Mid-Term Project

Session: 2021 – 2025

Submitted by:

Yasir Mahmood 2021-CS-124

Supervised by:

Mr. Nazeef-Ul-Haq

Department of Computer Science

University of Engineering and Technology

Lahore Pakistan

Acknowledgement

I would like to express my gratitude to several individuals who have helped me with this database project. First and foremost, I would like to thank my supervisor **Mr. Nazeef-Ul-Haq**, whose guidance and support have been instrumental in the development of this project.

I would also like to extend my appreciation to my class fellows at University, who have generously provided me with the necessary data and resources to complete this project. Their contributions have been invaluable to the success of this endeavor.

Finally, I would like to thank my family and friends for their unwavering support and encouragement during this challenging yet rewarding journey. Their love and encouragement have been a constant source of inspiration for me. Thank you all for your help and support!

Contents

List of Figures

Chapter 1

Introduction

A brief description of the project is discussed below:

1.1 Case Study

Department of Computer Science UET Lahore holds committee for management of final year project. Each year, list of project titles is opened for the students by the committee after the compilation of ideas from the faculty members. Students are privileged to form the group and select any of the ideas from the list after approval from the faculty advisor. Once the project is selected by a group of students, an advisory board is assigned to the project which consists of main advisor, co-advisor, and industry advisor. Throughout the year, multiple evaluations are taken against the project.For this project data base design has provided by the teacher.

1.2 Short Description

This is actually a **desktop-based application** which has made by using the .NET Framework.Following features will be implemented in the application:

- Manage Students
- Manage Advisor
- Manage Projects
- Formation of Student Group and its Management
- Assignment of Project to Student's Group
- Assignment of Multiple Advisor to Project

- Manage Evaluation
- Mark the Evaluation

1.2.1 Manage Students:

In this feature, **CRUD (Insert , Edit , Delete)** operation has performed on the person table and the student table because the id of the person is used as a primary key and as a foreign key in the student table.

1.2.2 Manage Advisor:

CRUD (Insert, Edit, Delete) operations has performed on the person and the advisor table because the id of the person is primary key which is used as foreign key in advisor table.

1.2.3 Assignment of Project to Student's Group:

In this feature, we assign the projects which was added earlier to the student groups which were also added earlier.Here the ProjectID will come from the Project table and GroupID will come from the GroupID and they both act as a primary key for the GroupProject table.In this way we assign the project to the groups.

1.2.4 Formation of Student Group and its Management:

This is the key feature in this project because firstly I perform **CRUD (Insert, Edit, Delete)** on the group table and then I add student into it. As I manage student earlier then the studentID will come from the student table and the groupID will come from the group.

1.2.5 Assignment of Multiple Advisor to Project:

As I manage advisors earlier. Now in this feature I will assign advisors to the project using the projectAdvisor table.Here the AdvisorID will come from the Advisor table which was added earlier and the projectID will come from the Project table which was also added earlier. In this way they both act as a primary key for the ProjectAdvisor table.

1.2.6 Manage Evaluation:

In this feature, I simply performed **CRUD (Edit, Update, Delete)** operation on the Evaluation table to manage the evaluation.

1.2.7 Mark the Evaluation:

In this feature, I take GroupID from the Group table which was made earlier and then the evaluationID from the Evaluation table which was also made earlier. In this way I take the evaluation of a group.

1.3 Project Users:

This project is mainly for **teachers** only. Later we could connect this project with the admin for the smooth transfer/sharing of data.

Chapter 2

Wire-Frames

Following are the samples of my project:

2.1 Main Form:

This is the main screen of the application which is shown below:

FIGURE 2.1: Main Form

2.2 Manage Student:

In this form, a grid view is shown in which edit and delete button are used. Below the grid view there are options for the add student and view student.

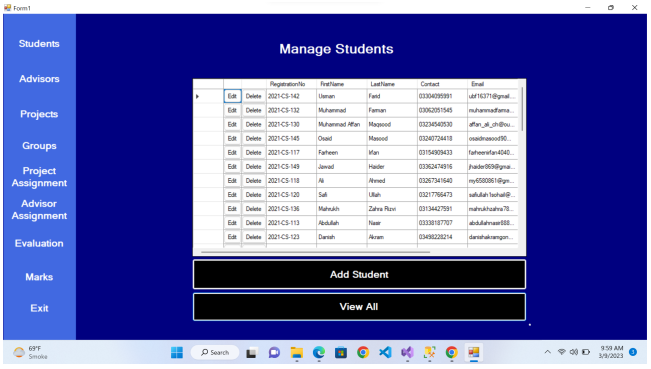

FIGURE 2.2: Manage Students

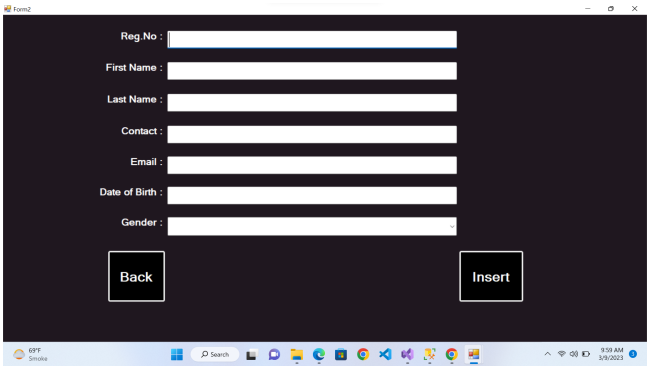

FIGURE 2.3: Add Student

2.2.1 Code for Add Student:

```
if(textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == "" || text
        {
            MessageBox.Show("Please fill all the fields.");
        }

        else
        {
            int id;

           var con = Configuration.getInstance().getConnection();
            SqlCommand cmd = new SqlCommand("Insert into Person values(@Firs
            cmd.Parameters.AddWithValue("@Firstname", textBox2.Text);
```

```
            cmd.Parameters.AddWithValue("@LastName", textBox3.Text);
            cmd.Parameters.AddWithValue("@Contact", textBox4.Text);
            cmd.Parameters.AddWithValue("@Email", textBox5.Text);
            cmd.Parameters.AddWithValue("@DateOfBirth", textBox6.Text);

            if (comboBox1.Text == "Male")
            {
                cmd.Parameters.AddWithValue("@Gender", 1);
            }

            else
            {
                cmd.Parameters.AddWithValue("@Gender", 2);
            }

cmd.ExecuteNonQuery();

            SqlCommand cmd3 = new SqlCommand("Select MAX(ID) from Person", c
            object result = cmd3.ExecuteScalar();
            id = int.Parse(result.ToString());



            SqlCommand cmd2 = new SqlCommand("Insert into Student values (@I
            cmd2.Parameters.AddWithValue("@ID", id);
            cmd2.Parameters.AddWithValue("@RegistrationNo", textBox1.Text);

            cmd2.ExecuteNonQuery();


            MessageBox.Show("Saved Successfully");
            this.Hide();
        }
```

2.2.2 Code for Edit Student:

```
if (dataGridView1.Columns[e.ColumnIndex].Name == "Delete")
        {
            string value = dataGridView1.Rows[e.RowIndex].Cells["Registratio
```

```
            var con = Configuration.getInstance().getConnection();
            SqlCommand cmd = new SqlCommand("DELETE FROM Student WHERE Regis
            cmd.Parameters.AddWithValue("@RegistrationNo", value);
            cmd.ExecuteNonQuery();

            value = dataGridView1.Rows[e.RowIndex].Cells["FirstName"].Value
            SqlCommand cmd2 = new SqlCommand("DELETE FROM Person WHERE First
            cmd2.Parameters.AddWithValue("@FirstName", value);
            cmd2.ExecuteNonQuery();
            MessageBox.Show("Successfully Deleted.");


        }

    //Form3 Edit Button event code

    var con = Configuration.getInstance().getConnection();
        SqlCommand cmd = new SqlCommand("UPDATE Student SET  RegistrationNo
        cmd.Parameters.AddWithValue("@RegistrationNo", textBox1.Text);
        cmd.Parameters.AddWithValue("@id", id);
        cmd.ExecuteNonQuery();

        string Gender = global2;
        SqlCommand cmd1 = new SqlCommand("UPDATE Person SET  FirstName = @Fi
        cmd1.Parameters.AddWithValue("@FirstName", textBox2.Text);
        cmd1.Parameters.AddWithValue("@LastName", textBox3.Text);
        cmd1.Parameters.AddWithValue("@Contact", textBox4.Text);
        cmd1.Parameters.AddWithValue("@Email", textBox5.Text);
        cmd1.Parameters.AddWithValue("@DateOfBirth", textBox6.Text);
        cmd1.Parameters.AddWithValue("@id", id);

        if (comboBox1.Text == "Male")
        {
            cmd1.Parameters.AddWithValue("@Gender", 1);

        }

        else
```

```
        {
            cmd1.Parameters.AddWithValue("@Gender", 2);
        }

        cmd1.ExecuteNonQuery();

        MessageBox.Show("Editted Successfully");
        this.Hide();
```

2.2.3 Code for Delete Student:

```
if (dataGridView1.Columns[e.ColumnIndex].Name == "Delete")
        {
            string value = dataGridView1.Rows[e.RowIndex].Cells["Registratio
            var con = Configuration.getInstance().getConnection();
            SqlCommand cmd = new SqlCommand("DELETE FROM Student WHERE Regis
            cmd.Parameters.AddWithValue("@RegistrationNo", value);
            cmd.ExecuteNonQuery();

            value = dataGridView1.Rows[e.RowIndex].Cells["FirstName"].Value
            SqlCommand cmd2 = new SqlCommand("DELETE FROM Person WHERE First
            cmd2.Parameters.AddWithValue("@FirstName", value);
            cmd2.ExecuteNonQuery();
            MessageBox.Show("Successfully Deleted.");


        }
```

2.3 Manage Advisor:

In this form, a grid view is shown in which edit and delete button are used. Below the grid view there are options for the add advisor and view advisor. On clicking the add advisor button, another form will be open to take the credentials to add advisor.

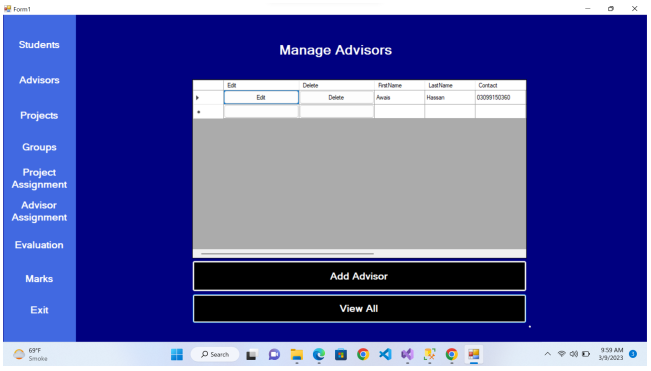

FIGURE 2.4: Manage Advisor

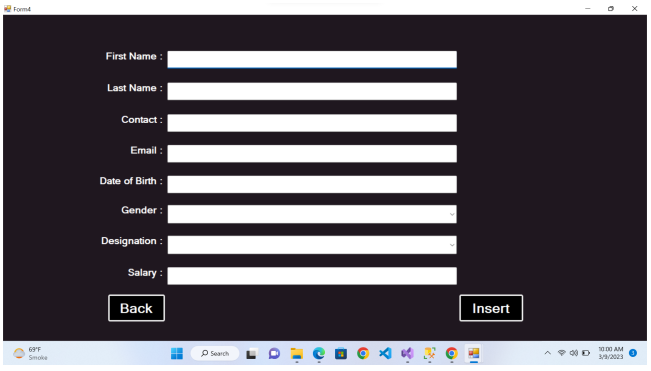

FIGURE 2.5: Add Advisor

2.3.1 Code for Add Advisor:

```
if(textBox2.Text == "" || textBox3.Text == "" || textBox4.Text == "" || text
        {
            MessageBox.Show("Please fill all the fields.");
        }

        else
        {
            int id;

            var con = Configuration.getInstance().getConnection();
```

```
SqlCommand cmd = new SqlCommand("Insert into Person values(@Firs
cmd.Parameters.AddWithValue("@Firstname", textBox2.Text);
cmd.Parameters.AddWithValue("@LastName", textBox3.Text);
cmd.Parameters.AddWithValue("@Contact", textBox4.Text);
cmd.Parameters.AddWithValue("@Email", textBox5.Text);
cmd.Parameters.AddWithValue("@DateOfBirth", textBox6.Text);

if (comboBox1.Text == "Male")
{
    cmd.Parameters.AddWithValue("@Gender", 1);
}

else
{
    cmd.Parameters.AddWithValue("@Gender", 2);
}

cmd.ExecuteNonQuery();

SqlCommand cmd3 = new SqlCommand("Select MAX(ID) from Person", c
object result = cmd3.ExecuteScalar();
id = int.Parse(result.ToString());

SqlCommand cmd2 = new SqlCommand("Insert into Advisor values (@I
cmd2.Parameters.AddWithValue("@ID", id);

if(comboBox2.Text == "Professor")
{
    cmd2.Parameters.AddWithValue("@Designation", 6);
}

else if(comboBox2.Text == "Associate Professor")
{
    cmd2.Parameters.AddWithValue("@Designation", 7);
}

else if(comboBox2.Text == "Assisstant Professor")
{
```

```
                cmd2.Parameters.AddWithValue("@Designation", 8);
            }

            else if(comboBox2.Text == "Lecturer")
            {
                cmd2.Parameters.AddWithValue("@Designation", 9);
            }

            else if(comboBox2.Text == "Industry Professional")
            {
                cmd2.Parameters.AddWithValue("@Designation", 10);
            }

            cmd2.Parameters.AddWithValue("@Salary", textBox7.Text);

            cmd2.ExecuteNonQuery();


            MessageBox.Show("Saved Successfully");
            this.Hide();
```

2.3.2 Code for Edit Advisor:

```
else if (dataGridView1.Columns[e.ColumnIndex].Name == "Edit")
      {

          string value1 = dataGridView1.Rows[e.RowIndex].Cells["FirstName"
          string value2 = dataGridView1.Rows[e.RowIndex].Cells["LastName"]
          string value3 = dataGridView1.Rows[e.RowIndex].Cells["Contact"]
          string value4 = dataGridView1.Rows[e.RowIndex].Cells["Email"].Va
          string value5 = dataGridView1.Rows[e.RowIndex].Cells["DateOfBirt
          string value6 = dataGridView1.Rows[e.RowIndex].Cells["Gender"].V
          string value7 = dataGridView1.Rows[e.RowIndex].Cells["Designatio
          string value8 = dataGridView1.Rows[e.RowIndex].Cells["Salary"].V
          Form5 f = new Form5(value1 , value2 , value3 , value4 , value5 ,
          f.ShowDialog();
      }

  //Form5 Edit Button event code
```

```
var con = Configuration.getInstance().getConnection();
SqlCommand cmd = new SqlCommand("UPDATE Advisor SET  Designation = C



if(comboBox2.Text == "Professor")
{
    cmd.Parameters.AddWithValue("@Designation", 6);
}

else if(comboBox2.Text == "Associate Professor")
{
    cmd.Parameters.AddWithValue("@Designation", 7);
}

else if(comboBox2.Text == "Assisstant Professor")
{
    cmd.Parameters.AddWithValue("@Designation", 8);
}

else if(comboBox2.Text == "Lecturer")
{
    cmd.Parameters.AddWithValue("@Designation", 9);
}

else
{
    cmd.Parameters.AddWithValue("@Designation", 10);
}
cmd.Parameters.AddWithValue("@Salary", textBox7.Text);
cmd.Parameters.AddWithValue("@id", id);
cmd.ExecuteNonQuery();

string Gender = global1;
SqlCommand cmd1 = new SqlCommand("UPDATE Person SET  FirstName = @Fi
cmd1.Parameters.AddWithValue("@FirstName", textBox2.Text);
cmd1.Parameters.AddWithValue("@LastName", textBox3.Text);
cmd1.Parameters.AddWithValue("@Contact", textBox4.Text);
```

```
cmd1.Parameters.AddWithValue("@Email", textBox5.Text);
cmd1.Parameters.AddWithValue("@DateOfBirth", textBox6.Text);
cmd1.Parameters.AddWithValue("@id", id);

if (comboBox1.Text == "Male")
{
    cmd1.Parameters.AddWithValue("@Gender", 1);

}

else
{
    cmd1.Parameters.AddWithValue("@Gender", 2);
}

cmd1.ExecuteNonQuery();

MessageBox.Show("Editted Successfully");
this.Hide();
```

2.3.3 Code for Delete Student:

```
if (dataGridView1.Columns[e.ColumnIndex].Name == "Delete")
        {
            string value = dataGridView1.Rows[e.RowIndex].Cells["Salary"].Va
            var con = Configuration.getInstance().getConnection();
            SqlCommand cmd = new SqlCommand("DELETE FROM Advisor WHERE Salar
            cmd.Parameters.AddWithValue("@Salary", value);
            cmd.ExecuteNonQuery();

            value = dataGridView1.Rows[e.RowIndex].Cells["FirstName"].Value
            SqlCommand cmd2 = new SqlCommand("DELETE FROM Person WHERE First
            cmd2.Parameters.AddWithValue("@FirstName", value);
            cmd2.ExecuteNonQuery();
            MessageBox.Show("Successfully Deleted.");

        }
```

2.4 Manage Project:

In this form, a grid view is shown in which edit and delete button are used. Below the grid view there are options for the add project and view delete. On clicking the add advisor button, another form will be open to take the credentials for adding project.

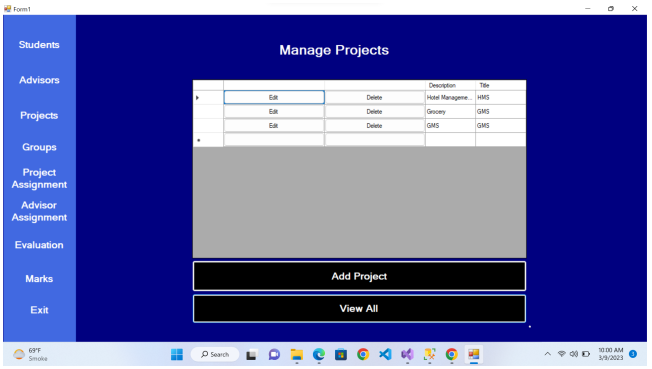

FIGURE 2.6: Manage Project

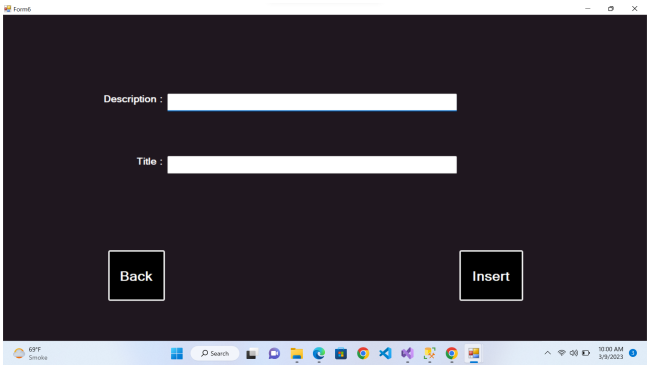

FIGURE 2.7: Add Project

2.4.1 Code for Add Project:

```
if(textBox5.Text == "" || textBox6.Text == "")
        {
            MessageBox.Show("Please fill all the fields.");
        }

        else
        {
            var con = Configuration.getInstance().getConnection();
            SqlCommand cmd2 = new SqlCommand("Insert into Project values (@
            cmd2.Parameters.AddWithValue("@Description", textBox5.Text);
```

```
            cmd2.Parameters.AddWithValue("@title", textBox6.Text);

            cmd2.ExecuteNonQuery();


            MessageBox.Show("Saved Successfully");
            this.Hide();
        }
```

2.4.2 Code for Edit Project:

```
 private void button2_Click(object sender, EventArgs e)
    {
        // string Gender = global1;
        var con = Configuration.getInstance().getConnection();
        SqlCommand cmd1 = new SqlCommand("UPDATE Project SET  Description =
        cmd1.Parameters.AddWithValue("@Description", textBox5.Text);
        cmd1.Parameters.AddWithValue("@Title", textBox6.Text);
        cmd1.Parameters.AddWithValue("@id", id);


        cmd1.ExecuteNonQuery();

        MessageBox.Show("Editted Successfully");
        this.Hide();
    }
```

2.4.3 Code for Delete project:

```
if (dataGridView1.Columns[e.ColumnIndex].Name == "Delete")
        {
            string value = dataGridView1.Rows[e.RowIndex].Cells["Description
            var con = Configuration.getInstance().getConnection();
            SqlCommand cmd = new SqlCommand("DELETE FROM Project WHERE Descr
            cmd.Parameters.AddWithValue("@Description", value);
            cmd.ExecuteNonQuery();

            MessageBox.Show("Successfully Deleted.");
        }
```

2.5 Manage Group:

In this form, a grid view is shown in which delete button is used. Above the grid view there is a button to make a group by giving the assignment date from the date time piker.In this way the group is formed. After this you can select the studentID and GroupID from the drop down menu and then make the add the students to the groups.

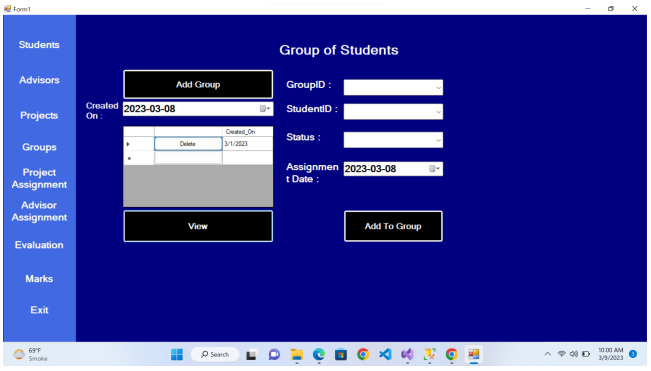

FIGURE 2.8: Manage Groups

2.5.1 Code for Add Group:

```
private void button1_Click(object sender, EventArgs e)
    {
        label1.Visible = true;
        dateTimePicker1.Visible = true;




        string value = dateTimePicker1.Text;

        var con = Configuration.getInstance().getConnection();
        SqlCommand cmd2 = new SqlCommand("Insert into [Group] values (@Creat
        cmd2.Parameters.AddWithValue("@Created_On", value);
        cmd2.ExecuteNonQuery();

        MessageBox.Show("Saved Successfully");

    }
```

2.5.2 Code for View Group :

```
private void button2_Click(object sender, EventArgs e)
    {

        //dataGridView1.ReadOnly = true;

        var con = Configuration.getInstance().getConnection();
        SqlCommand cmd = new SqlCommand("Select Created_On from [Group]", co
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        DataTable dt = new DataTable();
        da.Fill(dt);
        dataGridView1.DataSource = dt;
        dataGridView1.Columns["Created_On"].DefaultCellStyle.ForeColor = Col

    }
```

2.5.3 Code for Drop Down Menu:

```
\\For taking GroupID
 private void comboBox1_Click(object sender, EventArgs e)
    {
        var con = Configuration.getInstance().getConnection();
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        SqlCommand cmd = new SqlCommand("Select Id From [Group]", con);
        SqlDataReader reader = cmd.ExecuteReader();
        // Create list to store retrieved data
        List<object> dataList = new List<object>();
        // Loop through reader and add retrieved data to list
        while (reader.Read())
        {
            // Retrieve data from reader
            object data = reader["Id"]; // ider apnay column ka name likha h
            // Add data to list
            dataList.Add(data);
        }
```

```
        // Close reader and connection
        reader.Close();

        comboBox1.DataSource = dataList;
    }

    \\For taking studentID
     private void comboBox2_Click(object sender, EventArgs e)
    {
        var con = Configuration.getInstance().getConnection();
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        SqlCommand cmd2 = new SqlCommand("Select Id From Student", con);
        SqlDataReader reader2 = cmd2.ExecuteReader();
        // Create list to store retrieved data
        List<object> dataList2 = new List<object>();
        // Loop through reader and add retrieved data to list
        while (reader2.Read())
        {
            // Retrieve data from reader
            object data2 = reader2["Id"]; // ider apnay column ka name likha
            // Add data to list
            dataList2.Add(data2);
        }
        // Close reader and connection
        reader2.Close();

        comboBox2.DataSource = dataList2;
    }
```

2.5.4 Code for Add Student to Group :

```
        if(comboBox1.Text == "" || comboBox2.Text == "" || comboBox3.Text ==
        {
            MessageBox.Show("Please fill all the fields");
        }
```

```
else if(comboBox3.Text != "Active" && comboBox3.Text != "In Active")
{
    MessageBox.Show("Selected Status is not valid");
}

else
{
    var con = Configuration.getInstance().getConnection();
    SqlCommand cmd2 = new SqlCommand("Insert into GroupStudent value
    cmd2.Parameters.AddWithValue("@GroupID", comboBox1.Text);
    cmd2.Parameters.AddWithValue("@StudentID", comboBox2.Text);

    if(comboBox3.Text == "Active")
    {
        cmd2.Parameters.AddWithValue("@Status", 3);
    }

    else if(comboBox3.Text == "In Active")
    {
        cmd2.Parameters.AddWithValue("@Status", 4);
    }

    cmd2.Parameters.AddWithValue("@AssignmentDate", dateTimePicker1

    cmd2.ExecuteNonQuery();


    MessageBox.Show("Saved Successfully");
    comboBox1.SelectedIndex = -1;
    comboBox2.SelectedIndex = -1;

    comboBox3.SelectedIndex = -1;
    //date
```

2.6 Project Assignment to Group:

In this form, projectID, groupID will be taken from drop down menu and the date will be given from the date time piker and then click on the assign project to assign a particular project to a particular group.

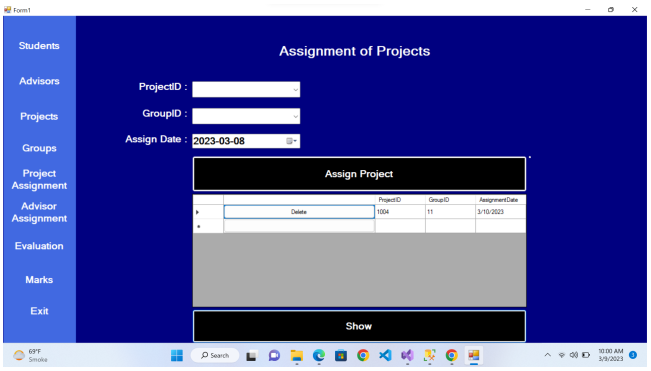

FIGURE 2.9: Manage Project

2.6.1 Code for Drop Down Menu:

```
\\For taking GroupID
 private void comboBox2_Click(object sender, EventArgs e)
    {
        var con = Configuration.getInstance().getConnection();
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        SqlCommand cmd = new SqlCommand("Select Id From [Group]", con);
        SqlDataReader reader = cmd.ExecuteReader();
        // Create list to store retrieved data
        List<object> dataList = new List<object>();
        // Loop through reader and add retrieved data to list
        while (reader.Read())
        {
            // Retrieve data from reader
            object data = reader["Id"]; // ider apnay column ka name likha h
            // Add data to list
            dataList.Add(data);
        }
        // Close reader and connection
```

```
            reader.Close();

            comboBox2.DataSource = dataList;
        }

        \\For taking ProjectID
        private void comboBox1_Click(object sender, EventArgs e)
        {
            var con = Configuration.getInstance().getConnection();
            if (con.State == ConnectionState.Closed)
            {
                con.Open();
            }
            SqlCommand cmd = new SqlCommand("Select Id From Project", con);
            SqlDataReader reader = cmd.ExecuteReader();
            // Create list to store retrieved data
            List<object> dataList = new List<object>();
            // Loop through reader and add retrieved data to list
            while (reader.Read())
            {
                // Retrieve data from reader
                object data = reader["Id"]; // ider apnay column ka name likha h
                // Add data to list
                dataList.Add(data);
            }
            // Close reader and connection
            reader.Close();

            comboBox1.DataSource = dataList;
        }
```

2.6.2 Code for Assign project :

```
    if(comboBox1.Text == "" || comboBox2.Text == "" || dateTimePicker2.Text ==
            {
                MessageBox.Show("Please fill all the fields");
            }

            else
```

```
{
    var con = Configuration.getInstance().getConnection();
    if (con.State == ConnectionState.Closed)
    {
        con.Open();
    }
    SqlCommand cmd = new SqlCommand("Insert into GroupProject (Proje
    cmd.Parameters.AddWithValue("@ProjectId", comboBox1.Text);
    cmd.Parameters.AddWithValue("@GroupId", comboBox2.Text);
    cmd.Parameters.AddWithValue("@AssignmentDate", dateTimePicker2.
    cmd.ExecuteNonQuery();
    MessageBox.Show("Data Inserted Successfully");
    comboBox1.SelectedIndex = -1;
    comboBox2.SelectedIndex = -1;



}
```

2.7 Advisors Assignment to Project:

In this form, advisorID, projectID will be taken from drop down menu , the date will be given from the date time piker , advisor role will be taken from the LookUp table and then click on the assign assign to assign a particular advisor to a particular project.

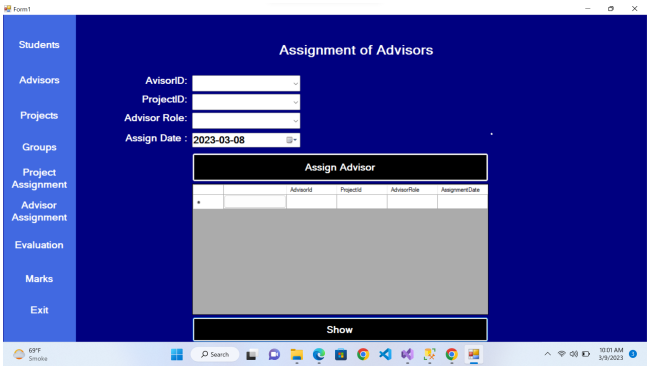

FIGURE 2.10: Advisor Assignment to Project

2.7.1 Code for Drop Down Menu :

```
\\For Picking AdvisorID
 private void comboBox1_Click(object sender, EventArgs e)
    {
        var con = Configuration.getInstance().getConnection();
        if (con.State == ConnectionState.Closed)
        {
            con.Open();
        }
        SqlCommand cmd = new SqlCommand("Select Id From Advisor", con);
        SqlDataReader reader = cmd.ExecuteReader();
        // Create list to store retrieved data
        List<object> dataList = new List<object>();
        // Loop through reader and add retrieved data to list
        while (reader.Read())
        {
            // Retrieve data from reader
            object data = reader["Id"]; // ider apnay column ka name likha h
            // Add data to list
            dataList.Add(data);
        }
```

```
            // Close reader and connection
            reader.Close();

            comboBox1.DataSource = dataList;


        \\For Picking ProjectID
         private void comboBox2_Click(object sender, EventArgs e)
        {
            var con = Configuration.getInstance().getConnection();
            if (con.State == ConnectionState.Closed)
            {
                con.Open();
            }
            SqlCommand cmd = new SqlCommand("Select Id From Project", con);
            SqlDataReader reader = cmd.ExecuteReader();
            // Create list to store retrieved data
            List<object> dataList = new List<object>();
            // Loop through reader and add retrieved data to list
            while (reader.Read())
            {
                // Retrieve data from reader
                object data = reader["Id"]; // ider apnay column ka name likha h
                // Add data to list
                dataList.Add(data);
            }
            // Close reader and connection
            reader.Close();

            comboBox2.DataSource = dataList;
        }
```

2.7.2 Code for Assign Advisor :

```
    if(comboBox1.Text == "" || comboBox2.Text == "" || comboBox3.Text == "" || c
            {
                MessageBox.Show("Please fill all the fields");
            }
```

```
else if(comboBox3.Text != "Main Advisor" && comboBox3.Text != "Co-Ad
{
    MessageBox.Show("Selected Role is Invalid");
}

else
{
    var con = Configuration.getInstance().getConnection();
    if (con.State == ConnectionState.Closed)
    {
        con.Open();
    }
    SqlCommand cmd = new SqlCommand("Insert into ProjectAdvisor (Adv
    cmd.Parameters.AddWithValue("@AdvisorID", comboBox1.Text);
    cmd.Parameters.AddWithValue("@ProjectID", comboBox2.Text);
    cmd.Parameters.AddWithValue("@AssignmentDate", dateTimePicker2.T

    if(comboBox3.Text == "Main Advisor")
    {
        cmd.Parameters.AddWithValue("@AdvisorRole", 11);
    }

    else if(comboBox3.Text == "Co-Advisror")
    {
        cmd.Parameters.AddWithValue("@AdvisorRole", 12);
    }

    else if (comboBox3.Text == "Industry Advisor")
    {
        cmd.Parameters.AddWithValue("@AdvisorRole", 13);
    }

    cmd.ExecuteNonQuery();
    MessageBox.Show("Data Inserted Successfully");
    comboBox1.SelectedIndex = -1;
    comboBox2.SelectedIndex = -1;
    comboBox3.SelectedIndex = -1;
}
```

2.8 Manage Evaluations:

In this form, grid view show the evaluations and when the add evaluation button is clicked then another form will open which will take credentials to add evaluations.

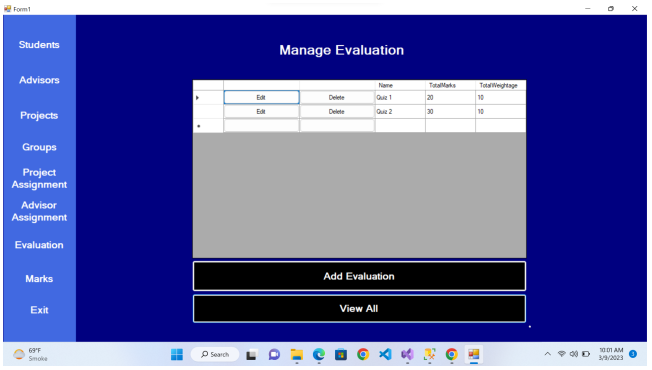

FIGURE 2.11: Manage Evaluation

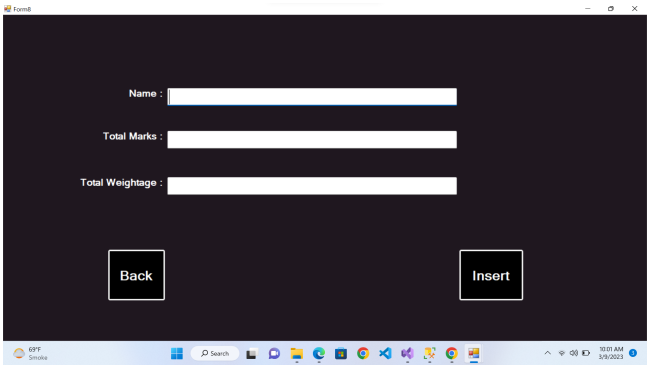

FIGURE 2.12: Add Evaluation

2.8.1 Code for Add Evaluation :

```
if(textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == "")
        {
            MessageBox.Show("Please fill all the fields.");
        }

        else
        {
            var con = Configuration.getInstance().getConnection();
            SqlCommand cmd2 = new SqlCommand("Insert into Evaluation values
            cmd2.Parameters.AddWithValue("@Name", textBox1.Text);
            cmd2.Parameters.AddWithValue("@TotalMarks", int.Parse(textBox2.
```

```
            cmd2.Parameters.AddWithValue("@TotalWeightage", int.Parse(textBo

            cmd2.ExecuteNonQuery();


            MessageBox.Show("Saved Successfully");
            this.Hide();
        }
```

2.8.2 Code for Edit Evaluation :

```
else if (dataGridView1.Columns[e.ColumnIndex].Name == "Edit")
        {
            string value1 = dataGridView1.Rows[e.RowIndex].Cells["Name"].Val
            string value2 = dataGridView1.Rows[e.RowIndex].Cells["TotalMarks
            string value3 = dataGridView1.Rows[e.RowIndex].Cells["TotalWeigh


            Form9 f = new Form9(value1 , value2 , value3);
            // this.Hide();
            f.ShowDialog();
        }

        \\form9 edit button click code
         private void button2_Click(object sender, EventArgs e)
    {
        var con = Configuration.getInstance().getConnection();
        SqlCommand cmd1 = new SqlCommand("UPDATE Evaluation SET  Name = @Nam
        cmd1.Parameters.AddWithValue("@Name", textBox1.Text);
        cmd1.Parameters.AddWithValue("@TotalMarks", int.Parse(textBox2.Text)
        cmd1.Parameters.AddWithValue("@TotalWeightage", int.Parse(textBox3.T
        cmd1.Parameters.AddWithValue("@id", id);


        cmd1.ExecuteNonQuery();

        MessageBox.Show("Editted Successfully");
        this.Hide();
    }
```

2.8.3 Code for Delete Evaluation :

```
if (dataGridView1.Columns[e.ColumnIndex].Name == "Delete")
        {
            string value = dataGridView1.Rows[e.RowIndex].Cells["Name"].Valu
            var con = Configuration.getInstance().getConnection();
            SqlCommand cmd = new SqlCommand("DELETE FROM Evaluation WHERE Na
            cmd.Parameters.AddWithValue("@Name", value);
            cmd.ExecuteNonQuery();


            MessageBox.Show("Successfully Deleted.");


        }
```

Chapter 3

Final Thoughts

This chapter discusses the overall experience when doing this project. Also issues faced while making this project.

3.1 Issues faced

When making GUI alot of issues weer faced. From Visual Studio being crashed to all the annoying bugs. But the most important issue of them all, were the Foreign Key issues. Details of some the issues are discussed below:

3.1.1 Foreign Key Issue:-

The type of data base that was provided was some what connected in some way. The relations were connected in such a way that ID of X relation was used in relation Y. Therefore when performing one of the CRUD operations on Relations, it would give the error called the *"Foreign Key"* error.

3.1.2 GUI Issues:-

Although the GUI made was relatively easy, but I faced some really annoying issues, like the button keeps disappearing etc. Some of the problems faced are discussed below:

3.1.2.1 Bring to Front

When working with Side panels especially multiple of them, there are high chances that we mix them altogether. For example I faced an issue where the side panel should Theoretically change colour when pressing the button. But instead it kept disappearing. The issue was resolved by **Bringing it to front**.

3.1.2.2 UserControls

The theme of my GUI was several Forms and numerous User controls. The issue I faced was how I manipulated them. This was done by click events, like when pressing the Student button, student control should show up and etc.

3.1.3 Future Improvements

As there is no login form due to which any one can excess this application easily which is not good regarding the security issues. There is only one user in the application as I discussed earlier which is only a teacher. So two improvements should be done which are as under :

- There should be a login page to resolve the issue of security.
- Further more we can add more users in this application for example students can be the user so that they can view their marks , project evaluation etc.

3.2 My Remarks

The process of this project was really great. I enjoyed making this project on C#. It really is a sort of joy when you make a GUI in your head and then see you GUI in action on the screen. Although I wasn't able to complete the entire project like not completing the Evaluation of students, But I am satisfied with the amount of effort that I put in.