



Final Project

ESOFT METRO CAMPUS

Shaping Lives, Creating Futures.

Final Project – Assessment Sheet	
Programme	Diploma in Information Technology
Assessor/Lecturer	Mr. Pasan weththasinghe
Unit	Final Project
Project Title	Skills International School
Student's Name	D.W.A.N.N. Wadisingha
Student's Registration No	00128157
Submission Date	19/12/2022
Marks Achieved (%)	
Assessor/Lecturer Feedback	

Important Points.

- Carefully check the hand in date and the instructions given with the project. **Late submissions will not be accepted.**
- Ensure that sufficient time is spent to complete the assignment by the due date.
- Do not wait till the last minute to get feedback on the project. Such excuses will not be accepted for late submissions.
- You must be responsible for efficient management of your time.
- If you are unable to hand in your project on time and have valid reasons such as illness, you may apply (in writing) for an extension.
- Failure to achieve at least 40.00% marks will result in a Repeat grade.
- Non-submission of work without valid reasons will lead to an automatic Repeat. You will then be asked to complete an alternative project.
- If you use other people's work or ideas in your assignment, it must be properly referenced, using the HARVARD referencing system, in your text or any bibliography. Otherwise, you'll be found guilty of committing plagiarism.
- If you are caught plagiarising, your grade will be reduced to a Repeat or at worst, you could be excluded from the course.

Project Guidelines

- You need to **create a software** solution for Skills International School using **visual studio** as your final project.
- Use **C# Programming language** for your project
- Follow the structure given in this document. Design given forms and apply value additions for it according to your requirement.
- Software **should** contain **minimum two forms**. Additional forms and features will get additional marks. (Login form and one Registration form is compulsory)
- Please follow the given instructions before you start the project.

Instructions

- Design the software system according to the given instructions.
- Use appropriate fonts, colors, and images for Frontend (GUI design) of your software.
- Proposed Software system **must have a database and properly connected** to the system. (Database should be designed using SQL Server Management Studio)
- Database must contain minimum **one Table** (Table column structure provided in Figure X).
- **Test the software** and submit the project on or before the given deadline.
- Create a **Microsoft Word Document** including all the screenshots of forms in your system and code snippets of each form.
- The entire project brief (These nine pages) should be attached as the first section of your project documentation.

Word Processing Rules

- The font size should be 12 point and should be in the style of Time New Roman.
- Set line spacing to 1.15. Justify all paragraphs.
- Ensure that all headings are consistent in terms of size and font style.
- Use the spell check and grammar check function of the word processing application to review the use of language on your assignment.

Proposed Project



The image shows a Windows-style login window titled "Login - Skills International". At the top center is the Skills International logo, which consists of the word "SKILLS" in a bold, sans-serif font, with a colorful circular graphic to its right. Below the logo, the text "Skills International" is displayed in a large, bold, black font. Underneath this, there is a section titled "Login" in a smaller font. This section contains two labels, "Username" and "Password", each followed by a corresponding text input box. Below the input boxes are two buttons: "Clear" and "Login". At the bottom left of the window, outside the main login section, is an "Exit" button. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Figure I – Login Form

- Above form consist of;
 - One Picture Box
 - One Group Box
 - Three Labels
 - Skills International
 - Username
 - Password
 - Two Text Boxes
 - Username
 - Password
 - Three Buttons
 - Clear
 - Login
 - Exit

Design the above form using these tools and change respective properties using suitable values.

Student Registration - Skills International

Logout

Skills International

Student Registration

Reg No

Basic Details

First Name

Last Name

Date of Birth

Gender ☐ Male ☐ Female

Contact Details

Address

Email

Mobile Phone Home Phone

Parent Details

Parent Name

NIC

Contact Number

Register Update Clear Delete

Exit

Figure II – Registration Form

- Above form consist of;
 - Four Group Boxes
 - Student Registration
 - Basic Details
 - Contact Details
 - Parent Details
 - Thirteen Labels
 - Skills International
 - Reg No
 - First Name
 - Last Name
 - Date of Birth
 - Gender
 - Address
 - Email
 - Mobile Phone
 - Home Phone
 - Parent Name
 - NIC
 - Contact No
 - Nine Text Boxes
 - First Name
 - Last Name
 - Address (Multiline)
 - Email
 - Mobile Phone
 - Home Phone
 - Parent Name
 - NIC
 - Contact No
 - Four Buttons
 - Register
 - Update
 - Clear
 - Delete
 - Two LinkLabels
 - Logout
 - Exit
 - Two Radio Buttons
 - Male
 - Female
 - One DateTimePicker
 - Date of Birth
 - One ComboBox
 - Reg No

Design the above form using these tools and change respective properties using suitable values.

Project Instructions

Designing

- Design the project using Microsoft Visual Studio. **Designing above two forms is a must**, you can add more forms based on your imagination and those will be considered when marking.
- **Change the properties** of each object with suitable values.

Functional Operations

- When user runs the project, Login form needs to be appeared.

Form I – Login Form (Figure I)

- There is **one user** in the system. User name is “**Admin**” & Password is “**Skills@123**”. When user click on Login () button after applying “Admin” as the username and “Skills@123” as the Password, the system needs to **hide the login form** and **open the second form** which is Registration form (Figure II). If the login credentials are wrong (Either Username, either Password or both), system needs to generate an error message (Figure III).

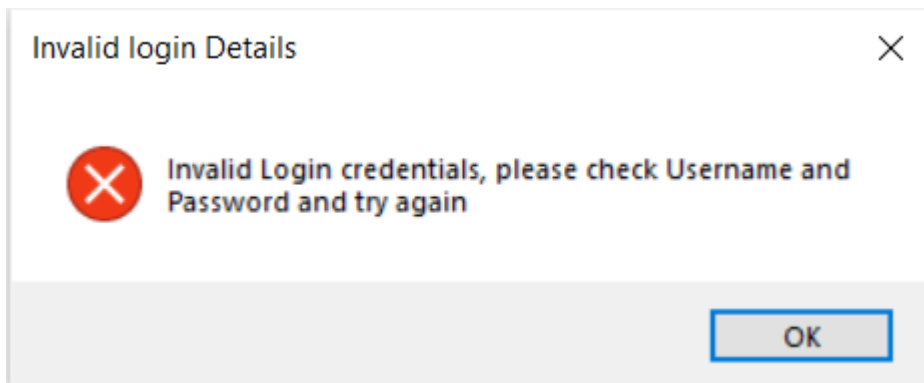


Figure III – Error Message

- You can **increase number of users** for the system. If so, there needs to be a table in the database to track those usernames and passwords as shown in figure IX.
- When user clicks on Clear () button, texts in both username textbox and password textbox needs to be cleared and the username textbox needs to be **focused**.
- When you click on the Exit Button () it needs to appear the following message box.

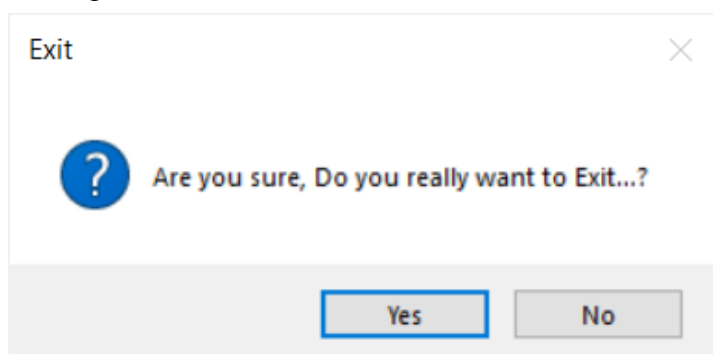


Figure IV – Exit Message

- If you click on Yes button, the entire system needs to be closed and if you click on No, the system stays without a change, but the dialogue box needs to be closed.

Form II – Registration Form (Figure II)

- The Form II (Figure II) is designated form for Student Registrations.
- Create the Database using Microsoft SQL Server Management Studio. Name the database as **Student**.
- Create the **Registration Table**.
- Connect the created Student Database with your C# project.
- In student Registration form, a registration can be done by clicking on Register Button (). Once the user clicks on **Register Button** after providing all the details in related fields in the registration form, the details need to be added in the “**Registration**” Table (Figure X) of “**Student**” Database and needs to appear following message box as well.

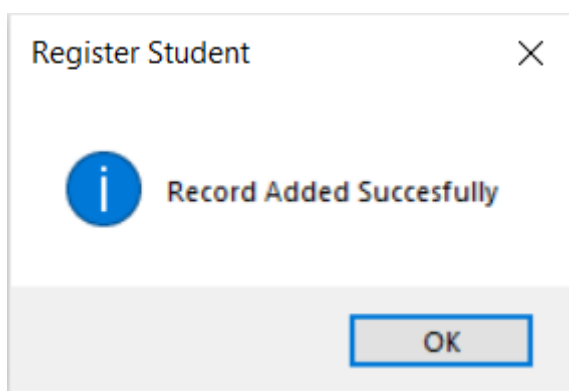


Figure V – Register Message Box

- Give the facility to the user to update records using **Update Button** ().
- Once the update executed, show the following message box.

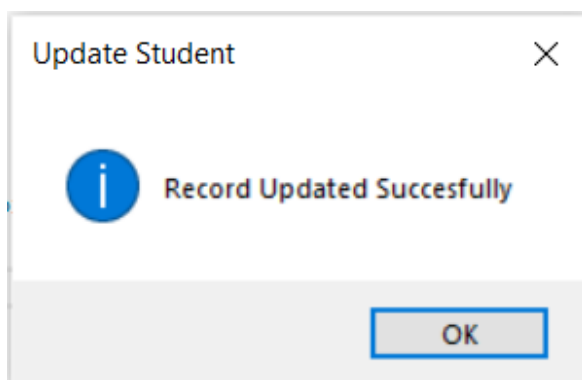


Figure VI – Update Message Box

- Give the facility for the user to Delete a selected record from the Database Table using the **Delete Button** (). Before Execution of Delete function, need to get users confirmation to delete. For that, use following message box or similar one.

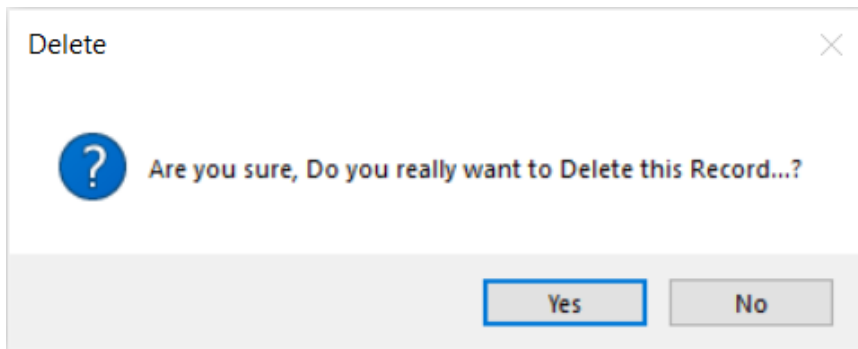


Figure VII – Delete Message Box

- Once the user clicks on Yes from the appearing message box, the details needs to be deleted from the system database based on the Registration Number of the record. After the execution of the Delete, following message box needs to be appeared.

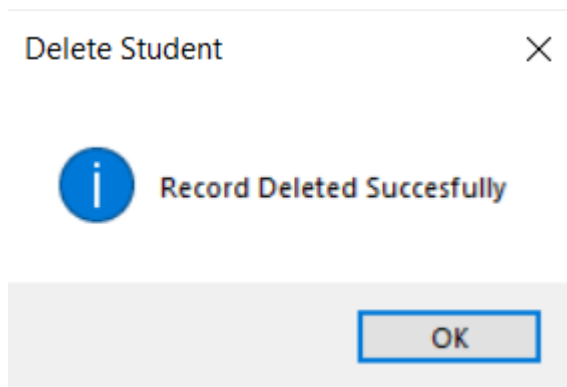


Figure VIII – Delete Confirmation Message Box

- When the user clicks on the **Reg No dropdown list** (), the user can see the registration numbers (Previously registered) appearing in the drop down list. When the user selects a registration number, it needs to work as a **Search** function when user changes the text of this and need to fill all other fields in the form by focusing the Reg No from the Database Table.
- When you click on the Exit Link Label ([Exit](#)) it needs to appear the Exit message box (Figure IV).
- When user clicks on Logout Link Label ([Logout](#)), it needs to logout from the system, disappear the Registration form and appear the login form (Figure I).

Database Information

Set Database name as “**Student**” and Registration Table (Figure X) needs to be there inside. When designing the **Registration** Table in the Database, apply **regNo** column as the **primary key** for the table.

	username	password
pk	NULL	NULL

Figure IX – Database Table - **Logins**

	regNo	firstName	lastName	dateOfBirth	gender	address	email	mobilePho...	homePhone	parentName	nic	contactNo
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure X – Database Table - **Registration**

Table Structure

Use the following table to identify data types for each column when you create the “**Registration**” table.

Field Name	Datatype
regNo	Integer
firstName	varchar(50)
lastName	varchar(50)
dateOfBirth	dateTime
gender	varchar(50)
address	varchar(50)
email	varchar(50)
mobilePhone	Integer
homePhone	Integer
parentName	varchar(50)
nic	varchar(50)
contactNo	Integer



SKILLS INTERNATIONAL SCHOOL

THIS SOFTWARE HELPS TO IDENTIFY REGISTRATION DETAILS OF APPLICANTS EASILY. The main goal of this system is to maintain functions effectively and in an accurate manner. This system reduces the manual work as everything is computerized. In here we have mentioned to apply details of individuals using language of C# and we have used SQL server management studio and visual studio. This software system is very useful in the identification of individual details of students easily and everything we mentioned are stored in the database.

Table of Contents

Form 3 - Splash screen (Figure 1)	13
Form 3 (Figure 1) – Property Settings & Coding	13
Form 3	13
Label 1	13
Label 2	14
Panel 1	14
Panel 2	14
Timer 1	14
Timer 2	15
Timer 3	15
Form 3 (Figure 1) – Coding	16
Form 1 – Login Form (Figure 5)	18
Form 1 (Figure 5) – Property Settings & Coding	20
Form 1	20
Picture Box 1	20
Group Box 1	20
Label 1	20
Label 2	21
Label 3	21
Check Box 1	21
Text Box 1	22
Text box 2	22
Button 1	22
Button 2	22
Button 3	24
Form 1 (Figure 5) – Coding	25
Form 2 – Registration Form (Figure 16)	28
Form 2 (Figure 16) – Property Settings	31
Group Boxes Property Settings	31
Labels Property Settings	32
Text Boxes Property Settings	33
Buttons Property Settings	33

Link Labels Property Settings.....	34
Radio Buttons Property Settings.....	34
DateTimePicker 1 Property Settings.....	34
ComboBox 1 Property Settings	34
Form 2 (Figure 16) – Coding	35
ComboBox Code.....	35
Register Button Code	36
Update Button Code.....	37
Clear Button Code.....	38
Delete Button Code.....	39
Link Label Logout Code	40
Link Label Exit Code.....	41
My Final Project Link	48
References	49
THANK YOU!!	50

Form 3 - Splash screen (Figure 1)



Figure 1 – Splash Screen Design

When the user runs the project, first this loading window is shown and after the loading is complete, the splash screen is hidden and the first form which is Login form (Figure 5) is opened.

Form 3 (Figure 1) – Property Settings & Coding

Form 3

- Name - frmsplash
- Size - 815, 489

Label 1

- Name - lblsi
- Text - Skills International
- Back color - Transparent
- Font - Stencil, 13.8pt, style=Bold
- Fore Color - Navy
- Size - 289, 27

Label 2

- Name - lblloading
- Text - Loading.....
- Back color - Transparent
- Font - Century Gothic, 15pt, style=Bold, Italic
- Fore Color – Black
- Size - 155, 32

Panel 1

- Name - panel1
- Back color - Crimson
- Size - 443, 19

Panel 2

- Name - panel2
- Back Color - White
- Size - 12, 19

Timer 1

- Name - timer1
- Interval - 25
- Code -

```
private void timer1_Tick(object sender, EventArgs e)
{
    lblloading.Top -= 1;
    if(lblloading.Top<=300)
    {
        timer1.Stop();
        timer2.Start();
    }
}
```

Timer 2

- Name - timer2
- Interval - 25
- Code -

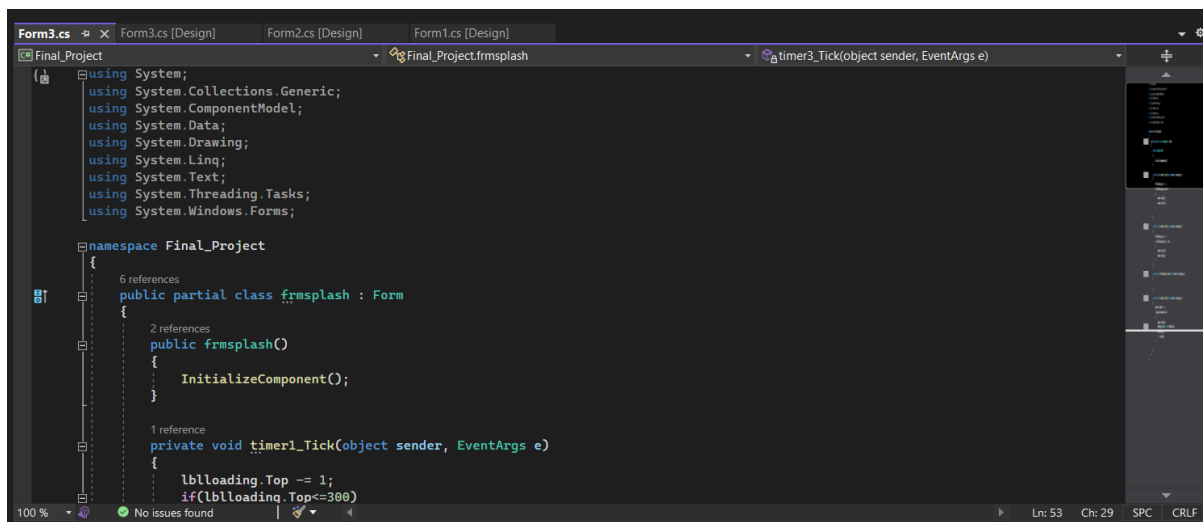
```
private void timer2_Tick(object sender, EventArgs e)
{
    lblloading.Top += 1;
    if (lblloading.Top >= 360)
    {
        timer1.Start();
        timer2.Stop();
    }
}
```

Timer 3

- Name - timer3
- Interval - 50
- Code -

```
private void timer3_Tick(object sender, EventArgs e)
{
    panel2.Width += 3;
    if(panel2.Width>=443)
    {
        timer3.Stop();
        frmlogin form = new frmlogin();
        form.Show();
        this.Hide();
    }
}
```


Form 3 (Figure 1) – Coding



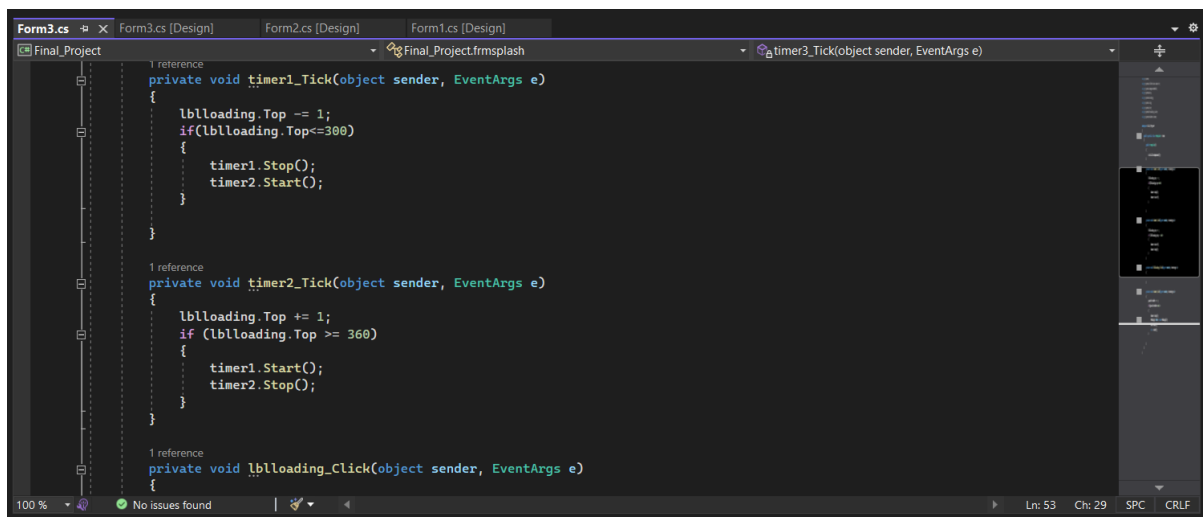
```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Final_Project
{
    public partial class frmSplash : Form
    {
        public frmSplash()
        {
            InitializeComponent();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            lblloading.Top -= 1;
            if(lblloading.Top<=300)
        }
    }
}
    
```

Figure 2 – Splash Screen Coding 1



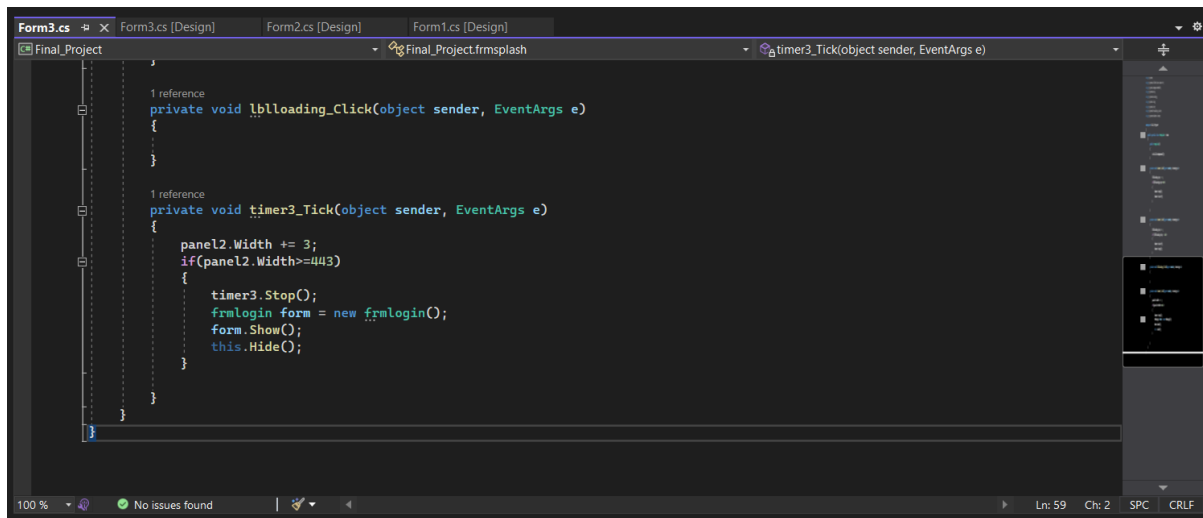
```

private void timer1_Tick(object sender, EventArgs e)
{
    lblloading.Top -= 1;
    if(lblloading.Top<=300)
    {
        timer1.Stop();
        timer2.Start();
    }
}

private void timer2_Tick(object sender, EventArgs e)
{
    lblloading.Top += 1;
    if (lblloading.Top >= 360)
    {
        timer1.Start();
        timer2.Stop();
    }
}

private void lblloading_Click(object sender, EventArgs e)
{
}
    
```

Figure 3 – Splash Screen Coding 2



```

Form3.cs
Form3 [Design]
Form2.cs [Design]
Form1.cs [Design]
Final_Project
Final_Project.frm splash
timer3.Tick(object sender, EventArgs e)

1 reference
private void lblloading_Click(object sender, EventArgs e)
{
}

1 reference
private void timer3_Tick(object sender, EventArgs e)
{
    panel2.Width += 3;
    if(panel2.Width>=443)
    {
        timer3.Stop();
        frmlogin form = new frmlogin();
        form.Show();
        this.Hide();
    }
}
    
```

100 % No issues found Ln: 59 Ch: 2 SPC CRLF

Figure 4 – Splash Screen Coding 3

Form 1 – Login Form (*Figure 5*)

Figure 5 – Login Form Design

- The password is **Skills@123** and the username is **Admin**.
- When user clicks on **Login button** after applying “Admin” as the username and “Skills@123” as the Password, the system hides the login form and opens the second form which is Registration form (*Figure 16*).
- If the login credentials are wrong (Either Username, either Password or both), system generates an error message (*Figure 6*).

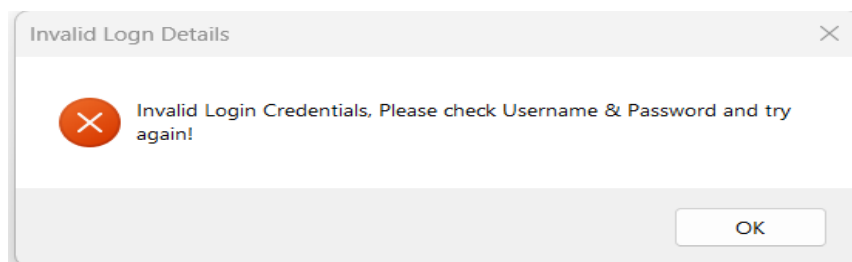


Figure 6 – Error Message

- The system allows three login attempts and if some users are unable to enter the correct username and password in those three login attempts, system generates error message (*Figure 7*) and hides the login form and opens the third form which is Splash screen (*Figure 1*) and after loading that form, the system displays Login form (*Figure 5*) again.

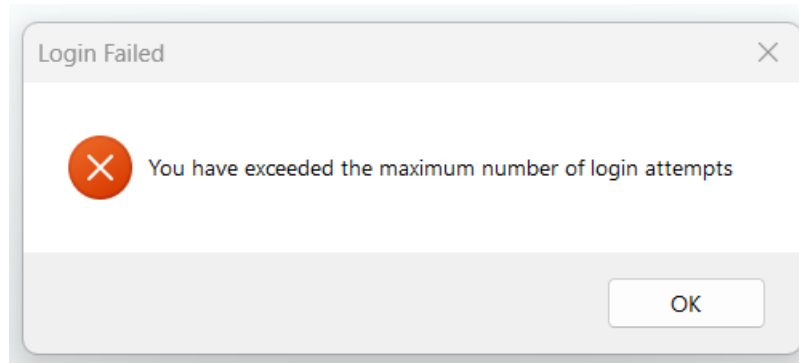


Figure 7 – Error Message

- The system also has the ability to show and hide the password in the password text box. When the user clicks on the show password option(☐ Show Password), the password will be displayed in the password text box and click on the option again to hide the password.
- There is a database created using Microsoft SQL Server Management Studio.
- The database is named “Student”, and has a table named “Login”.
- There is proper connection between “Student” database and C# project.
- If the developer wants to increase the number of users for the system, there is a “Login” table in the database to track those usernames and passwords as shown in *Figure 8*.

	username	password
▶	Admin	Skills@123
	abc	123
*	NULL	NULL

Figure 8 – Database Table – Login

- When the user clicks on the **Clear button**, the text in both the Username textbox and the Password textbox is cleared and the Username textbox gets focus.
- When the user clicks on the **Exit button** the following message box (*Figure 9*) will appear and if the user clicks on its Yes button, then the entire system is closes and if the user clicks on No, the system remains unchanged but closes the dialogue box.

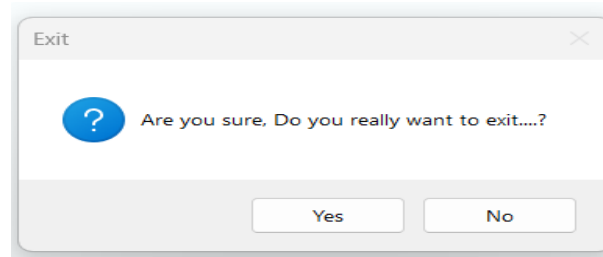


Figure 9 – Exit Message

Form 1 (Figure 5) – Property Settings & Coding

Form 1

- Name - frmlogin
- Text - Login-Skills International
- Back color - White
- Size - 741, 699

Picture Box 1

- Name - picbox1
- Size - 262, 173

Group Box 1

- Name - grpboxlogin
- Text - Login
- Back color - Snow
- Font - Segoe UI, 9pt
- Size - 494, 320

Label 1

- Name - lblskillsinternational
- Text - Skills International
- Font - Stencil, 36pt
- Fore Color - Navy
- Size - 706, 71

Label 2

- Name - lblusername
- Text - Username
- Back Color - Snow
- Font - Book Antiqua, 16.2pt, style=Bold
- Fore Color - Black
- Size - 143, 32

Label 3

- Name - lblpassword
- Text - Password
- Back Color - Snow
- Font - Book Antiqua, 16.2pt, style=Bold
- Fore Color - Black
- Size - 136, 32

Check Box 1

- Name - checkboxshowpw
- Text - Show Password
- Back Color - Snow
- Font - Segoe UI Semibold, 10.2pt, style=Bold
- Fore Color – Black
- Size - 151, 27
- Code -

```
private void checkboxshowpw_CheckedChanged(object sender,
EventArgs e)
{
    if (checkboxshowpw.Checked)
    {
        txtpassword.UseSystemPasswordChar = false;
    }
    else
    {
        txtpassword.UseSystemPasswordChar = true;
    }
}
```

Text Box 1

- Name - txtusername
- Back color - Window
- Font - Segoe UI Semibold, 10.8pt, style=Bold
- Fore Color - Window Frame
- Size - 294, 31

Text box 2

- Name - txtpassword
- Back color - Window
- Font - Segoe UI Semibold, 10.8pt, style=Bold
- Fore Color - Window Frame
- Size - 294, 31

Button 1

- Name - btnclr
- Text - Clear
- Back Color - ButtonShadow
- Font - Segoe UI Variable Text Semibold, 12pt, style=Bold
- Fore Color - Black
- Size - 94, 39
- Code -

```
private void btnclr_Click(object sender, EventArgs e)
{
    txtpassword.Clear();
    txtusername.Clear();
    txtusername.Focus();
}
```

Button 2

- Name - btnlogin
- Text - Login
- Back Color - ButtonShadow
- Font - Segoe UI Variable Text Semibold, 12pt, style=Bold

- Fore Color - Black
- Size - 94, 39
- Code -

```
private void btnlogin_Click(object sender, EventArgs e)
{
    if (con.State == ConnectionState.Closed) con.Open();
    string username = txtusername.Text;
    string password = txtpassword.Text;
    string query_select = "SELECT * FROM login WHERE username='" +
username + "' AND password='" + password + "'";
    SqlCommand cmdnd = new SqlCommand(query_select, con);
    using (var row = cmdnd.ExecuteReader())
    {
        if (row.HasRows)
        {
            this.Hide();
            Frmstudentreg obj = new Frmstudentreg();
            obj.Show();
        }
        else
        {
            logAttempt += 1;

            if (logAttempt == 3)
            {
                MessageBox.Show("You have exceeded the maximum number of login
attempts", "Login Failed", MessageBoxButtons.OK, MessageBoxIcon.Stop);
                txtusername.Enabled = false;
                txtpassword.Enabled = false;
                btnlogin.Enabled = false;

                this.Hide();
                frmsplash obj = new frmsplash();
                obj.Show();
            }
            else

                MessageBox.Show("Invalid Login Credentials, Please check Username &
Password and try again!", "Invalid Logn Details", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                Reset();
                txtusername.Clear();
            }
        }
    }
}
```



```
txtpassword.Clear();  
txtusername.Focus();
```

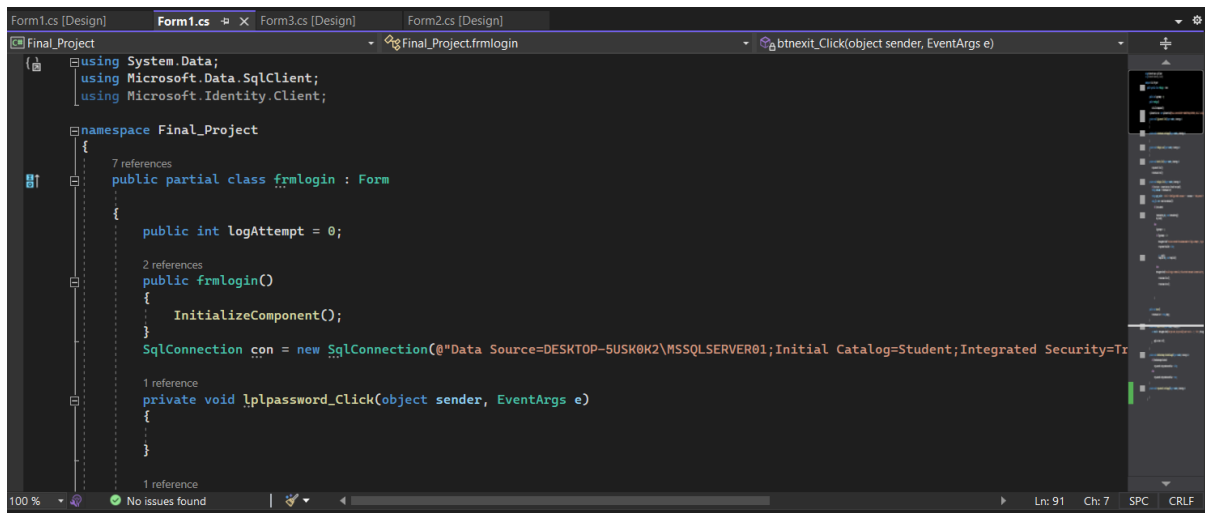
```
    }  
  }  
}
```

Button 3

- Name - btnexit
- Text - Exit
- Back Color - ButtonShadow
- Font - Segoe UI Variable Text Semibold, 12pt, style=Bold
- Fore Color - Black
- Size - 94, 39
- Code -

```
private void btnexit_Click(object sender, EventArgs e)  
{  
    var result = MessageBox.Show("Are you sure, Do you really want to exit....?",  
"Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  
    if (result == DialogResult.Yes)  
    {  
        Application.Exit();  
    }  
}
```

Form 1 (Figure 5) – Coding



```

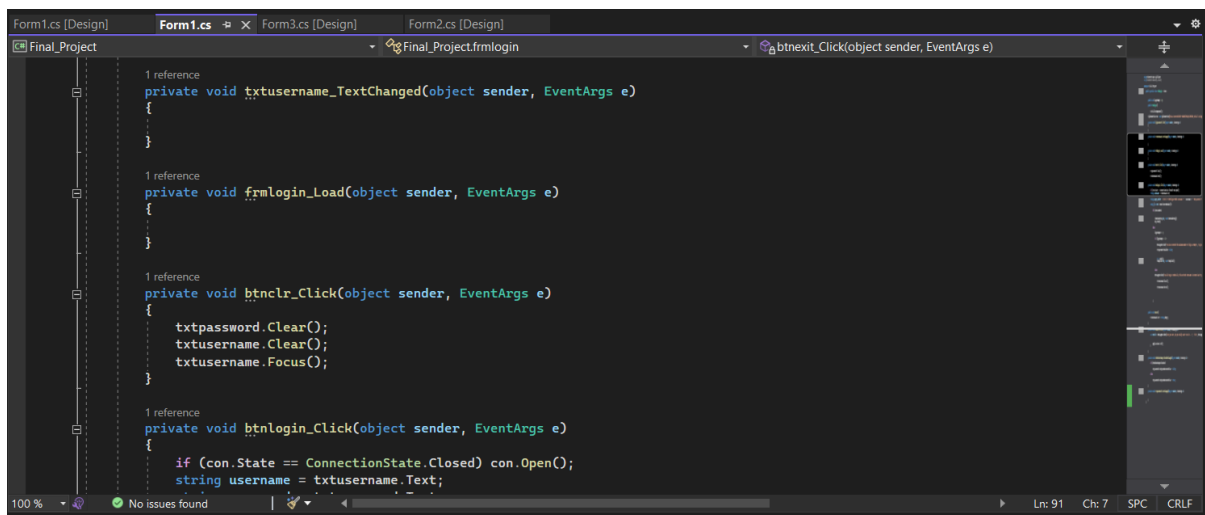
using System.Data;
using Microsoft.Data.SqlClient;
using Microsoft.Identity.Client;

namespace Final_Project
{
    7 references
    public partial class frmlogin : Form
    {
        public int logAttempt = 0;

        2 references
        public frmlogin()
        {
            InitializeComponent();
        }

        1 reference
        private void btnlogin_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-SUSK0K2\MSSQLSERVER01;Initial Catalog=Student;Integrated Security=True;User=sa;Password=1234567890;");
        }
    }
}
    
```

Figure 10 – Login Form Coding 1



```

1 reference
private void txtusername_TextChanged(object sender, EventArgs e)
{
}

1 reference
private void frmlogin_Load(object sender, EventArgs e)
{
}

1 reference
private void btnclr_Click(object sender, EventArgs e)
{
    txtpassword.Clear();
    txtusername.Clear();
    txtusername.Focus();
}

1 reference
private void btnlogin_Click(object sender, EventArgs e)
{
    if (con.State == ConnectionState.Closed) con.Open();
    string username = txtusername.Text;
}
    
```

Figure 11 – Login Form Coding 2

```

string username = txtusername.Text;
string password = txtpassword.Text;
string query_select = "SELECT * FROM login WHERE username ='" + username + "' AND password ='" + password + "'";
SqlCommand cmdnd = new SqlCommand(query_select, con);
using (var row = cmdnd.ExecuteReader())
{
    if (row.HasRows)
    {
        this.Hide();
        Frmstudentreg obj = new Frmstudentreg();
        obj.Show();
    }
    else
    {
        logAttempt += 1;

        if (logAttempt == 3)
        {
            MessageBox.Show("You have exceeded the maximum number of login attempts", "Login Failed", MessageBoxButtons.OK,
            txtusername.Enabled = false;
            txtpassword.Enabled = false;
            btnlogin.Enabled = false;

            this.Hide();
            frmSplash obj = new frmSplash();

```

Figure 12 - Login Form Coding 3

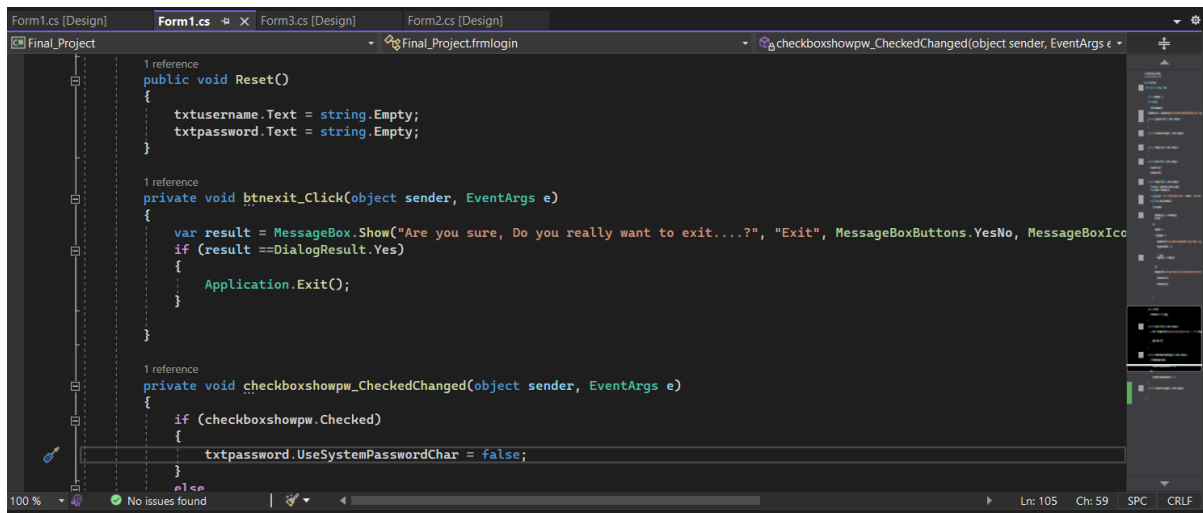
```

        frmSplash obj = new frmSplash();
        obj.Show();
    }
    else
    {
        MessageBox.Show("Invalid Login Credentials, Please check Username & Password and try again!", "Invalid Login Details"
        Reset();
        txtusername.Clear();
        txtpassword.Clear();
        txtusername.Focus();
    }
}

1 reference
public void Reset()
{
    txtusername.Text = string.Empty;
    txtpassword.Text = string.Empty;
}

```

Figure 13 – Login Form Coding 4



```

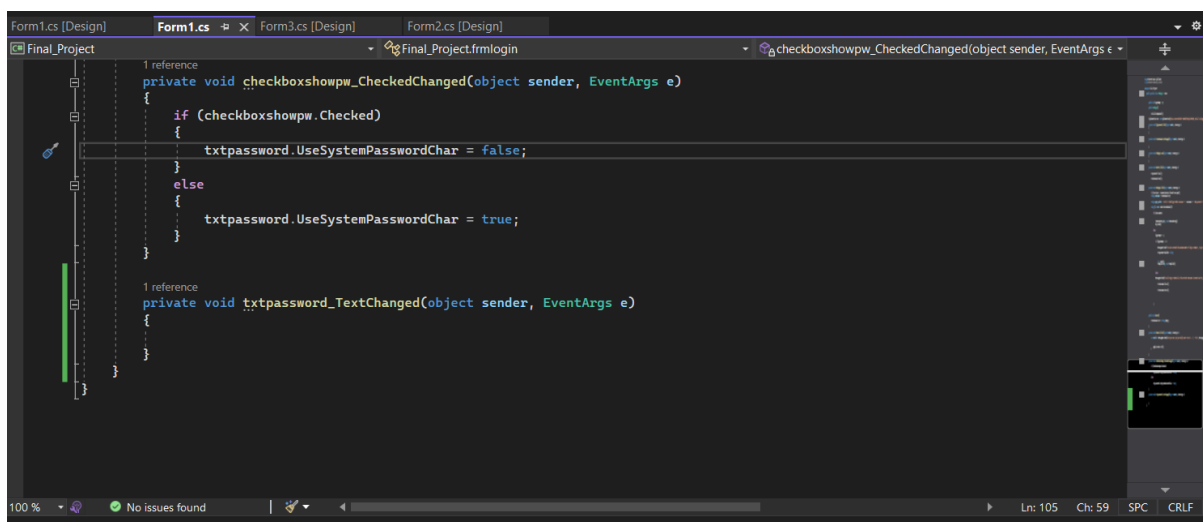
1 reference
public void Reset()
{
    txtusername.Text = string.Empty;
    txtpassword.Text = string.Empty;
}

1 reference
private void btnexit_Click(object sender, EventArgs e)
{
    var result = MessageBox.Show("Are you sure, Do you really want to exit....?", "Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
    if (result == DialogResult.Yes)
    {
        Application.Exit();
    }
}

1 reference
private void checkboxshowpw_CheckedChanged(object sender, EventArgs e)
{
    if (checkboxshowpw.Checked)
    {
        txtpassword.UseSystemPasswordChar = false;
    }
    else
    {
        txtpassword.UseSystemPasswordChar = true;
    }
}

```

Figure 14 - Login Form Coding 5



```

1 reference
private void checkboxshowpw_CheckedChanged(object sender, EventArgs e)
{
    if (checkboxshowpw.Checked)
    {
        txtpassword.UseSystemPasswordChar = false;
    }
    else
    {
        txtpassword.UseSystemPasswordChar = true;
    }
}

1 reference
private void txtpassword_TextChanged(object sender, EventArgs e)
{
}

```

Figure 15 - Login Form Coding 6

Form 2 – Registration Form (Figure 16)

Student Registration - Skills International

[Logout](#) **SKILLS INTERNATIONAL**

Student Registration

Reg No

Basic Details

First Name

Last Name

Date of Birth

Gender ☐ Male ☐ Female

Contact Details

Address

Email

Mobile Phone **Home Phone**

Parent Details

Parent Name

NIC

Contact Number

[Exit](#)

Figure 16 – Registration Form Design

- The Form 2 (Figure 16) is designated form for Student Registrations.
- There is another table named “Registration” in the database named “Student”.
- In student Registration form, a registration can be done by clicking on **Register Button**.
- Once the user clicks the Register Button after providing all the details in the relevant field of the registration form, the provided details are added to the “Registration” table (Figure 17) of “Student” database and the following message box (Figure 18) is also displayed.

regNo	firstName	lastName	dateOfBirth	gender	address	email	mobilePho...	homePhone	parentName	nic	contactNo
1	Ishan	Poorna	2001-12-09 ...	Male	yatagala	ishanpoorn...	776218353	713114840	Gamage	19584552364	774404232
2	Nawoda	Nethmini	2001-07-03 ...	Female	"Wimukthi",...	nawodanet...	757253414	912227110	D.W.A.D.Nis...	662602019V	770079904
3	Sahan	Wimukthi	1996-10-20 ...	Male	"Wimukthi",...	wimukthi@...	777255836	912614785	H.M.A.S.Her...	75473698V	701258796
4	Dulmi	Apeksha	2001-09-04 ...	Female	114/AMora...	apeksha@g...	751478963	912589631	Wasanthi Gi...	728456931V	705896472
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 17 - Database Table – Registration

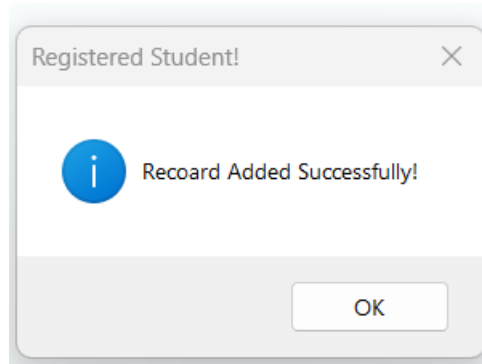


Figure 18 - Register Message Box

- User can update records using **Update Button**.
- Once the update executed, the following message box (Figure 19) will be displayed.

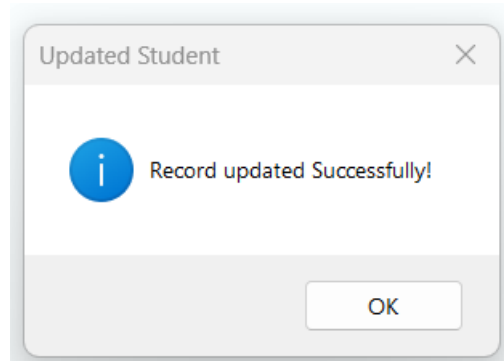


Figure 19 - Update Message Box

- By using **Delete Button**, enables the user to delete the selected record from the database table. Before executing the delete function, users must provide confirmation to delete. The following message box (Figure 20) is used for that.
- After the user clicks on **Yes** from the message box that appears, details based on the Registration Number of the record, deleted from the system database. After executing Delete, the following message box (Figure 21) will appear.
- if the user clicks on **No**, the system remains unchanged but closes the dialogue box.

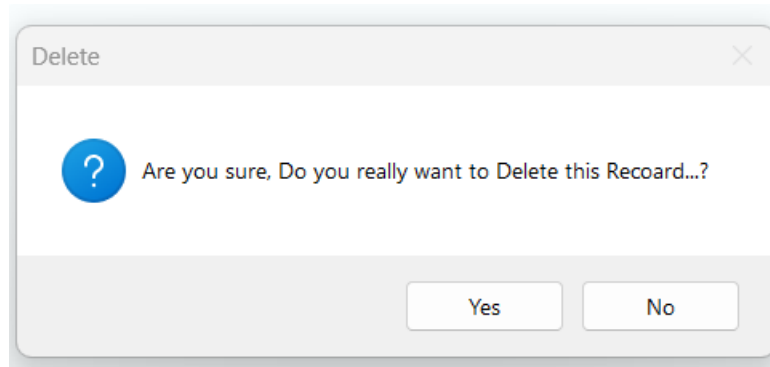


Figure 20 - Delete Confirmation Message Box

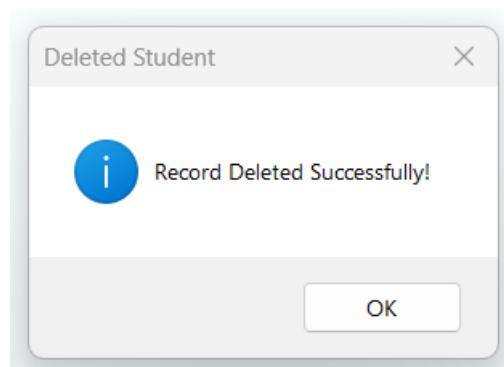


Figure 21 - Delete Message Box

- When the user clicks on the **Reg No dropdown list**, the user can see the registration numbers (Previously registered) appearing in the dropdown list. When the user selects a registration number, it works as a Search function when user changes the text of this and fill all other fields in the form by focusing the Reg No from the Database Table.
- When the user clicks on the **Exit Link label**, it displays the Exit message box (Figure 22).

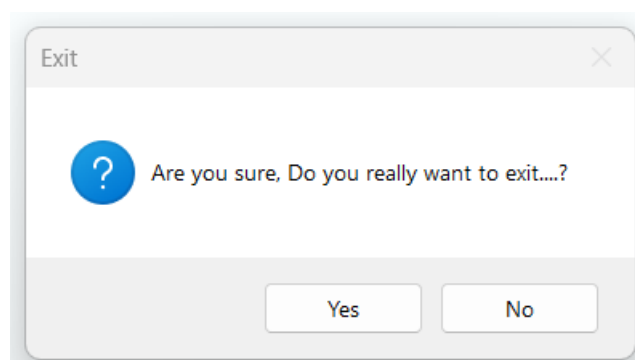


Figure 22 – Exit Message Box

- If the user clicks on its Yes button, then the entire system is closes and if the user clicks on No, the system remains unchanged but closes the dialogue box.

- When the user clicks on the **Logout Link Label**, it shows the logout message box (Figure 23) and user clicks on its Yes button then leaves the system. The registration form (Figure 16) disappears and the login form (Figure 5) appears.

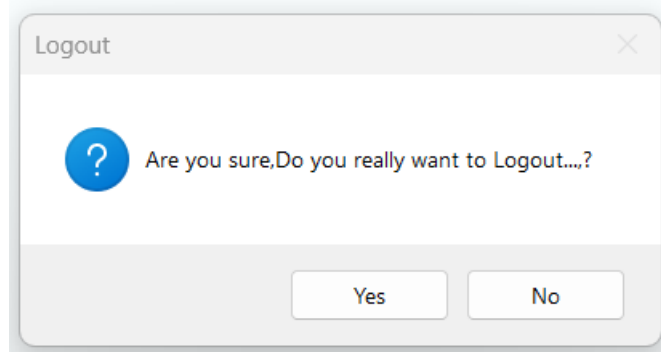


Figure 23 – Logout Message Box

Form 2 (Figure 16) – Property Settings

Group Boxes Property Settings

Group Box No	Name	Text	Back Color	Font	Fore Color	Size
1	grpboxstudentreg	Student Registration	Control	Segoe UI Semi bold, 9pt, style=Bold	Control Text	840, 733
2	grpboxbasicdetails	Basic Details	Control	Segoe UI Semi bold, 9pt, style=Bold	Control Text	790, 185
3	grpboxcondetails	Contac Details	Control	Segoe UI Semi bold, 9pt, style=Bold	Control Text	790, 204
4	grpboxparentdetails	Parent Details	Control	Segoe UI Semi bold, 9pt, style=Bold	Control Text	790, 160

Labels Property Settings

Label No	Name	Text	Back Color	Font	Fore Color	Size
1	lblskillsinternational2	Skills International	Control	Stencil, 15pt	Navy	293, 30
2	lblregno	Reg No	Control	Bookman Old Style, 10.8pt, style=Bold	Control Text	74, 21
3	lblfirstname	First Name	Control	Bahnschrift SemiBold, 10.8pt, style=Bold	Control Text	101, 22
4	lbllastname	Last Name	Control	Bahnschrift SemiBold, 10.8pt, style=Bold	Control Text	98, 22
5	lbldateofbirth	Date of Birth	Control	Bahnschrift SemiBold, 10.8pt, style=Bold	Control Text	115, 22
6	lblgender	Gender	Control	Bahnschrift SemiBold, 10.8pt, style=Bold	Control Text	70, 22
7	lbladdress	Address	Control	Bahnschrift SemiBold, 10.8pt, style=Bold	Control Text	78, 22
8	lblemail	Email	Control	Bahnschrift SemiBold, 10.8pt, style=Bold	Control Text	56, 22
9	lblmobilephone	Mobile Phone	Control	Bahnschrift SemiBold, 10.8pt, style=Bold	Control Text	120, 22
10	lblhomephone	Home Phone	Control	Bahnschrift SemiBold, 10.8pt, style=Bold	Control Text	113, 22
11	lblparentname	Parent Name	Control	Bahnschrift SemiBold, 10.8pt, style=Bold	Control Text	118, 22
12	lblnic	NIC	Control	Bahnschrift SemiBold, 10.8pt, style=Bold	Control Text	39, 22
13	lblcontactnum	Contact Number	Control	Bahnschrift SemiBold, 10.8pt, style=Bold	Control Text	143, 22

Text Boxes Property Settings

Text Box No	Name	Back Color	Font	Fore Color	Size
1	txtfirstname	Window	Century, 12pt	Window Text	360, 32
2	txtlastname	Window	Century, 12pt	Window Text	360, 32
3	txtaddress	Window	Century, 12pt	Window Text	360, 89
4	txtemail	Window	Century, 12pt	Window Text	360, 32
5	txtmobilephone	Window	Century, 12pt	Window Text	220, 32
6	txthomephone	Window	Century, 12pt	Window Text	220, 32
7	txtparentname	Window	Century, 12pt	Window Text	360, 32
8	txtnic	Window	Century, 12pt	Window Text	293, 32
9	txtcontacnum	Window	Century, 12pt	Window Text	293, 32

Buttons Property Settings

Button No	Name	Text	Back color	Font	Fore Color	Size
1	btnregister	Register	Button Shadow	Franklin Gothic Book, 12pt, style=Bold	Control Text	111, 33
2	btnupdate	Update	Button Shadow	Franklin Gothic Book, 12pt, style=Bold	Control Text	111, 33
3	btnclear	Clear	Button Shadow	Franklin Gothic Book, 12pt, style=Bold	Control Text	111, 33
4	btndelete	Delete	Button Shadow	Franklin Gothic Book, 12pt, style=Bold	Control Text	111, 33

Link Labels Property Settings

Link Label No	Name	Text	Back Color	Font	Fore Color	Size
1	linklblexit	Exit	GradientInactive Caption	Times New Roman, 12pt, style=Bold	Control Text	45, 23
2	linklbllogout	Logout	GradientInactive Caption	Times New Roman, 12pt, style=Bold	Control Text	70, 23

Radio Buttons Property Settings

Radio Button No	Name	Text	Back Color	Font	Fore Color
1	radiobtnmale	Male	Control	Bahnschrift SemiBold, 12pt, style=Bold	Control Text
2	radiobtnfemale	Female	Control	Bahnschrift SemiBold, 12pt, style=Bold	Control Text

DateTimePicker 1 Property Settings

- Name - dateTimePicker1
- Custom Format - dd/MM/yyyy
- Font - Copperplate Gothic Light

ComboBox 1 Property Settings

- Name – comboBoxregno
- Back Color – Window
- Font - Century, 12pt
- Fore Color – Black
- Size - 220, 31

Form 2 (Figure 16) – Coding

ComboBox Code

```
private void comboboxregno_SelectedIndexChanged(object sender, EventArgs e)
{
    con.Open();
    int x = Convert.ToInt32(comboboxregno.SelectedItem.ToString());
    SqlCommand cmd = new(@"SELECT
        [regNo]
        ,[firstName]
        ,[lastName]
        ,[dateOfBirth]
        ,[gender]
        ,[address]
        ,[email]
        ,[mobilePhone]
        ,[homePhone]
        ,[parentName]
        ,[nic]
        ,[contactNo]
    FROM registration WHERE regNo = '" + x + "'", con);

    SqlDataReader reader;
    reader= cmd.ExecuteReader();
    if (reader.Read())
    {
        comboboxregno.Text = reader["regNo"].ToString();
        txtfirstname.Text = reader["firstName"].ToString();
        txtlastname.Text = reader["lastName"].ToString();
        dateTimePicker1.Format = DateTimePickerFormat.Custom;
        dateTimePicker1.CustomFormat = "dd/MM/yyyy";
        dateTimePicker1.Text= reader["dateOfBirth"].ToString();
        if (reader["gender"].ToString() == "Male")
        {
            radiobtnmale.Checked = true;
            radiobtnfemale.Checked = false;
        }
        else
        {
            radiobtnmale.Checked = false;
            radiobtnfemale.Checked = true;
        }
    }
}
```

```
    }  
    txtaddress.Text = reader["address"].ToString();  
    txtemail.Text = reader["email"].ToString();  
    txtmobilephone.Text = reader["mobilePhone"].ToString();  
    txthomephone.Text = reader["homePhone"].ToString();  
    txtparentname.Text = reader["parentName"].ToString();  
    txtnic.Text = reader["nic"].ToString();  
    txtcontacnum.Text = reader["contactNo"].ToString();  
    }  
    else  
    {  
        MessageBox.Show("No Data");  
    }  
    con.Close();  
}
```

Register Button Code

```
private void btnregister_Click(object sender, EventArgs e)  
{  
    try  
    {  
  
        int regno = int.Parse(comboxregno.Text);  
        string firstName = txtfirstname.Text;  
        string lastName = txtlastname.Text;  
        dateTimePicker1.Format = DateTimePickerFormat.Custom;  
        dateTimePicker1.CustomFormat = "dd/MM/yyyy";  
        string gender;  
  
        if (radiobtnmale.Checked)  
        {  
            gender = "Male";  
        }  
        else  
        {  
            gender = "Female";  
        }  
        string address = txtaddress.Text;  
        string email = txtemail.Text;
```

```

        int mobilePhone = int.Parse(txtmobilephone.Text);
        int homePhone = int.Parse(txthomephone.Text);
        string parentName = txtparentname.Text;
        string nic = txtnic.Text;
        int contactNo = int.Parse(txtcontacnum.Text);
        string query_insert = "insert into registration values('" + regno + "','" +
firstName + "','" + lastName + "','" + Convert.ToDateTime(dateTimePicker1.Value) +
"', '" + gender + "','" + address + "','" + email + "','" + mobilePhone + "','" + homePhone
+ "','" + parentName + "','" + nic + "','" + contactNo + "')";

        if (con.State == ConnectionState.Closed) con.Open();
        SqlCommand cmnd = new SqlCommand(query_insert, con);

        cmnd.ExecuteNonQuery();
        con.Close();
        MessageBox.Show("Recoard Added Successfully!", "Registered Student!",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (SqlException ex)
    {
        string msg = "Insert Error:";
        msg += ex.Message;
    }
}

```

Update Button Code

```

private void btnupdate_Click(object sender, EventArgs e)
{
    string regNo = comboboxregno.Text;
    if (regNo != "New Register")
    {
        int regno = int.Parse(comboboxregno.Text);
        string firstName = txtfirstname.Text;
        string lastName = txtlastname.Text;
        dateTimePicker1.Format = DateTimePickerFormat.Custom;
        dateTimePicker1.CustomFormat = "dd/MM/yyyy";
    }
}

```

```

        string gender;

        if (radiobtnmale.Checked)
        {
            gender = "Male";
        }
        else
        {
            gender = "Female";
        }
        string address= txtaddress.Text;
        string email = txtemail.Text;
        int mobilePhone = int.Parse(txtmobilephone.Text);
        int homePhone = int.Parse(txthomephone.Text);
        string parentName = txtparentname.Text;
        string nic = txtnic.Text;
        int contactNo = int.Parse(txtcontacnum.Text);

        string query_insert = "UPDATE registration SET regNo = '" + regNo +
        "',firstName = '" + firstName + "',lastName = '" + lastName + "',dateOfBirth = '" +
        Convert.ToDateTime(dateTimePicker1.Value) + "',gender ='" + gender + "',address =
        '" + address + "',email ='" + email + "',mobilePhone = '" + mobilePhone + "',
        homePhone ='" + homePhone + "',parentName ='" + parentName + "', nic = '" + nic +
        "',contactNo ='" + contactNo + "' WHERE regNO =" + regNo;

        if (con.State == ConnectionState.Closed) con.Open();
        SqlCommand cmdnd = new SqlCommand(query_insert, con);
        cmdnd.ExecuteNonQuery();
        con.Close();
        MessageBox.Show("Record updated Successfully!", "Updated
        Student", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

```

Clear Button Code

```

private void btnclear_Click(object sender, EventArgs e)
{
    comboboxregno.Text = "";
    txtfirstname.Text = "";
    txtlastname.Text = "";
    dateTimePicker1.Format = DateTimePickerFormat.Custom;
    dateTimePicker1.CustomFormat = "dd/MM/yyyy";
    DateTime thisDay = DateTime.Today;
    dateTimePicker1.Text= thisDay.ToString();
}

```

```

radiobtnmale.Checked= false;
radiobtnfemale.Checked= false;

txtaddress.Text = "";
txtemail.Text = "";
txtmobilephone.Text = "";
txthomephone.Text = "";
txtparentname.Text = "";
txtnic.Text = "";
txtcontacnum.Text = "";
}

```

Delete Button Code

```

private void btndelete_Click(object sender, EventArgs e)
{
    var result = MessageBox.Show("Are you sure, Do you really want to Delete this
Recoard...?", "Delete", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if(result == DialogResult.Yes)
    {
        string no = comboboxregno.Text;

        string query_insert = "DELETE FROM registration WHERE regNo = " + no
+ "";
        con.Open();
        SqlCommand cmdnd = new SqlCommand(query_insert, con);
        _ = cmdnd.ExecuteNonQuery();
        con.Close();
        MessageBox.Show("Record Deleted Successfully!", "Deleted Student",
MessageBoxButtons.OK, MessageBoxIcon.Information);

        comboboxregno.Text = "";
        txtfirstname.Text = "";
        txtlastname.Text = "";
        dateTimePicker1.Format = DateTimePickerFormat.Custom;
        dateTimePicker1.CustomFormat = "dd/MM/yyyy";
        DateTime thisDay = DateTime.Today;
        dateTimePicker1.Text = thisDay.ToString();

        radiobtnmale.Checked = false;
    }
}

```



```

        radiobtnfemale.Checked = false;

        txtaddress.Text = "";
        txtemail.Text = "";
        txtmobilephone.Text = "";
        txthomephone.Text = "";
        txtparentname.Text = "";
        txtnic.Text = "";
        txtcontacnum.Text = "";
    }
    else if (result == DialogResult.Yes)
    {
        this.Close();
    }
}

```

Link Label Logout Code

```

private void linklbllogout_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    var result = MessageBox.Show("Are you sure,Do you really want to Logout...,?", "Logout", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if(result == DialogResult.Yes)
    {
        frmlogin obj = new frmlogin();
        obj.Show();
        this.Close();
    }
    else if (result==DialogResult.Yes)
    {
        Frmstudentreg obj= new Frmstudentreg();
        obj.Show();
        this.Close();
    }
}

```

Link Label Exit Code

```
private void linklblexit_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    var result = MessageBox.Show("Are you sure, Do you really want to exit....?",
"Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (result == DialogResult.Yes)
    {
        Application.Exit();
    }
    else if (result == DialogResult.No)
    {
        this.Close();
    }
}
}
```

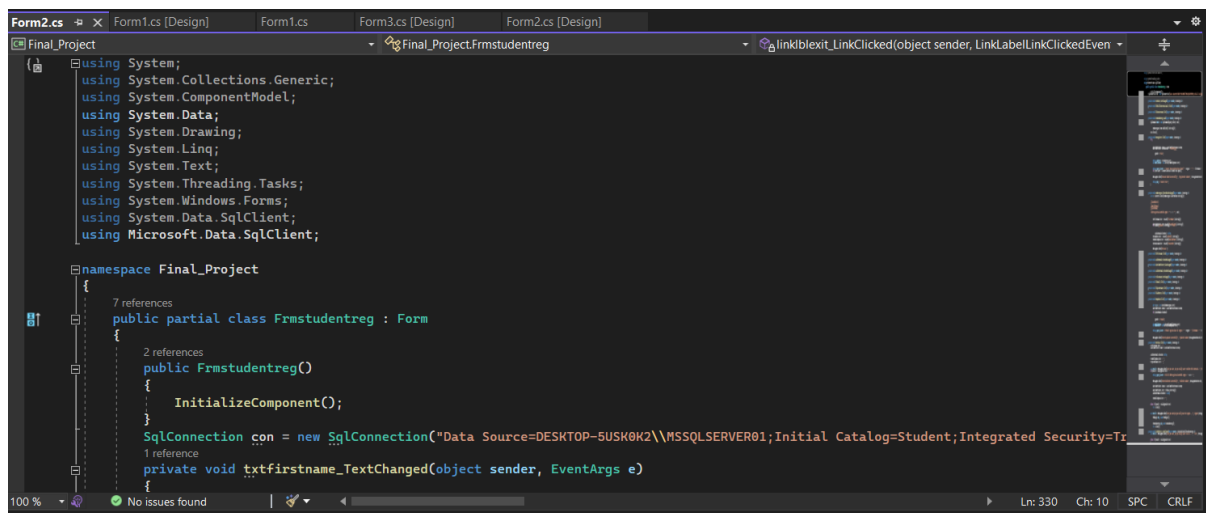
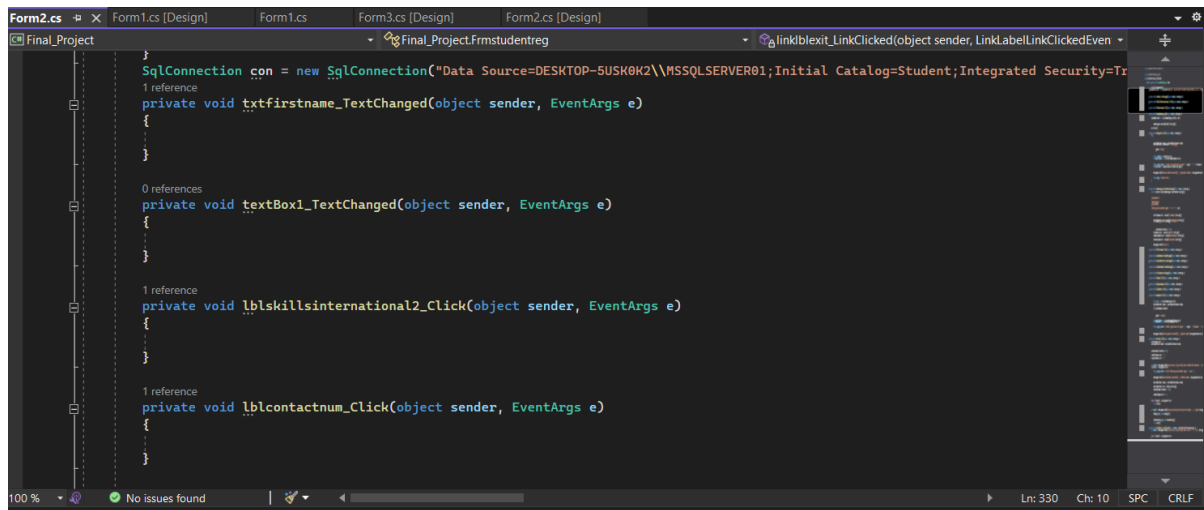


Figure 24 – Registration Form Coding 1



```

}
SqlConnection con = new SqlConnection("Data Source=DESKTOP-5USK0K2\\MSSQLSERVER01;Initial Catalog=Student;Integrated Security=Tr
1 reference
private void txtfirstname_TextChanged(object sender, EventArgs e)
{
}

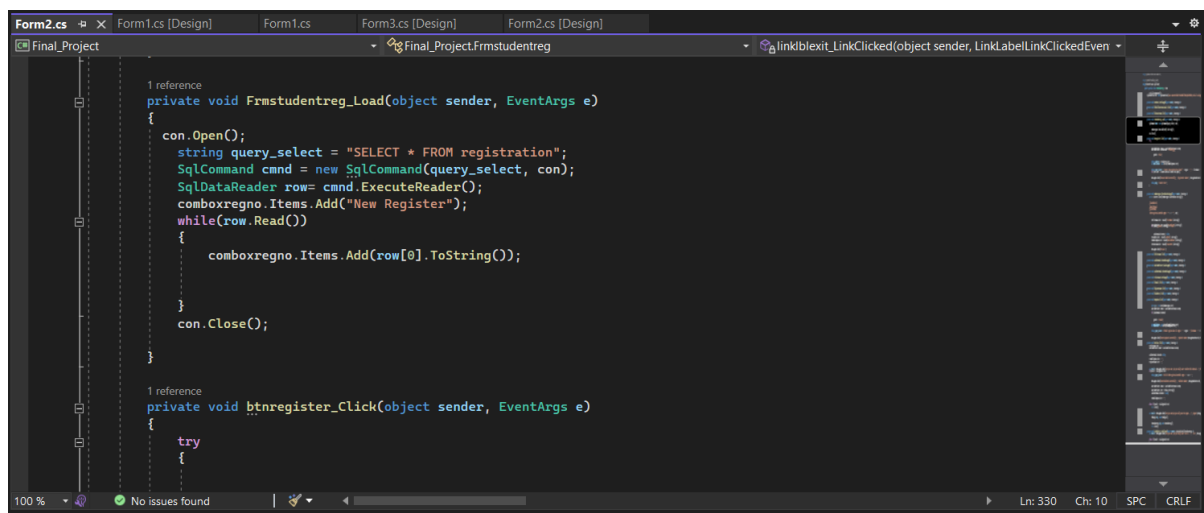
0 references
private void textBox1_TextChanged(object sender, EventArgs e)
{
}

1 reference
private void lblskillsinternational2_Click(object sender, EventArgs e)
{
}

1 reference
private void lblcontactnum_Click(object sender, EventArgs e)
{
}

```

Figure 25 - Registration Form Coding 2



```

1 reference
private void Frmstudentreg_Load(object sender, EventArgs e)
{
    con.Open();
    string query_select = "SELECT * FROM registration";
    SqlCommand cmdnd = new SqlCommand(query_select, con);
    SqlDataReader row= cmdnd.ExecuteReader();
    comboboxregno.Items.Add("New Register");
    while(row.Read())
    {
        comboboxregno.Items.Add(row[0].ToString());
    }
    con.Close();
}

1 reference
private void btnregister_Click(object sender, EventArgs e)
{
    try
    {

```

Figure 26 - Registration Form Coding 3

```

1 reference
private void btnregister_Click(object sender, EventArgs e)
{
    try
    {
        int regno = int.Parse(comboxregno.Text);
        string firstName = txtfirstname.Text;
        string lastName = txtlastname.Text;
        dateTimePicker1.Format = DateTimePickerFormat.Custom;
        dateTimePicker1.CustomFormat = "dd/MM/yyyy";
        string gender;

        if (radiobtnmale.Checked)
        {
            gender = "Male";
        }
        else
        {
            gender = "Female";
        }

        string address = txtaddress.Text;
        string email = txtemail.Text;
    }
}

```

Figure 27 - Registration Form Coding 4

```

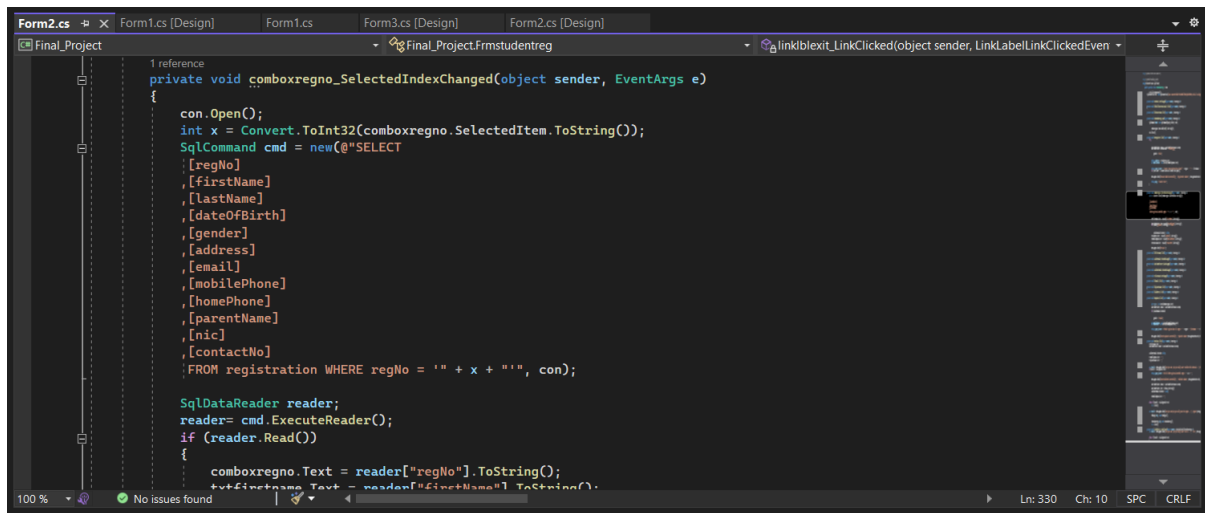
string email = txtemail.Text;
int mobilePhone = int.Parse(txtmobilephone.Text);
int homePhone = int.Parse(txthomephone.Text);
string parentName = txtparentname.Text;
string nic = txtnic.Text;
int contactNo = int.Parse(txtcontactnum.Text);
string query_insert = "insert into registration values('" + regno + "','" + firstName + "','" + lastName + "','" + Conve

if (con.State == ConnectionState.Closed) con.Open();
SqlCommand cmd = new SqlCommand(query_insert, con);

cmd.ExecuteNonQuery();
con.Close();
MessageBox.Show("Recoard Added Successfully!", "Registered Student!", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
catch (SqlException ex)
{
    string msg = "Insert Error:";
    msg += ex.Message;
}
}

```

Figure 28 - Registration Form Coding 5



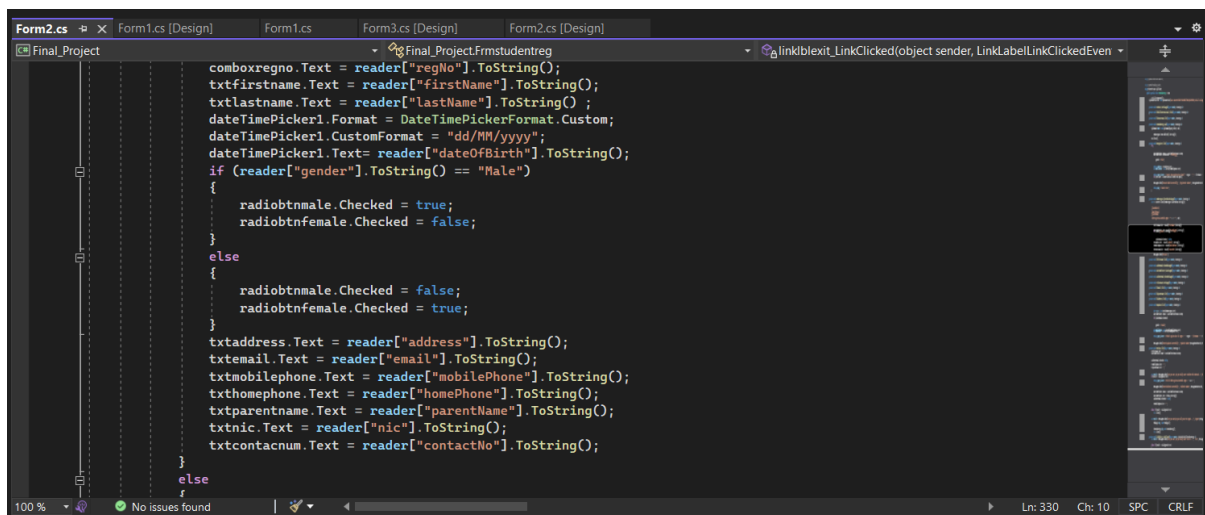
```

1 reference
private void comboBoxregno_SelectedIndexChanged(object sender, EventArgs e)
{
    con.Open();
    int x = Convert.ToInt32(comboBoxregno.SelectedItem.ToString());
    SqlCommand cmd = new(@"SELECT
        [regNo]
        ,[firstName]
        ,[lastName]
        ,[dateOfBirth]
        ,[gender]
        ,[address]
        ,[email]
        ,[mobilePhone]
        ,[homePhone]
        ,[parentName]
        ,[nic]
        ,[contactNo]
    FROM registration WHERE regNo = '" + x + "'", con);

    SqlDataReader reader;
    reader = cmd.ExecuteReader();
    if (reader.Read())
    {
        comboBoxregno.Text = reader["regNo"].ToString();
        txtfirstName.Text = reader["firstName"].ToString();
    }
}

```

Figure 29 - Registration Form Coding 6

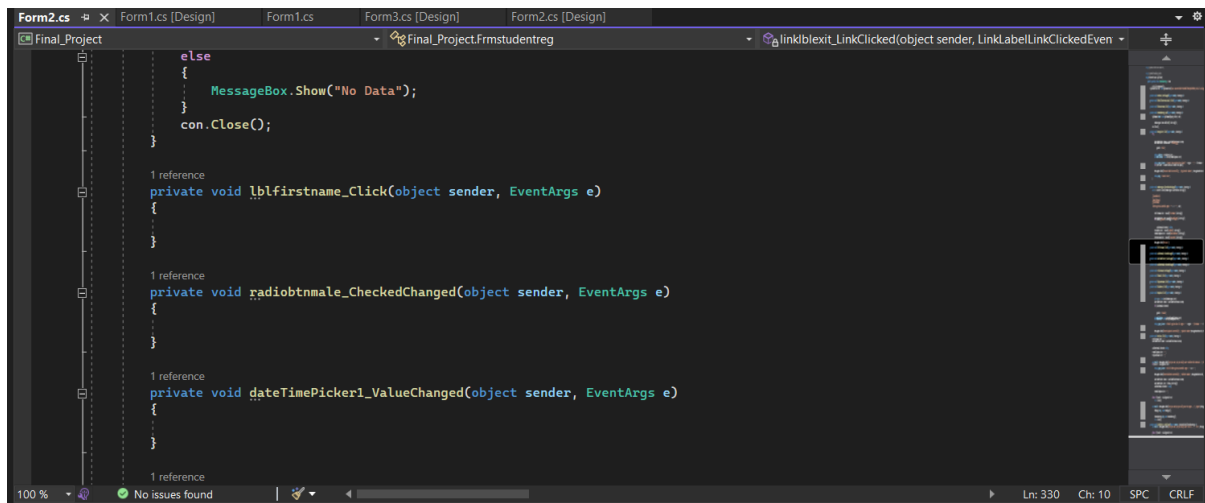


```

comboBoxregno.Text = reader["regNo"].ToString();
txtfirstName.Text = reader["firstName"].ToString();
txtlastName.Text = reader["lastName"].ToString();
dateTimePicker1.Format = DateTimePickerFormat.Custom;
dateTimePicker1.CustomFormat = "dd/MM/yyyy";
dateTimePicker1.Text = reader["dateOfBirth"].ToString();
if (reader["gender"].ToString() == "Male")
{
    radiobtnmale.Checked = true;
    radiobtnfemale.Checked = false;
}
else
{
    radiobtnmale.Checked = false;
    radiobtnfemale.Checked = true;
}
txtaddress.Text = reader["address"].ToString();
txtemail.Text = reader["email"].ToString();
txtmobilephone.Text = reader["mobilePhone"].ToString();
txtthomephone.Text = reader["homePhone"].ToString();
txtparentname.Text = reader["parentName"].ToString();
txttnic.Text = reader["nic"].ToString();
txtcontactnum.Text = reader["contactNo"].ToString();
}
else
{
}
}

```

Figure 30 - Registration Form Coding 7



```

else
{
    MessageBox.Show("No Data");
}
con.Close();
}

1 reference
private void lblfirstname_Click(object sender, EventArgs e)
{
}

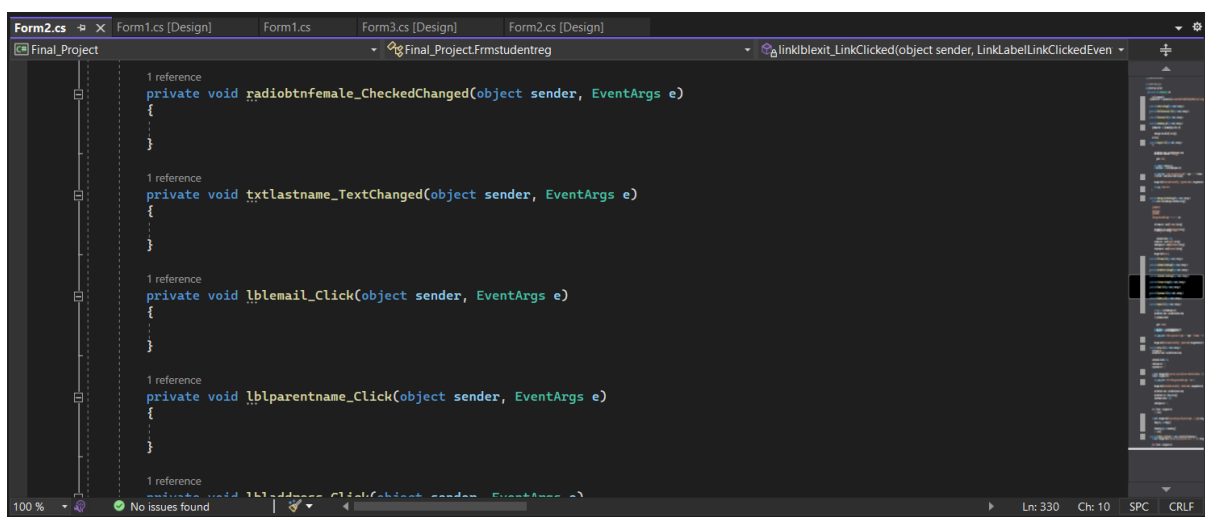
1 reference
private void radiobtnmale_CheckedChanged(object sender, EventArgs e)
{
}

1 reference
private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
{
}

1 reference

```

Figure 31 - Registration Form Coding 8



```

1 reference
private void radiobtnfemale_CheckedChanged(object sender, EventArgs e)
{
}

1 reference
private void txtlastname_TextChanged(object sender, EventArgs e)
{
}

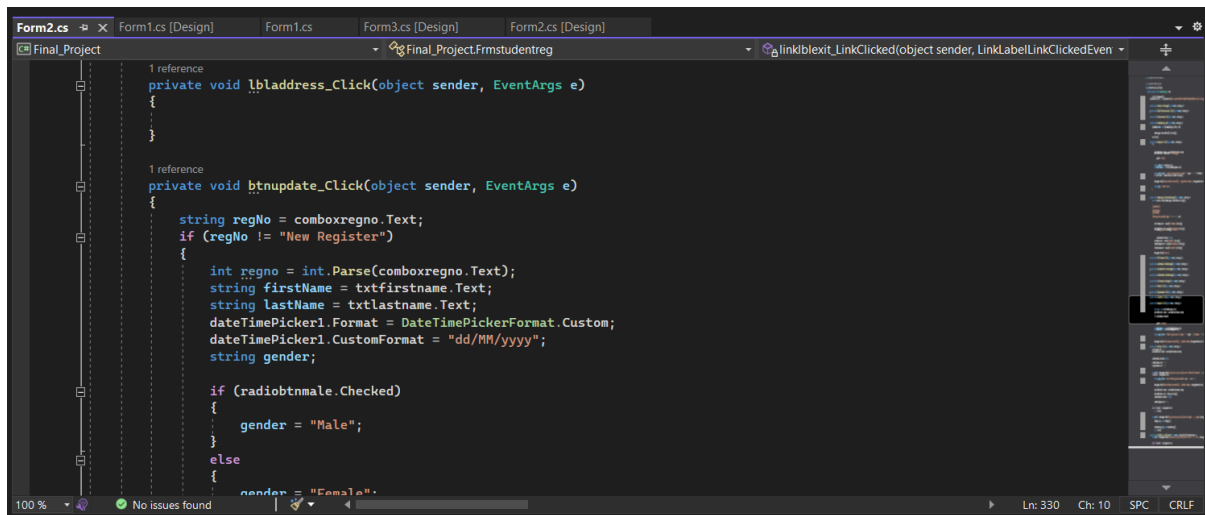
1 reference
private void lblmail_Click(object sender, EventArgs e)
{
}

1 reference
private void lblparentname_Click(object sender, EventArgs e)
{
}

1 reference
private void lbladdress_Click(object sender, EventArgs e)
{
}

```

Figure 32 - Registration Form Coding 9



```

1 reference
private void lbladdress_Click(object sender, EventArgs e)
{
}

1 reference
private void btnupdate_Click(object sender, EventArgs e)
{
    string regNo = comboboxregno.Text;
    if (regNo != "New Register")
    {
        int regno = int.Parse(comboboxregno.Text);
        string firstName = txtfirstname.Text;
        string lastName = txtlastname.Text;
        dateTimePicker1.Format = DateTimePickerFormat.Custom;
        dateTimePicker1.CustomFormat = "dd/MM/yyyy";
        string gender;

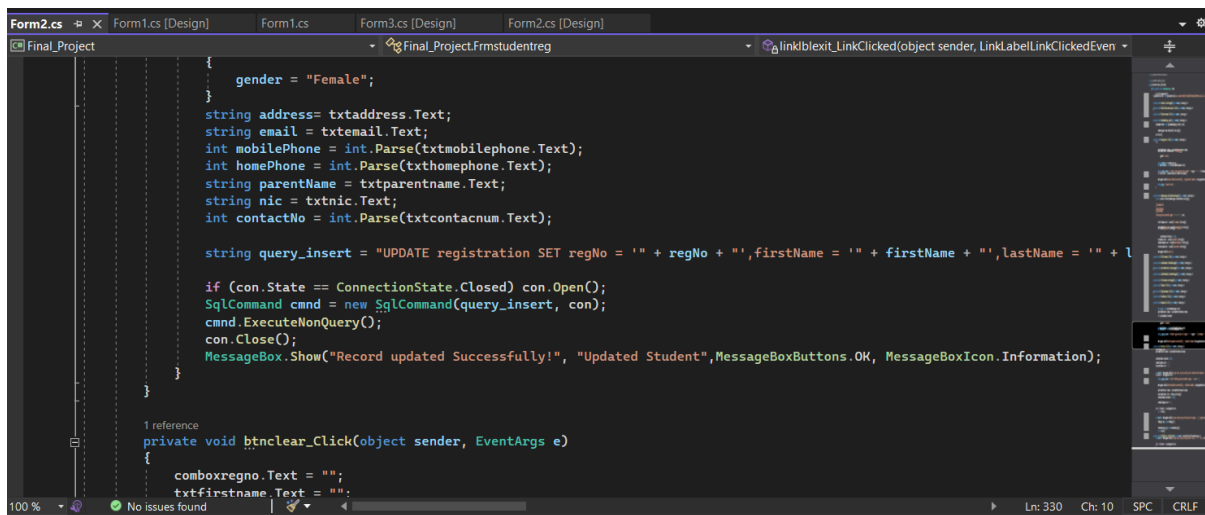
        if (radiobtnmale.Checked)
        {
            gender = "Male";
        }
        else
        {
            gender = "Female";
        }

        string query_insert = "UPDATE registration SET regNo = '" + regNo + "', firstName = '" + firstName + "', lastName = '" + lastName + "', mobilePhone = '" + mobilePhone + "', homePhone = '" + homePhone + "', parentName = '" + parentName + "', nic = '" + nic + "', contactNo = '" + contactNo + "'";

        if (con.State == ConnectionState.Closed) con.Open();
        SqlCommand cmd = new SqlCommand(query_insert, con);
        cmd.ExecuteNonQuery();
        con.Close();
        MessageBox.Show("Record updated Successfully!", "Updated Student", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

1 reference
private void btnclr_Click(object sender, EventArgs e)
{
    comboboxregno.Text = "";
    txtfirstname.Text = "";
}
    
```

Figure 33 - Registration Form Coding 10



```

        string query_insert = "UPDATE registration SET regNo = '" + regNo + "', firstName = '" + firstName + "', lastName = '" + lastName + "', mobilePhone = '" + mobilePhone + "', homePhone = '" + homePhone + "', parentName = '" + parentName + "', nic = '" + nic + "', contactNo = '" + contactNo + "'";

        if (con.State == ConnectionState.Closed) con.Open();
        SqlCommand cmd = new SqlCommand(query_insert, con);
        cmd.ExecuteNonQuery();
        con.Close();
        MessageBox.Show("Record updated Successfully!", "Updated Student", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

1 reference
private void btnclr_Click(object sender, EventArgs e)
{
    comboboxregno.Text = "";
    txtfirstname.Text = "";
}
    
```

Figure 34 - Registration Form Coding 11

```

1 reference
private void btnClear_Click(object sender, EventArgs e)
{
    comboBoxregno.Text = "";
    txtfirstname.Text = "";
    txtlastname.Text = "";
    dateTimePicker1.Format = DateTimePickerFormat.Custom;
    dateTimePicker1.CustomFormat = "dd/MM/yyyy";
    DateTime thisDay = DateTime.Today;
    dateTimePicker1.Text = thisDay.ToString();

    radioButtonmale.Checked = false;
    radioButtonfemale.Checked = false;

    txtaddress.Text = "";
    txtemail.Text = "";
    txtmobilephone.Text = "";
    txthomephone.Text = "";
    txtparentname.Text = "";
    txtnic.Text = "";
    txtcontactnum.Text = "";
}

1 reference
private void btnDelete_Click(object sender, EventArgs e)

```

Figure 35 - Registration Form Coding 12

```

1 reference
private void btnDelete_Click(object sender, EventArgs e)
{
    var result = MessageBox.Show("Are you sure, Do you really want to Delete this Recoard...?", "Delete", MessageBoxButtons.YesNo);
    if(result == DialogResult.Yes)
    {
        string no = comboBoxregno.Text;

        string query_insert = "DELETE FROM registration WHERE regNo = " + no + "";
        con.Open();
        SqlCommand cmd = new SqlCommand(query_insert, con);
        _ = cmd.ExecuteNonQuery();
        con.Close();
        MessageBox.Show("Record Deleted Successfully!", "Deleted Student", MessageBoxButtons.OK, MessageBoxIcon.Information);

        comboBoxregno.Text = "";
        txtfirstname.Text = "";
        txtlastname.Text = "";
        dateTimePicker1.Format = DateTimePickerFormat.Custom;
        dateTimePicker1.CustomFormat = "dd/MM/yyyy";
        DateTime thisDay = DateTime.Today;
        dateTimePicker1.Text = thisDay.ToString();

        radioButtonmale.Checked = false;
    }
}

```

Figure 36 - Registration Form Coding 13


```

radiobtnmale.Checked = false;
radiobtnfemale.Checked = false;

txtaddress.Text = "";
txtemail.Text = "";
txtmobilephone.Text = "";
txtthomphone.Text = "";
txtparentname.Text = "";
txtnic.Text = "";
txtcontactnum.Text = "";
}
else if (result == DialogResult.Yes)
{
    this.Close();
}
}

1 reference
private void linkbllogout_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    var result = MessageBox.Show("Are you sure,Do you really want to Logout...?","Logout",MessageBoxButtons.YesNo,MessageBoxIcon.Information);
    if(result == DialogResult.Yes)
    {
        frmlogin obj = new frmlogin();
        obj.Show();
    }
}

```

Figure 37 - Registration Form Coding 14

```

frmlogin obj = new frmlogin();
obj.Show();
this.Close();
}
else if (result==DialogResult.Yes)
{
    Frmstudentreg obj= new Frmstudentreg();
    obj.Show();
    this.Close();
}
}

1 reference
private void linkblexit_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    var result = MessageBox.Show("Are you sure, Do you really want to exit....?", "Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
    if (result == DialogResult.Yes)
    {
        Application.Exit();
    }
    else if (result == DialogResult.Yes)
    {
        this.Close();
    }
}

```

Figure 38 - Registration Form Coding

My Final Project Link

<https://drive.google.com/file/d/1uqHUhXBro0BEuzLUkJh2x3ukHEloEGje/view?usp=sharing>

References

Achmad, S., 2022. *Satya Achmad*. [Online]

Available at: <https://www.youtube.com/watch?v=YFsteFz80ZU>

[Accessed 3 December 2022].

CODDE, D., 2021. *show password with checkbox in C#*. [Online]

Available at: <https://www.youtube.com/watch?v=7NCmwa67e6s>

[Accessed 30 November 2022].

ESOFT Metro Campus, n.d. *Practical 11.1*, s.l.: s.n.

Igbe, C., 2022. *How to Create 3 Attempts Login System in C#*. [Online]

Available at: <https://www.youtube.com/watch?v=t4gzADIYg2M>

[Accessed December 4 2022].

Kiran, L. P. W., 2021. *Registration/SignUp form in C# Windows form App using Database Sql Server // Visual Studio 2022*. [Online]

Available at: <https://www.youtube.com/watch?v=eUur7KCm8i0>

[Accessed 1 December 2022].

ProgrammingKnowledge, 2013. *C# Tutorial 23: How to use DateTimePicker and save date in Database*. [Online]

Available at: <https://www.youtube.com/watch?v=Yb7R7Dr2DBk>

[Accessed 24 November 2022].

Tech, U., 2016. *How to connect SQL server database with visual Studio C#*. [Online]

Available at: <https://www.youtube.com/watch?v=V9r-Gp3uNCE>

[Accessed 26 November 2022].

THANK YOU!!

Especially I would like to thank our lecturer Mr. Pasan Weththasinghe for increased my knowledge about software development, especially because of him I was able to successfully complete my final project today. Apart from him I would also like to thank my family members, friends including Ishan and Dulmi who helped me to make my project successful. Finally, I would like to take this opportunity to thank all those who have supported me in the successful completion of this project.

