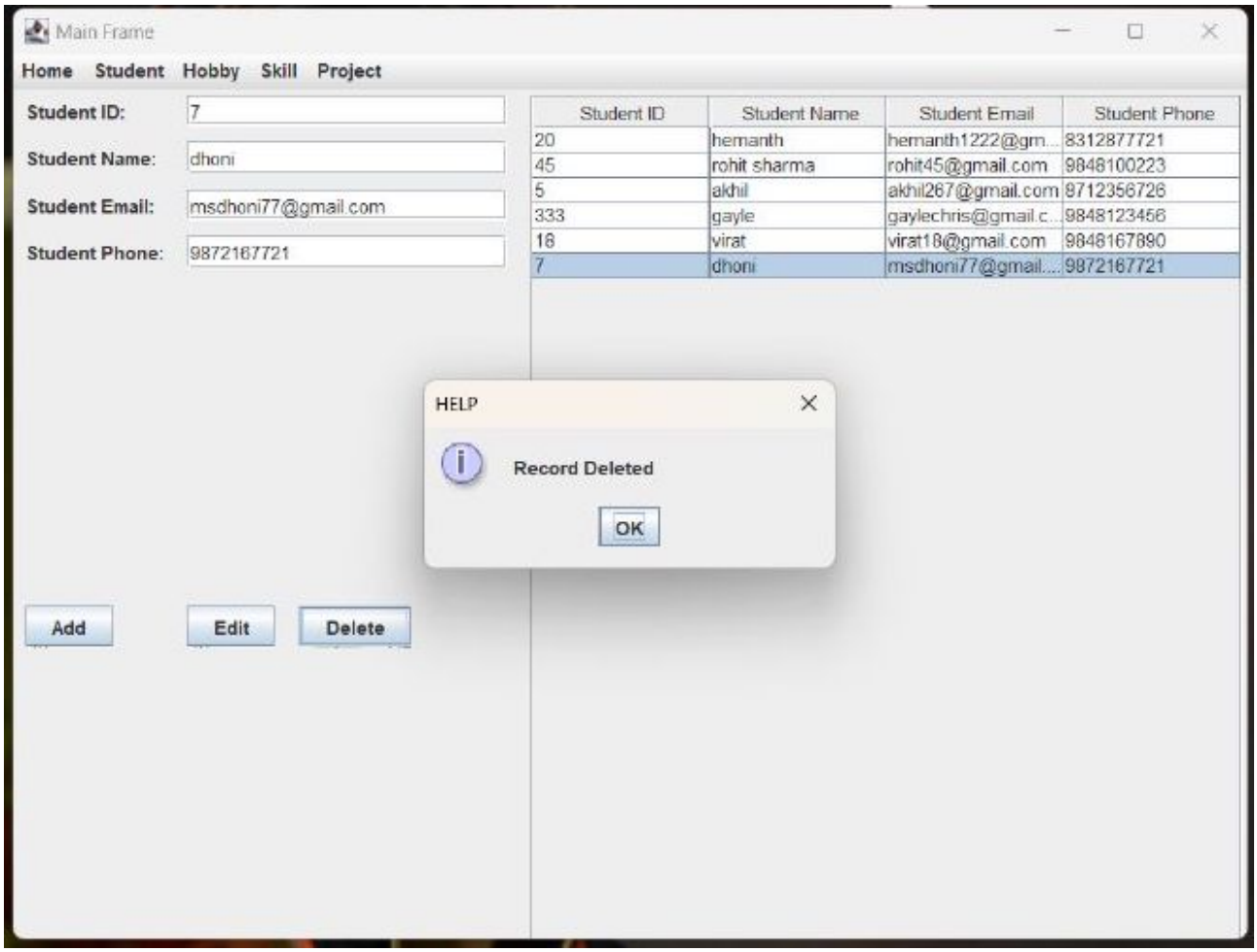
II.INSERT STUDENT TABLE INFORMATION



III. UPDATE STUDENT TABLE INFORMATION



IV.DELETE STUDENT TABLE INFORMATION

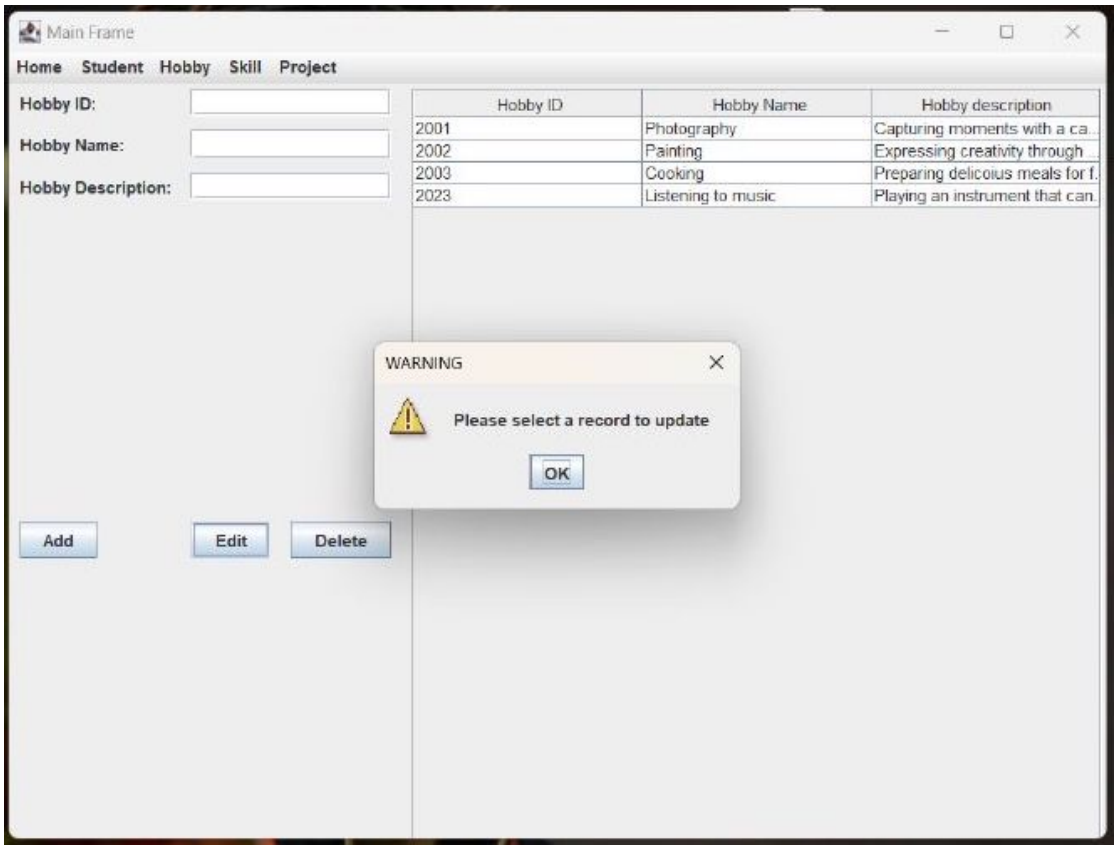


1602-21-737-005

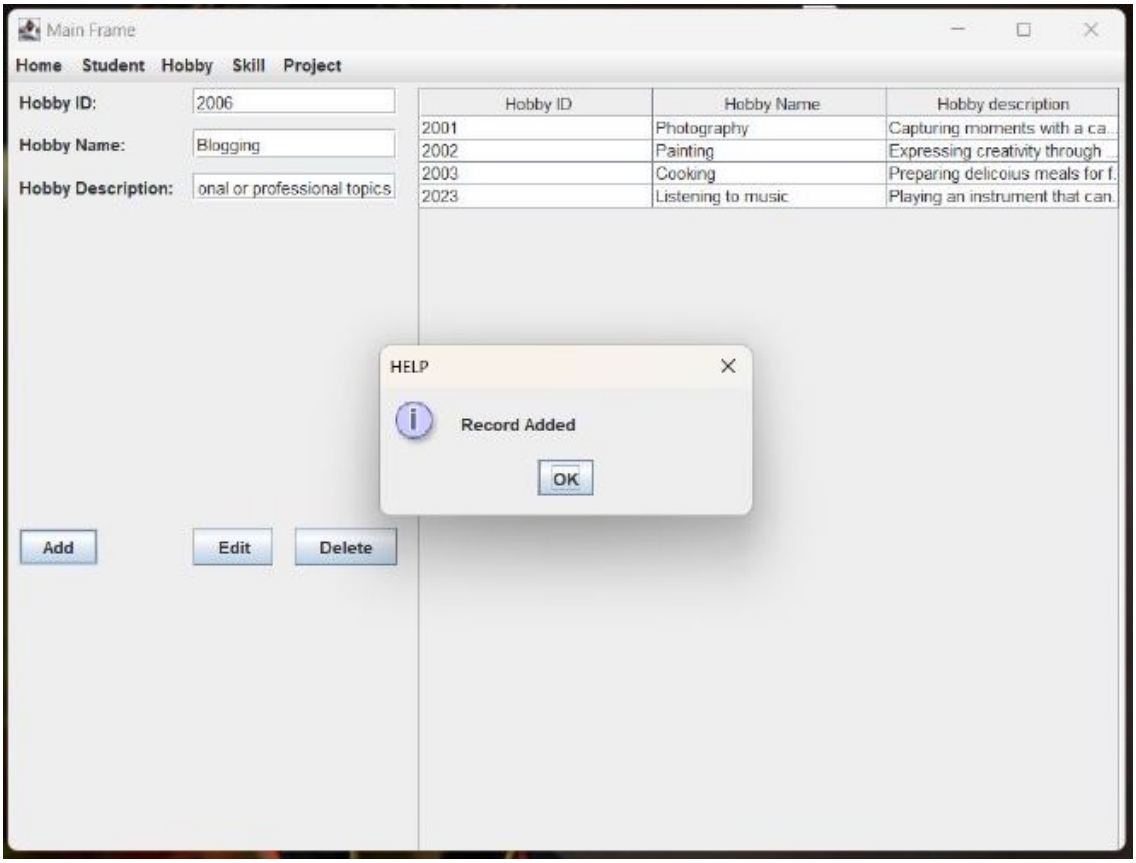
Akhil

HOBBY TABLE

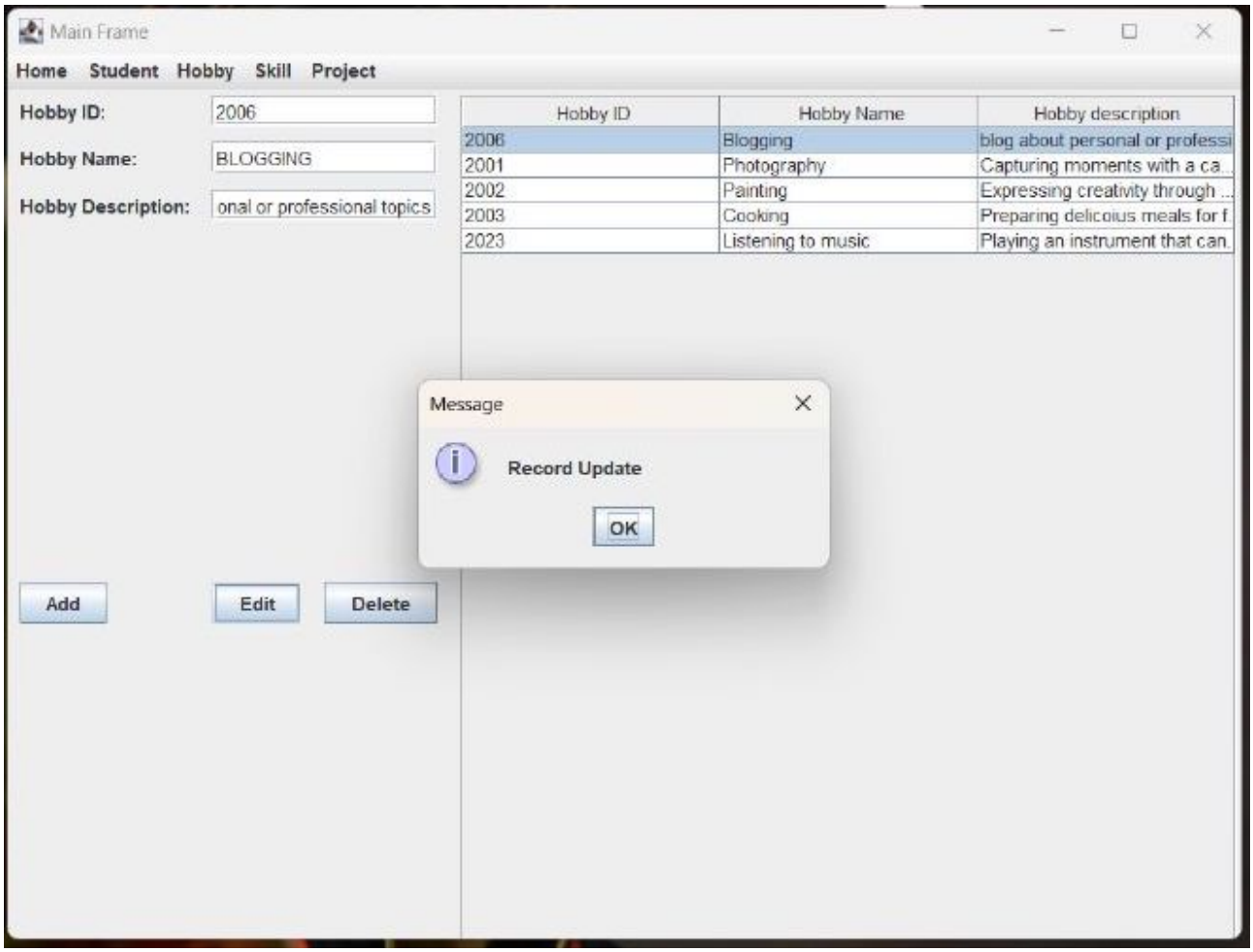
I.UPDATE WITHOUT INFORMATION



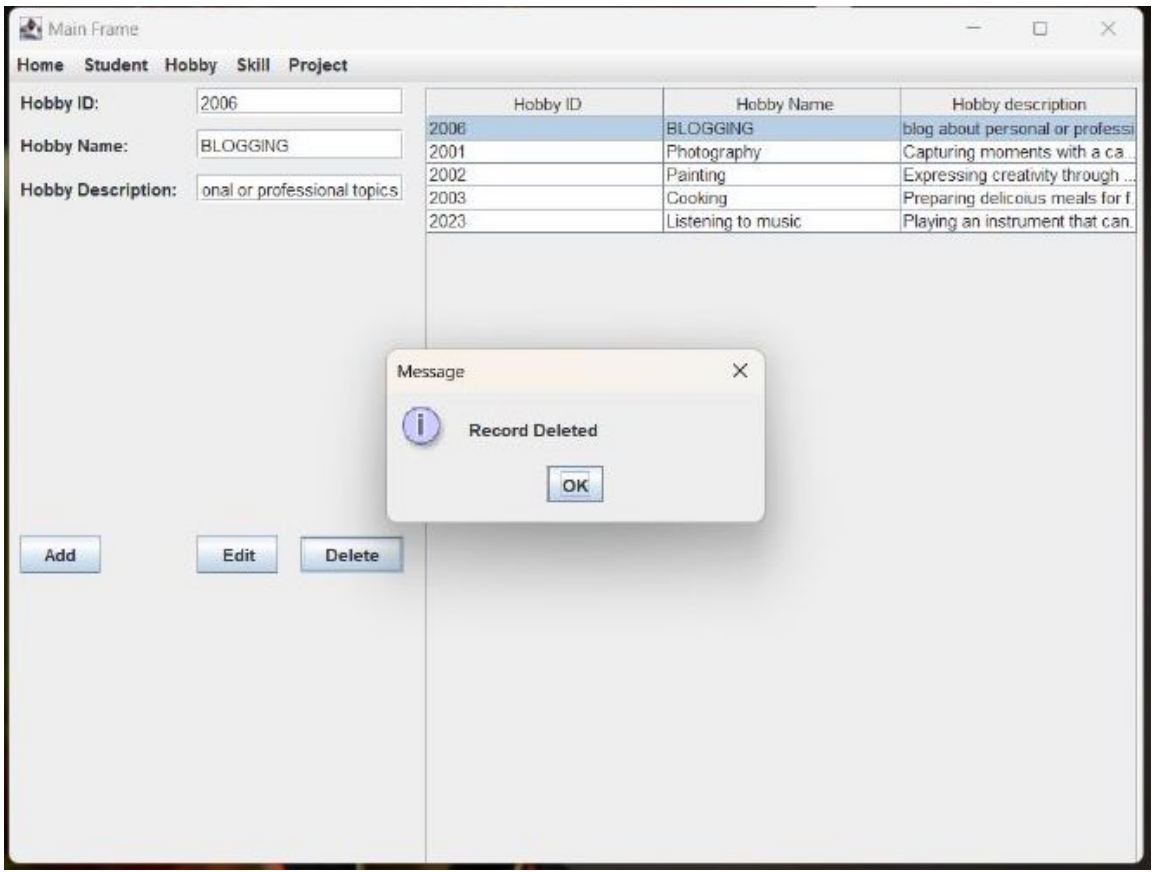
II.INSERT HOBBY TABLE INFORMATION



III.UPDATE HOBBY TABLE INFORMATION

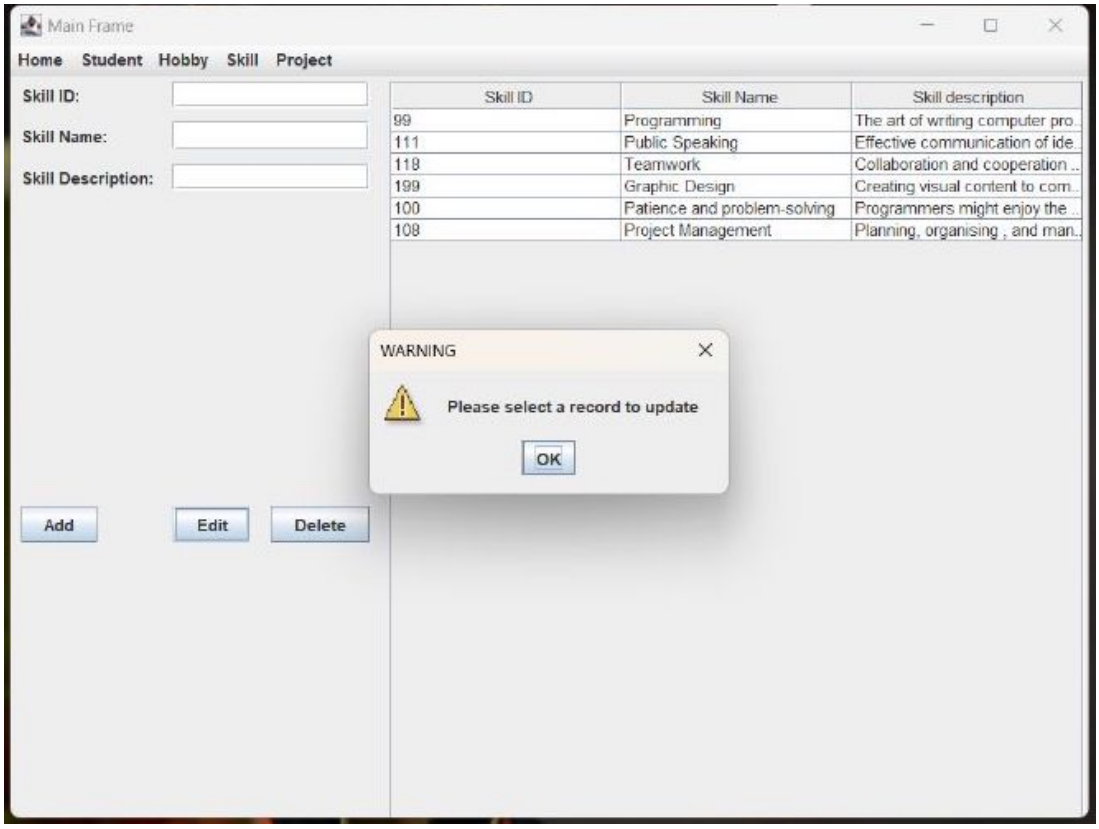


IV.DELETE HOBBY TABLE INFORMATION

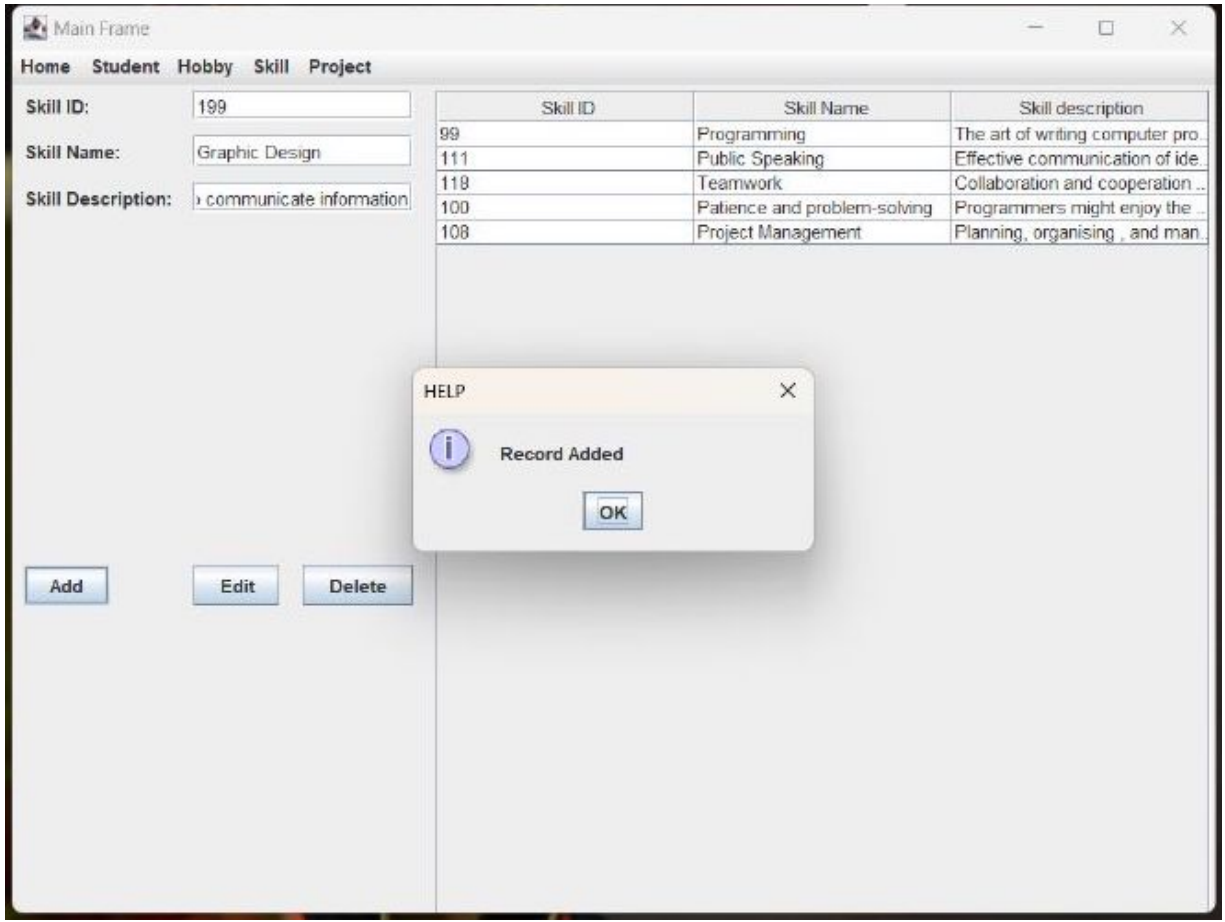


SKILL TABLE:

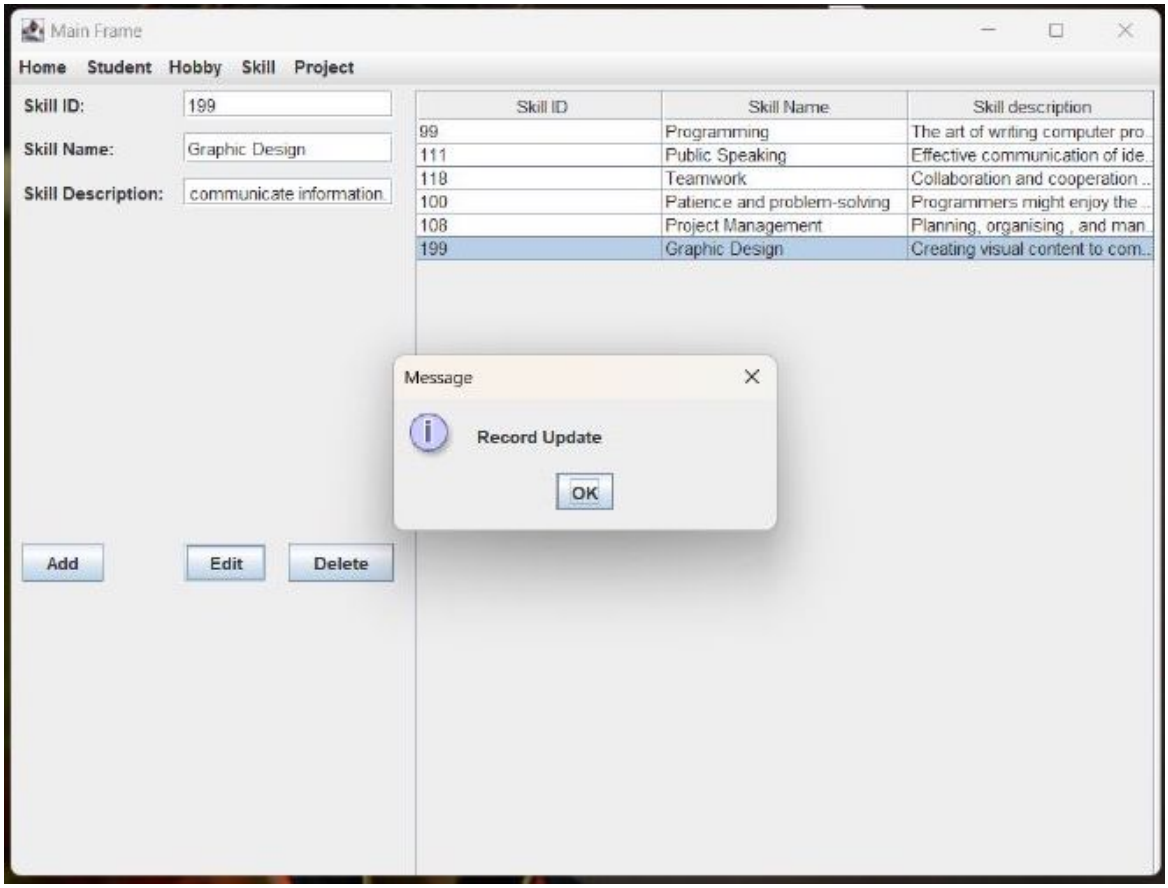
I.UPDATE WITHOUT INFORMATION



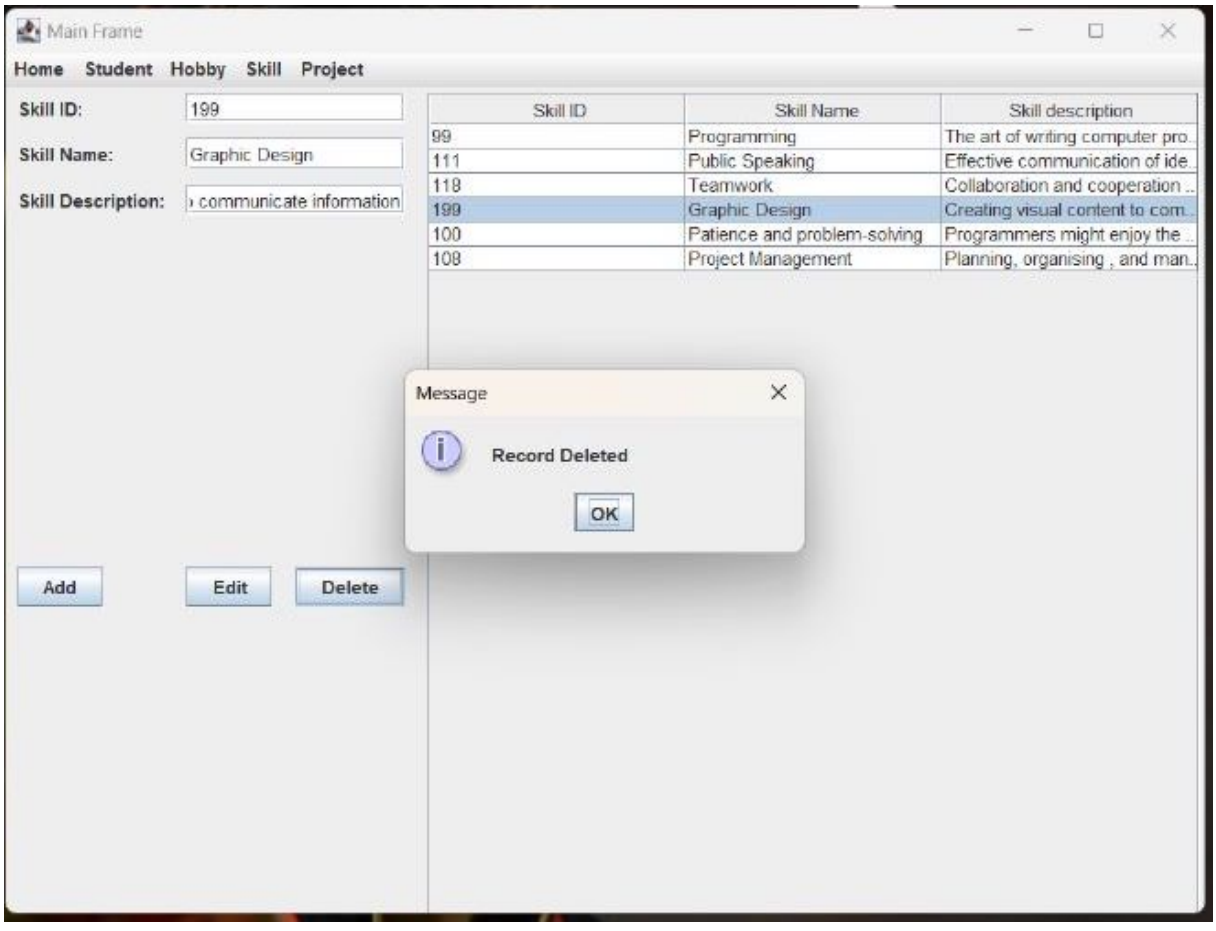
II.INSERT SKILL TABLE INFORMATION



III.UPDATE SKILL TABLE INFORMATION

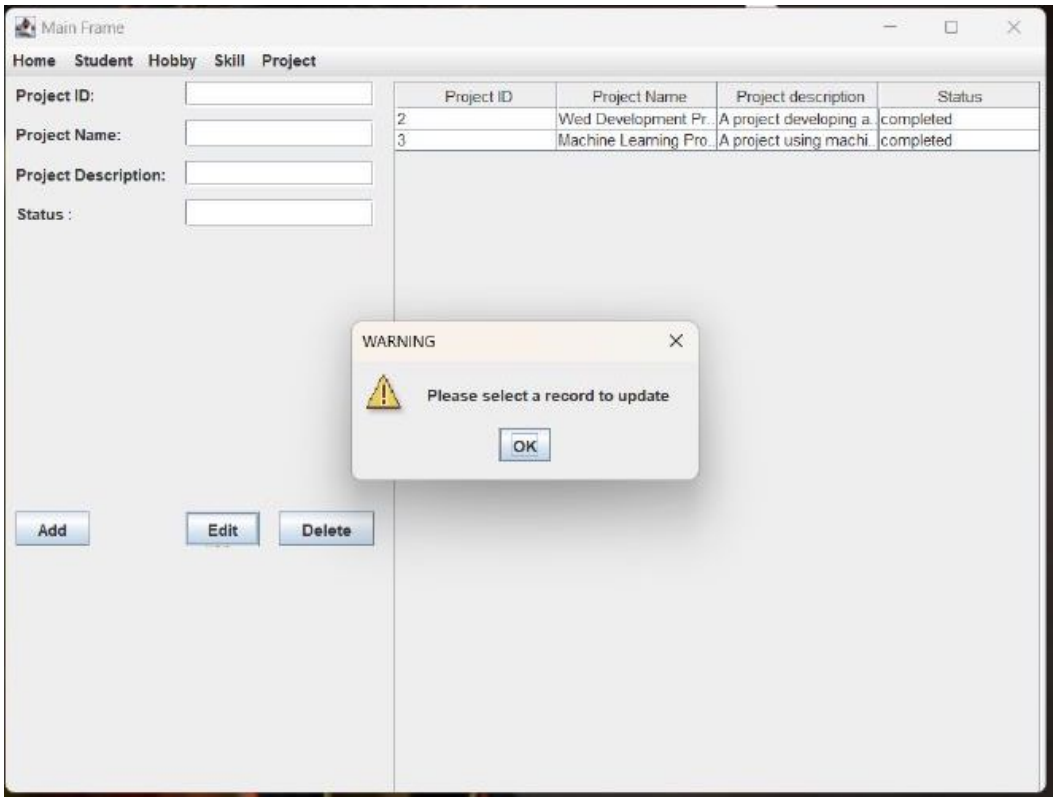


IV.DELETE SKILL TABLE INFORMATION

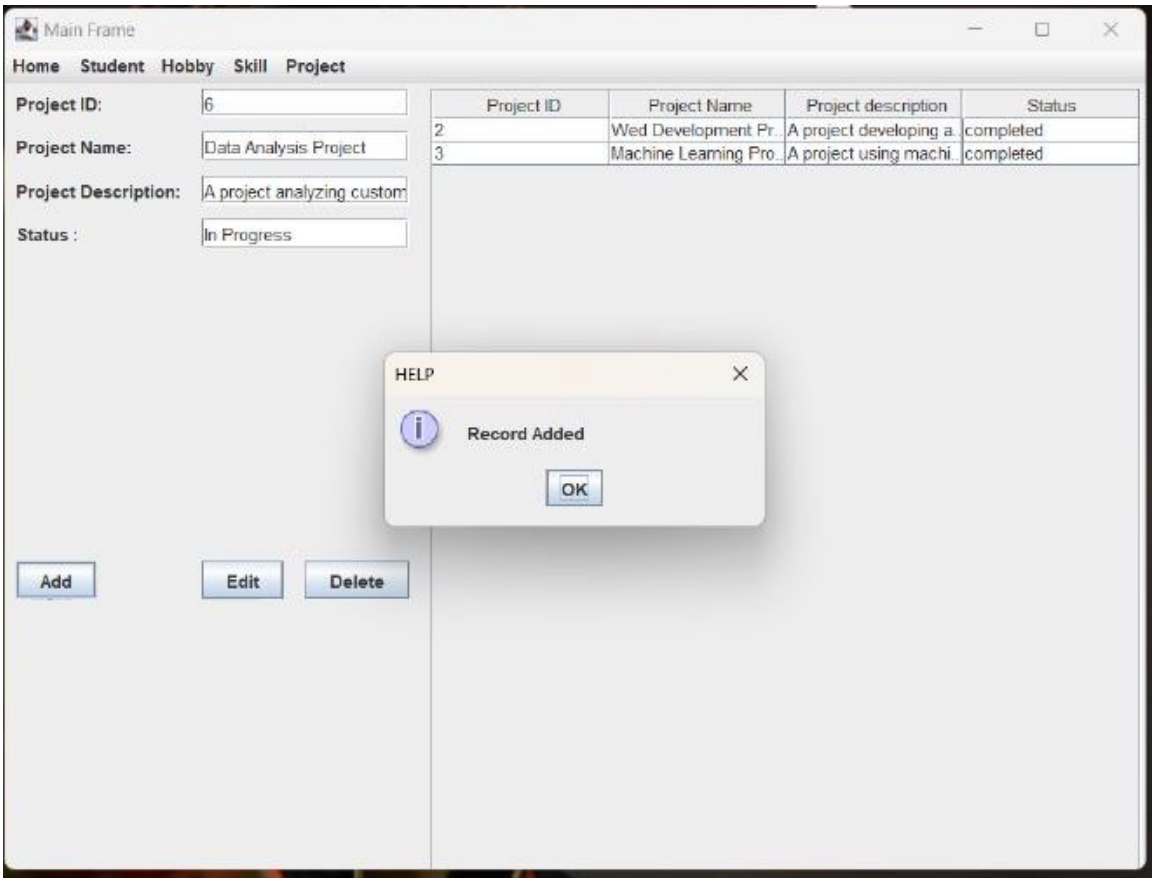


PROJECT TABLE:

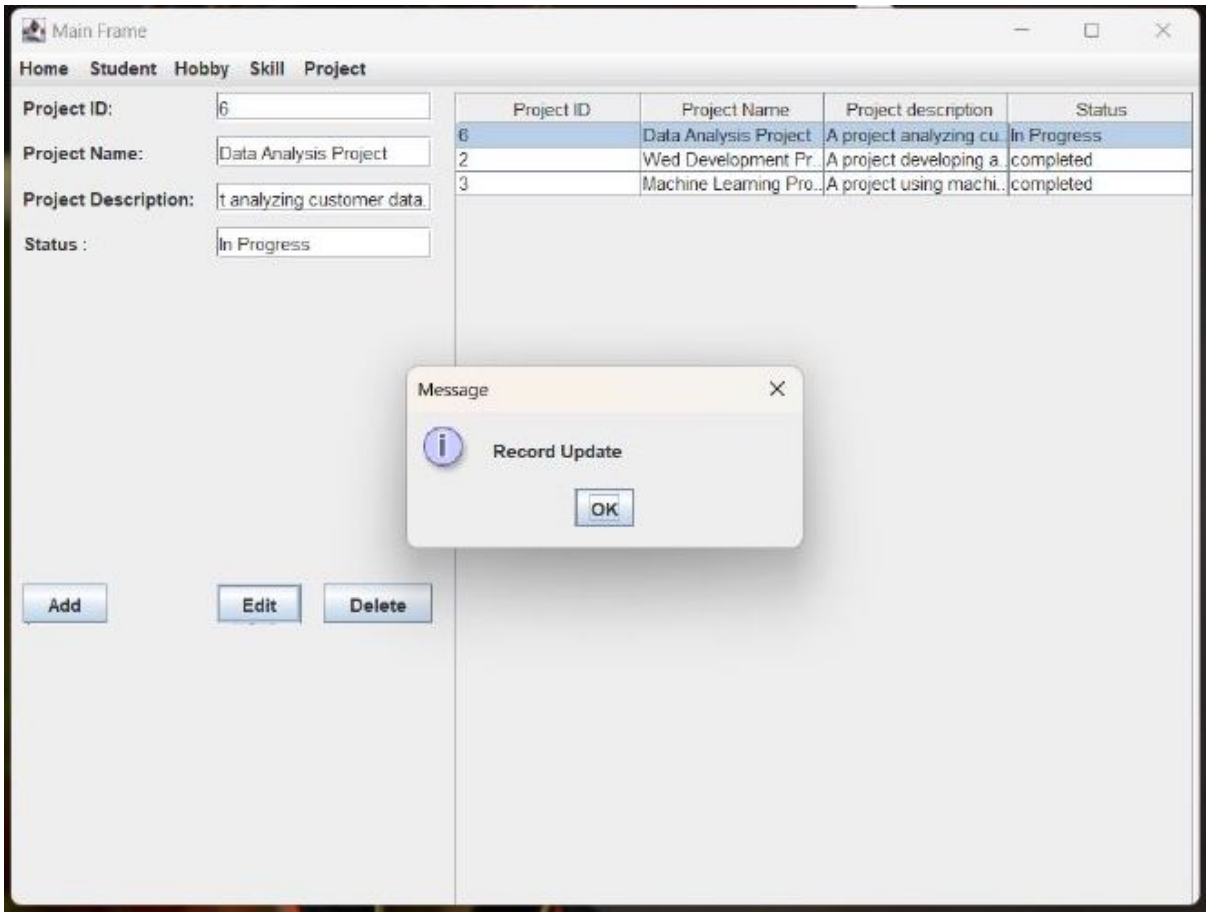
I.UPDATE WITHOUT INFORMATION



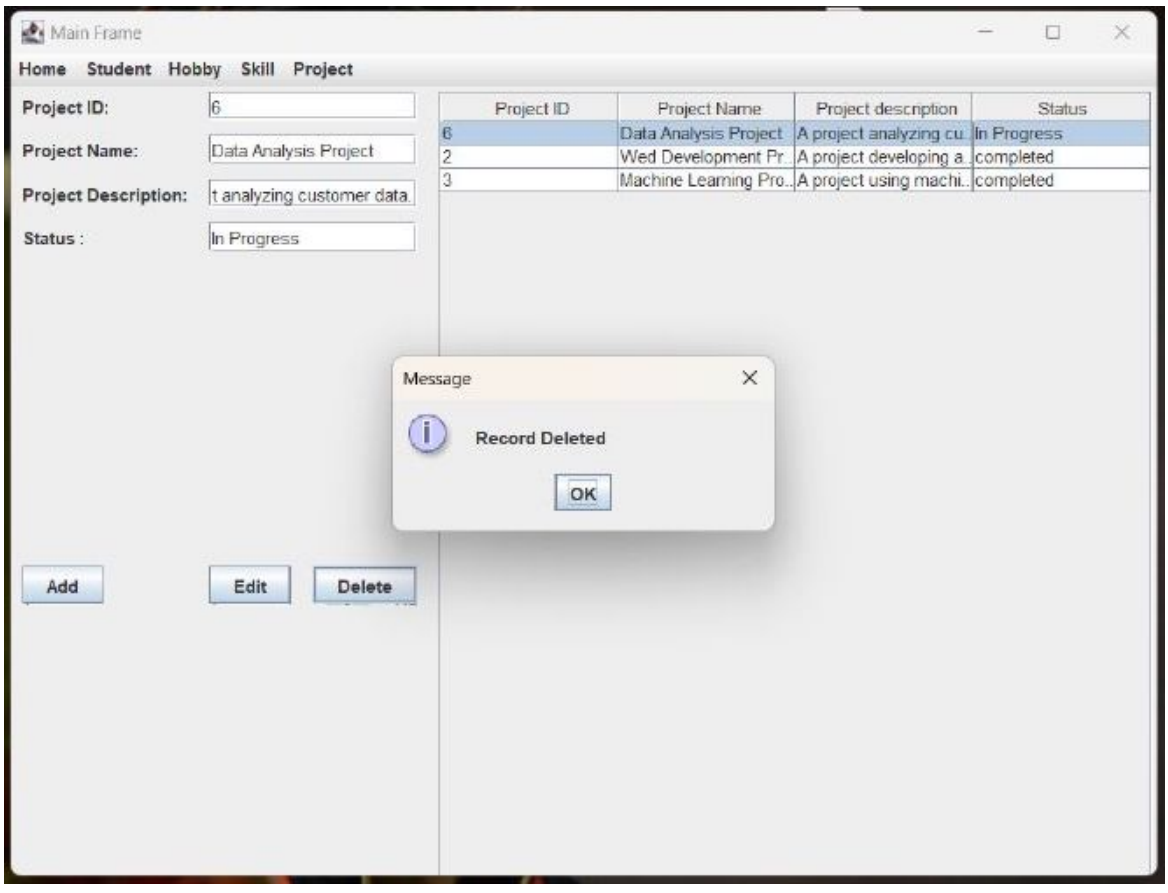
II.INSERT PROJECT TABLE INFORMATION



III.UPDATE PROJECT TABLE INFORMATION



IV.DELETE PROJECT TABLE INFORMATION



RESULTS

I have successful completed the mini-project "**HOBBIES DATABASE**"

DISCUSSION AND FUTURE WORK

For software developers, a Hobbies Database can provide valuable insights and resources related to their interests outside of coding. Here are some points to consider in relation to software developers and the Hobbies Database.By considering the unique needs and interests of software developers, the Hobbies Database can become a valuable resource for them to explore new hobbies, enhance their skills, connect with peers, and stay updated on the latest industry developments.

REFERENCES

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

1602-21-737-005

Akhil