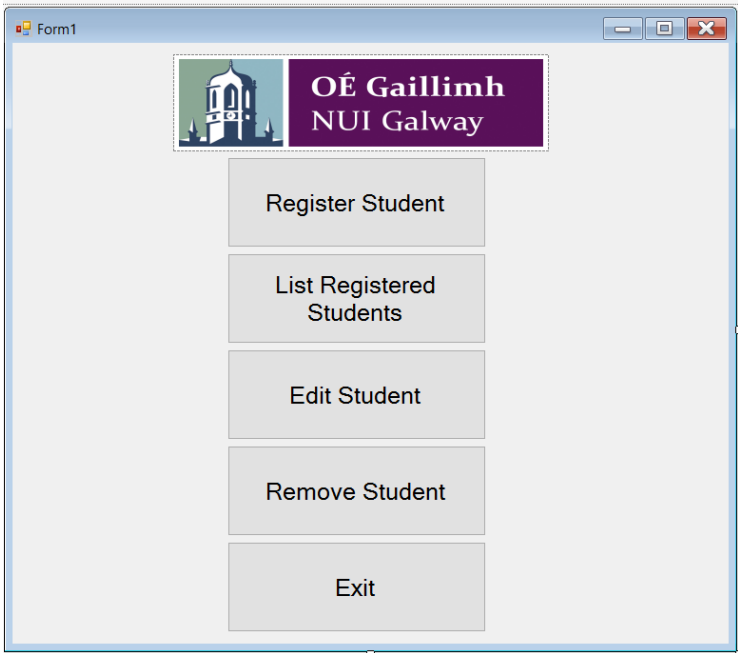


Name	Francis Murtagh
Student Number	12347506
CT331	

Source Code: Main.cs	Explanation
<pre>Application.Run(new Form1());</pre>	Runs the main menu form

Source Code: form1.cs	Explanation
	<p>Main menu of registration system With buttons labels and picturebox</p> <p>With 5 option as seen.</p>

Source Code: form1.cs	Explanation
<div data-bbox="204 322 598 757" data-label="Image"> </div> <pre> private void btn_regsave_Click(object sender, EventArgs e) { try { connection.ConnectionString = connection.Open(); String idstring = txtbox_Id.Text; String fnstring = txtbox_firstName.Text; String lnstring = txtbox_lastName.Text; String emailstring = txtbox_Email.Text; String addressstring = txtbox_Address.Text; String subjectstring = txtbox_Subject.Text; String caostring = txtbox_Cao.Text; String degreestring = txtbox_degreeGrade.Text; var cmd = new OleDbCommand("INSERT INTO Student ([Id],[Firstname], [Lastname] ,[Email] , [Address], [Subject], [intCAOdata], [doubleDegreegrade]) VALUES (@a, @b, @c, @d, @e, @f, @g, @h)"); cmd.Connection = connection; cmd.Parameters.AddRange(new[] { new OleDbParameter("@a", idstring), new OleDbParameter("@b", fnstring), new OleDbParameter("@c", lnstring), new OleDbParameter("@d", emailstring), new OleDbParameter("@e", addressstring), new OleDbParameter("@f", subjectstring), new OleDbParameter("@g", caostring), new OleDbParameter("@h", degreestring) }); cmd.ExecuteNonQuery(); MessageBox.Show("Student registered successfully."); this.Hide(); } catch (Exception ex) { MessageBox.Show("Error " + ex); } connection.Close(); } </pre>	<p>On pressing the register student button the user is presented with another form allowing them to enter the new students information</p> <p>It will then be sent to the access database when the “register new student” button is pressed</p> <p>I used parameterized sql queries as they are more secure than sql insertion</p>

Source Code: FormList.cs

Explanation

IDprimary	Id	FirstName	LastName	Email	Address	Subject
74	Id	FirstName	LastName	Email	Address	Subject
76	Id	FirstName	LastName	Email	Address	Subject
77	Id	FirstName	LastName	Email	Address	Subject
78	Id	FirstName	LastName	Email	Address	Subject
81	1234	Francis	Murtagh	francis@gmail	tubber	physics
82	5	francis	murtagh	fsdg	ghfgh	h
*						

Exit

```
private void FormList_Load(object sender, EventArgs e)
{
    this.studentTableAdapter.Fill(this.studentregdbDataSet.Student);
}

private void btn_ExitList_Click(object sender, EventArgs
e)
{
    this.Hide();
}
```

On pressing the “list registered students” button the user is presented with a dataset box outlining the information in the database

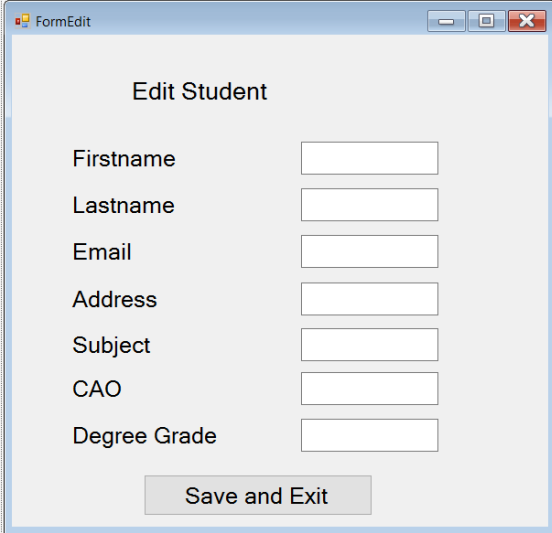
As well as an exit button to return to the main menu.

This screenshot is from when sql query had incorrectly referenced values but these work in the final version of the program.

Source Code: FormEdit.cs	Explanation
<div data-bbox="204 488 855 891" data-label="Image"> </div> <pre> command.CommandText = "select * from Student where Id='" + txtbox_editId.Text + "'"; OleDbDataReader reader = command.ExecuteReader(); int count = 0; while (reader.Read()) { count = count + 1; } if (count == 1) { MessageBox.Show("Student Id found."); this.Hide(); FormEdit f5 = new FormEdit(); f5.lookupId = lookupId; f5.ShowDialog(); } else if (count > 1) { MessageBox.Show("Duplicate Student Id."); } else { MessageBox.Show("Incorrect Student Id."); } </pre>	<p>Pressing the Edit student button gives a lookup for which searches the database Giving an error if the id is not present or is duplicated</p>

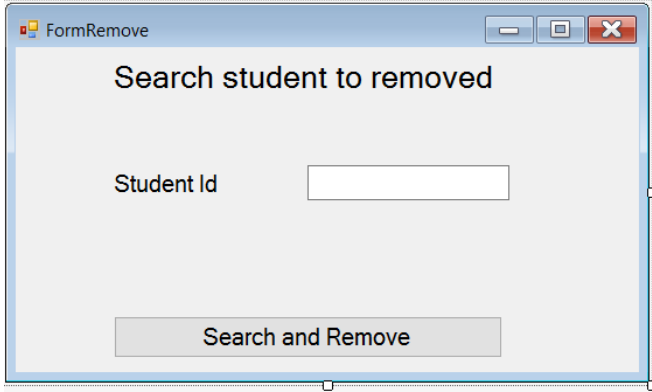
Source Code: Main.cs

Explanation



```
var cmd = new OleDbCommand("UPDATE Student Set Firstname =  
@b, Lastname = @c, Email = @d, Address = @e, Subject = @f,  
intCAOdata = @g, doubleDegreegrade = @h WHERE Id = @i");  
cmd.Connection = connection;  
cmd.Parameters.AddRange(new[] {  
    new OleDbParameter("@b", fnstring),  
    new OleDbParameter("@c", lnstring),  
    new OleDbParameter("@d", emailstring),  
    new OleDbParameter("@e", addressstring),  
    new OleDbParameter("@f", subjectstring),  
    new OleDbParameter("@g", caostring),  
    new OleDbParameter("@h", degreestring),  
    new OleDbParameter("@i", lookupId)
```

When the id is found the user can update whatever info they want and program uses a parameterized update statement to do so

Source Code: Main.cs	Explanation
<div data-bbox="204 398 858 788" data-label="Image"></div> <pre data-bbox="204 824 1129 1064">var cmd = new OleDbCommand("Delete * From Student WHERE Id = @i"); cmd.Connection = connection; cmd.Parameters.AddRange(new[] { new OleDbParameter("@i", removeId) }); cmd.ExecuteNonQuery(); MessageBox.Show("Student deleted successfully."); this.Hide();</pre>	<p data-bbox="1173 331 1380 721">On pressing the remove student button in the main menu, the user is given a form allowing them to search for and delete a student entry based on the ID they give.</p>