DiTEC
DIPLOMA IN INFORMATION TECHNOLOGY

Final Project

ESOFT METRO CAMPUS

Shaping Lives, Creating Futures.

Final Project – Assessment Sheet	
Programme	Assured Diploma in Information Technology
Assessor/Lecturer	Mr. Gihan Chanaka Ranasinghe
Unit	Final Project
Project Title	DITEC-110/145 – Final Project
Student's Name	K. Pramadu Shehan Chithakshana
Student's Registration No	E134048
Submission Date	05/10/2022
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

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.

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 (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).
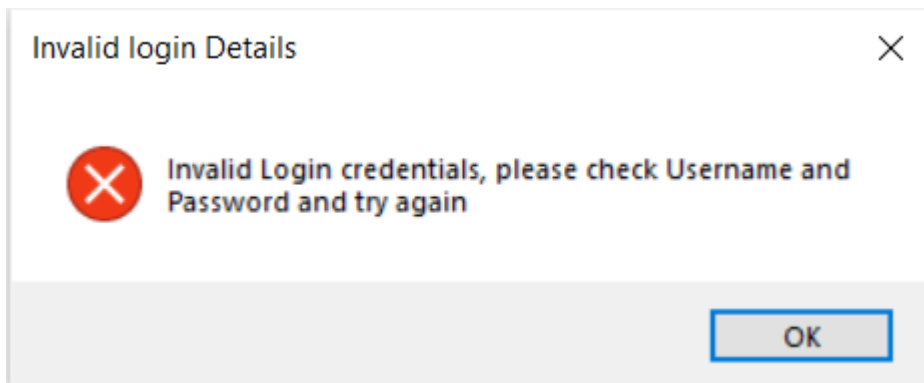

Figure III – Error Message

- 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 (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 (Exit) it needs to appear the following message box.
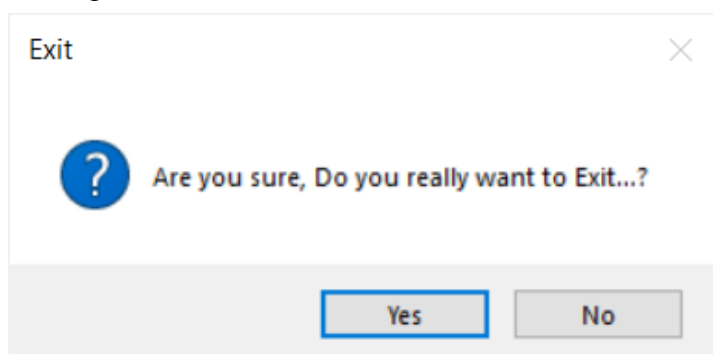

Figure IV – Exit Message

- 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 (Register). 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.
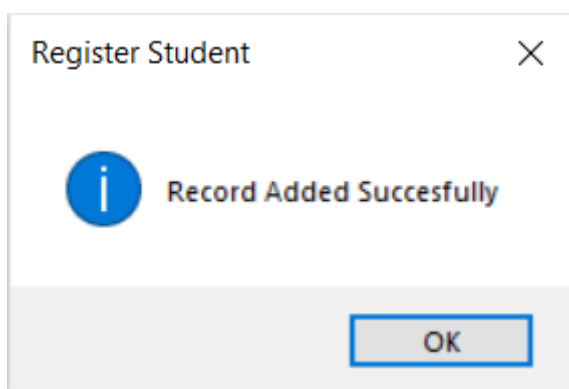
Figure V – Register Message Box

- Give the facility to the user to update records using **Update Butto**n (Update).
- 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** (Delete). 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.
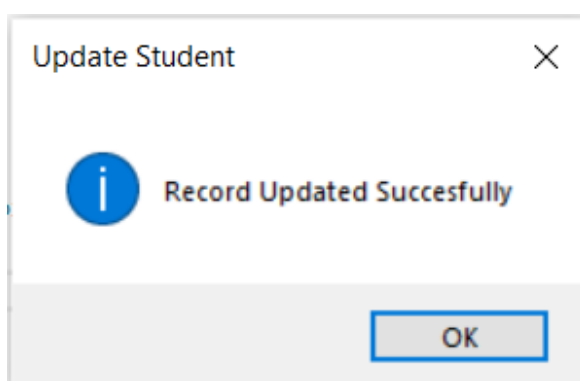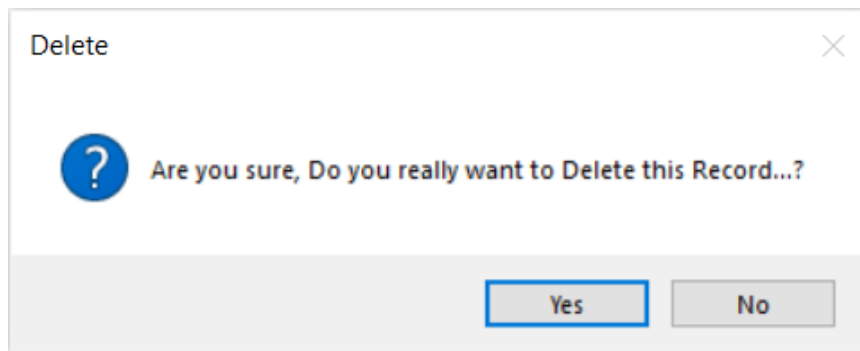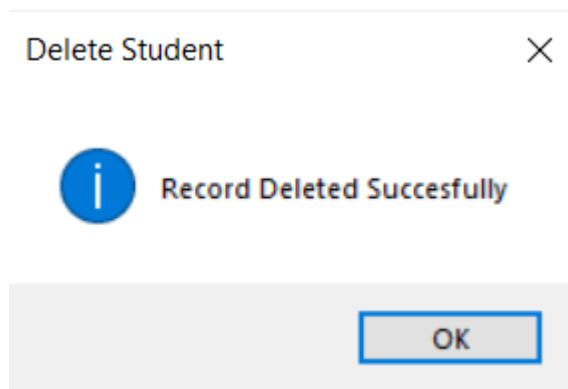

Figure VIII – Delete Confirmation Message Box

- When the user clicks on the **Reg No dropdown list** (Reg No ⌄), 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.

	usename	password
►*	*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	Varchar(50)
gender	varchar(50)
address	varchar(50)
email	varchar(50)
mobilePhone	Integer
homePhone	Integer
parentName	varchar(50)
nic	varchar(50)
contactNo	Integer

Acknowledgement

I would like to convey my special recognition to my lecturer Mr. Gihan Chanaka Ranasinghe who gave me this golden opportunity and help me to complete this project of software engineering on "FINAL PROJECT". Along with, I would like to thank with my parents & my friends who help me lot in finalizing in limited time frame.

Introduction

C# (pronounced "See Sharp") is a modern, object oriented and type-safe programming language. C# enables developers to build many types of secure and robust applications that runs in .NET. C# has its roots in the C family of languages and immediately familiar to C, C++, java, and JavaScript programmers. It was developed by Microsoft led Andres Hejlsberg and his team within the .NET initiative and was approved By European Computer Manufacture Association (ECMA) and International Standard Organization (ISO)

In my project I used C# programming language. Through that I made a software for "Skills - International School". This software that I have created can be used to register a new student, also allows easily update someone register details in the event of any changes or to remove details from the system & set up password with more secure only user can ingress to it.
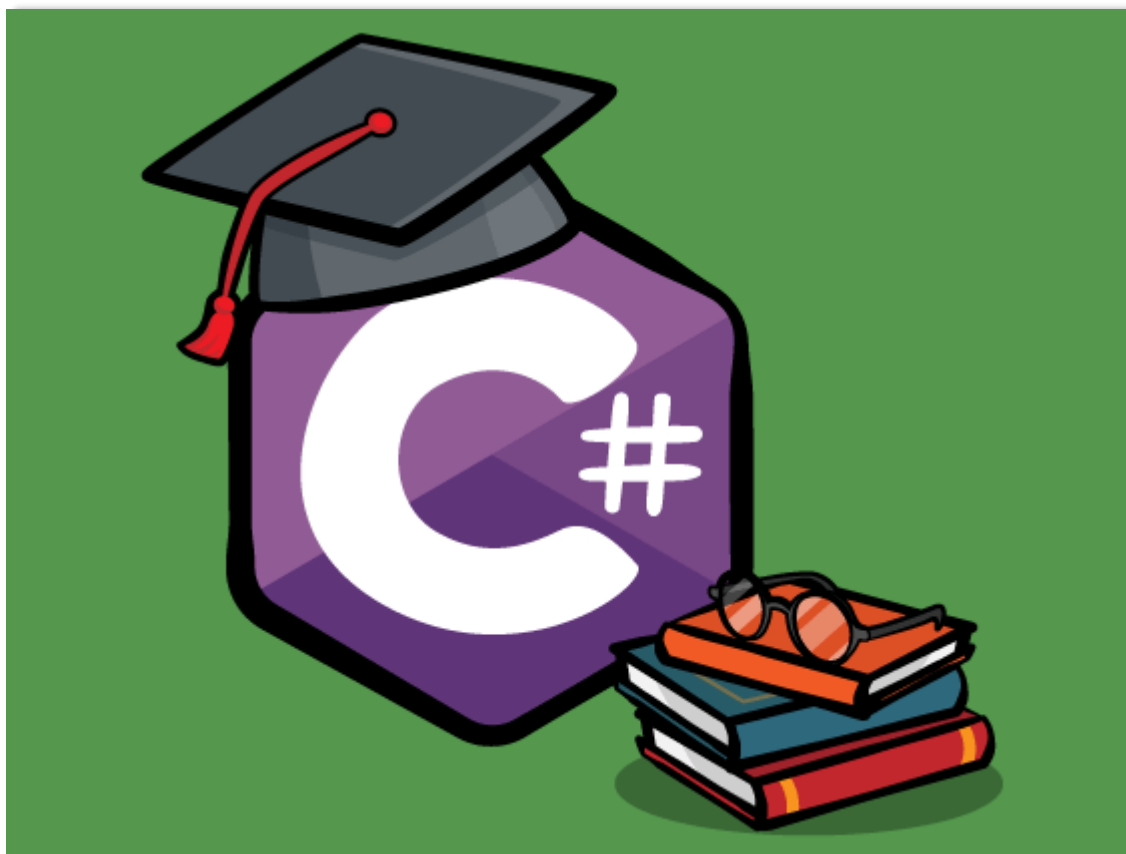
Table of Content

Table Of Figures

List Of Tables

Final Project

Diploma in Information Technology

ESOFT Metro Campus – Galle

01.   Form 01 (welcome window) -

a. Welcome Window Interface

figure 01

b. Properties

❖ Table 01: Textbox 01- Skills

Font	Century Schoolbook, 36pt, style=Bold, Italic
Name	textBox1
Size	340, 73
Text	"SKILLS"
Background color	41, 128, 185
Fore color	255, 224, 192

❖ Table 02: Textbox 02- International

Font	Century Schoolbook, 36pt, style=Bold, Italic
Name	textBox2
Size	340, 73
Text	"INTERNATIONAL"
Background color	41, 128, 185
Fore color	255, 224, 192

❖ Table 03: Progress Bar

Name	Progressbar1
Size	483, 46
Background color	white
Fore color	0, 192, 0
Style	Continuous

- **Background –**
  - Size = 834, 643
  - Start position = Center Screen
  - Layout = Stretch

c. Source Code
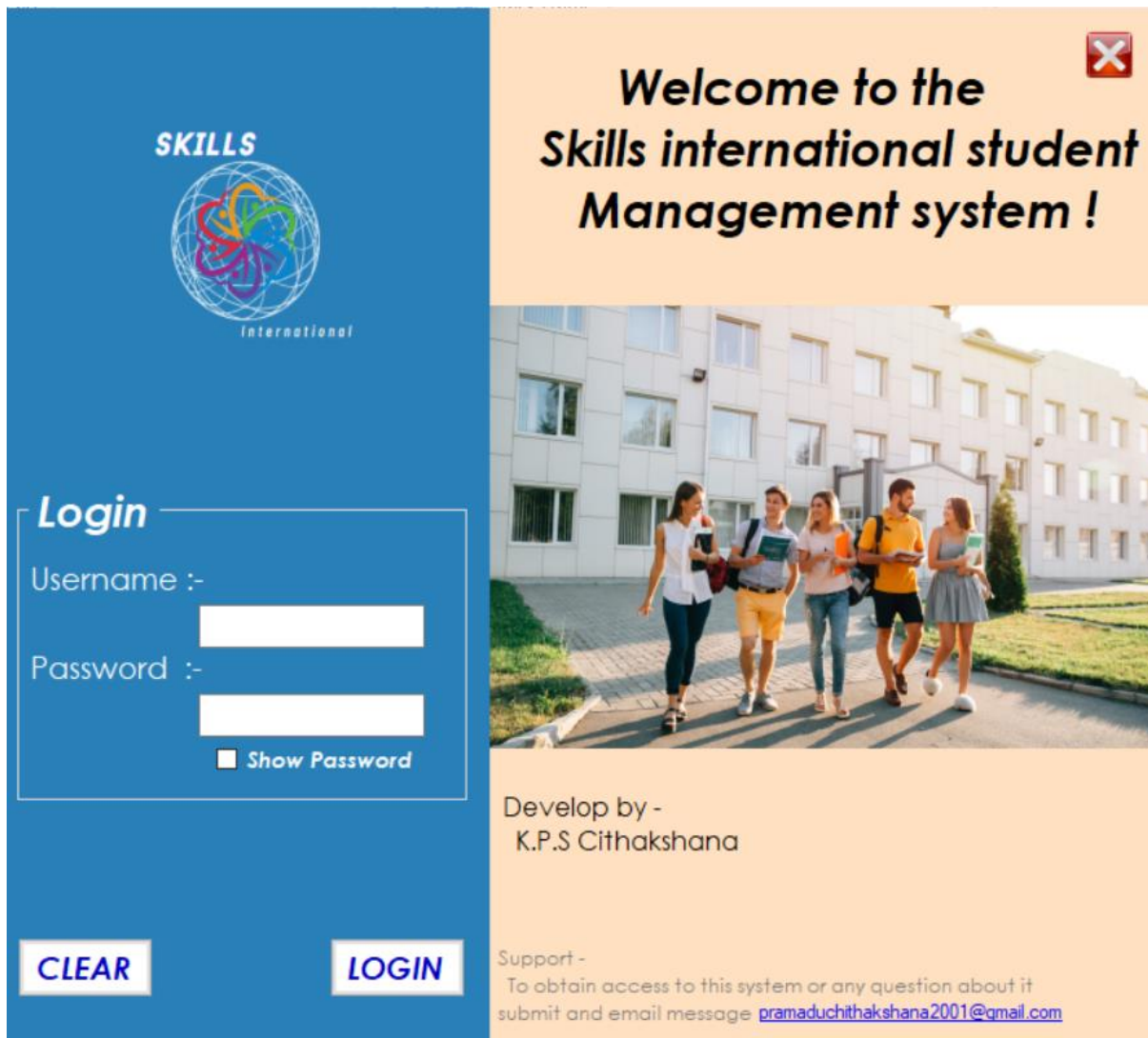
```csharp
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace DITEC_110_E134048_final_project
{
    public partial class welcomewindow : Form
    {
        public welcomewindow()
        {
            InitializeComponent();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            this.progressBar1.Increment(1);
            if (progressBar1.Value >= progressBar1.Maximum)
            {
                timer1.Stop();
                this.Hide();
                FormLogin log = new FormLogin();
                log.Show();
            }
        }
    }
}
```

02. Form 01 (Login) -

a. Welcome Window Interface

figure 02

b. Properties

❖ Table 04: Picture box 1

Name	Picturebox1
Size	286, 283
Location	62, 36
Background color	41, 128, 185
Background image	Simple Line Brown Minimalist Connectivity Logo (4)

❖ Table 05: Group box - Login

Font	Century Gothic, 19.8pt, style=Bold, Italic
Name	groupbox1
Size	370, 243
Text	"Login"
Background color	41, 128, 185
Fore color	White

❖ Table 06: Textbox 01 -Username

Font	Century Gothic, 10.8pt
Name	txtUN
Size	186, 30
Text	"Username"
Background color	Window
Fore color	Black
TabIndex	1

❖ Table 07: Textbox 01 – Password

Font	Century Gothic, 10.8pt
Name	txtPW
Size	186, 30
Text	"Password"
Background color	Window
Fore color	Black
TabIndex	2

❖ Table 08: Checkbox -Show password

Font	Century Gothic, 10.2pt, style=Bold, Italic
Name	checkBoxshow
Size	156, 25
Text	"Show Password"
Background color	41, 128, 185
Fore color	White
TabIndex	3

❖ Table 09: Button - Login

Font	Century Gothic, 13.8pt, style=Bold, Italic
Name	btnLogin
Size	112, 45
Text	"LOGIN"
Background color	White
Fore color	0, 0, 192
TabIndex	4

❖ Table 10: Button - Clear

Font	Century Gothic, 13.8pt, style=Bold, Italic
Name	btnClear
Size	112, 45
Text	"CLEAR"
Background color	White
Fore color	0, 0, 192
TabIndex	5

❖ Table 11: Button - Exit

Background image	Oxygen-Icons.org-Oxygen-Actions-application-exit
Name	btnExit
Size	5, 54
Text	--------
Background color	255, 224, 192
Fore color	255, 224, 192
TabIndex	6

❖ Table 12: Label 01

Font	Century Gothic, 13.8pt
Name	label1
Size	155, 30
Text	" Username :- "
Background color	41, 128, 185
Fore color	White

❖ Table 13: Label 02

Font	Century Gothic, 13.8pt
Name	Label2
Size	155, 30
Text	" Password :- "
Background color	41, 128, 185
Fore color	White

❖ Table 14: Label 03

Font	Century Gothic, 22.2pt, style=Bold, Italic
Name	Label3
Size	501, 138
Text	"Welcome to the Skills international student Management system!"
Background color	255, 224, 192
Fore color	ControlText

❖ Table 15: Label 04

Font	Century Gothic, 12pt
Name	Label4
Size	194, 46
Text	"Develop by - K.P.S Cithakshana"
Background color	255, 224, 192
Fore color	ControlText

❖ Table 16: Label 05

Font	Century Gothic, 9pt
Name	Label5
Size	413, 60
Text	"Support - To obtain access to this system or any question about it submit and email message to "
Background color	255, 224, 192
Fore color	Gray

➢ **Background –**
- Panel 1 [size = 401, 790]
- Panel 2 [size = 548, 790]
- Background image of Panel 2 = free_time_students_bachelor_s_campus_life_rhythm_five_friendly_students_are_walking
- Link label in panel 2 = pramaduchithakshana2001@gmail.com

c. Source Code

```csharp
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;
using System.Data.SqlClient;


namespace DITEC_110_E134048_final_project
{
    public partial class FormLogin : Form
    {
        public FormLogin()
        {
            InitializeComponent();
        }



        private void button3_Click(object sender, EventArgs e)
        {
            var result = MessageBox.Show("Are you sure,Do you realy want to
exit....?", "EXIT", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
            if (result == DialogResult.Yes)
            {
                Application.Exit();
            }
        }

        private void btnLogin_Click(object sender, EventArgs e)
        {
             string un, pw;
            un = txtUN.Text;
            pw = txtPW.Text;

            if (un == "Admin" && pw == "Skills@123")
            {
                form2 frm= new form2();
                frm.Show();
                this.Hide();
            }
            else
            {
                MessageBox.Show("Incorrect username or password!", "Unauthorized
Access Denied", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                txtUN.Text = "";
                txtPW.Text = "";
                txtUN.Focus();
            }
        }
        private void btnClear_Click(object sender, EventArgs e)
        {
            txtUN.Clear();
```

```csharp
            txtPW.Clear();
            txtUN.Focus();
        }

        private void label3_Click(object sender, EventArgs e)
        {

        }

        private void checkBoxshow_CheckedChanged(object sender, EventArgs e)
        {
            if (checkBoxshow.Checked == true)
            {
                txtPW.UseSystemPasswordChar = false;
            }
            else
            {
                txtPW.UseSystemPasswordChar= true;
            }
        }

        private void txtUN_TextChanged(object sender, EventArgs e)
        {

        }
    }
}
```

d. Message Boxes

❖ Login Button Message Box –
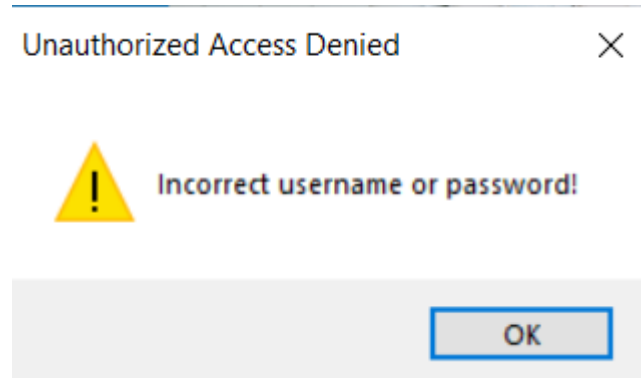
Figure 03

❖ Exit Button Message Box -
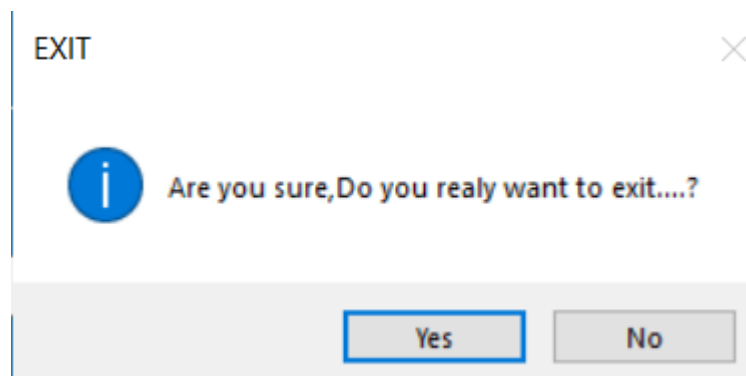
Figure 04

03. Form 03 (Registration) -

a. Welcome Window Interface

Figure 05

b. Properties

❖ Table 17: Link label – Logout

Font	Century Gothic, 13.8pt, style=Bold
Name	linkLOGOUT
Size	90, 28
Text	"Logout"
Background color	LightGray
Link color	0, 0, 255

❖ Table 18: Label 01

Font	Century Schoolbook, 28.2pt, style=Bold, Italic
Name	label1
Size	690, 57
Text	"SKILLS INTERNATIONAL"
Background color	41, 128, 185
Fore color	255, 224, 192

❖ Table 19: Label 02

Font	Century Schoolbook, 22.2pt, style=Bold, Italic
Name	label2
Size	441, 44
Text	"-Student Registration"
Background color	41, 128, 185
Fore color	White

❖ Table 20: Label 03

Font	Century Gothic, 12pt, style=Bold
Name	label3
Size	82, 23
Text	"Reg No"
Background color	Light Gray
Fore color	Black

❖ Table 21: Label 04

Font	Century Gothic, 12pt, style=Bold
Name	label4
Size	118, 23
Text	"Frist Name"
Background color	OldLace
Fore color	Black

❖ Table 22: Label 05

Font	Century Gothic, 12pt, style=Bold
Name	label5
Size	113, 23
Text	"Last Name"
Background color	OldLace
Fore color	Black

❖ Table 23: Label 06

Font	Century Gothic, 12pt, style=Bold
Name	label6
Size	128, 23
Text	"Date of Birth''
Background color	OldLace
Fore color	Black

❖ Table 24: Label 07

Font	Century Gothic, 12pt, style=Bold
Name	label7
Size	84, 23
Text	"Gender"
Background color	OldLace
Fore color	Black

❖ Table 25: Label 08

Font	Century Gothic, 12pt, style=Bold
Name	label8
Size	88, 23
Text	"Address"
Background color	OldLace
Fore color	Black

❖ Table 26: Label 09

Font	Century Gothic, 12pt, style=Bold
Name	label9
Size	173, 23
Text	"Contact Number"
Background color	OldLace
Fore color	Black

❖ Table 27: Label 10

Font	Century Gothic, 12pt, style=Bold
Name	label10
Size	144, 23
Text	"Mobile Phone"
Background color	OldLace
Fore color	Black

❖ Table 28: Label 11

Font	Century Gothic, 12pt, style=Bold
Name	label11
Size	136, 23
Text	"Home Phone"
Background color	OldLace
Fore color	Black

❖ Table 29: Label 12

Font	Century Gothic, 12pt, style=Bold
Name	label12
Size	137, 23
Text	"Parent Name"
Background color	OldLace
Fore color	Black

❖ Table 30: Label 13

Font	Century Gothic, 12pt, style=Bold
Name	label13
Size	47, 23
Text	"NIC"
Background color	OldLace
Fore color	Black

❖ Table 31: Label 14

Font	Century Gothic, 12pt, style=Bold
Name	label14
Size	62, 23
Text	"Email"
Background color	OldLace
Fore color	Black

❖ Table 32: Textbox - First Name

Font	Century Gothic, 12pt, style=Italic
Name	txtFN
Size	223, 32
Text	--------
Background color	Window
Fore color	Window text
TabIndex	2

❖ Table 33: Textbox - Last Name

Font	Century Gothic, 12pt, style=Italic
Name	txtLN
Size	223, 32
Text	----------
Background color	Window
Fore color	Window Text
TabIndex	3

❖ Table 33: Textbox - Address

Font	Century Gothic, 12pt, style=Italic
Name	txtADDRESS
Size	25, 102
Text	----------
Background color	Window
Fore color	Window Text
TabIndex	7

❖ Table 35: Textbox - Email

Font	Century Gothic, 12pt, style=Italic
Name	txtEmail
Size	280, 32
Text	-----------
Background color	Window
Fore color	Window Text
TabIndex	8

❖ Table 36: Textbox - Mobile Phone

Font	Century Gothic, 12pt, style=Italic
Name	txtMP
Size	
Text	----------
Background color	Window
Fore color	Window Text
TabIndex	9

❖ Table 37: Textbox - Home Phone

Font	Century Gothic, 12pt, style=Italic
Name	txtHP
Size	197, 32
Text	-----------
Background color	Window
Fore color	Window Text
TabIndex	10

❖ Table 38: Textbox - Parent Name

Font	Century Gothic, 12pt, style=Italic
Name	txtPARENT
Size	272, 32
Text	-----------
Background color	Window
Fore color	Window Text
TabIndex	11

❖ Table 39: Textbox - NIC

Font	Century Gothic, 12pt, style=Italic
Name	txtNIC
Size	197, 32
Text	-----------
Background color	Window
Fore color	Window Text
TabIndex	12

❖ Table 40: Textbox - Contact Number

Font	Century Gothic, 12pt, style=Italic
Name	txtCN
Size	
Text	-----------
Background color	Window
Fore color	Window Text
TabIndex	13

❖ Table 41: Button - Register

Font	Century Gothic, 13.2pt, style=Bold
Name	btnREG
Size	128, 42
Text	"Register"
Background color	255, 128, 0
Fore color	White
TabIndex	14

Table 42: Button – Update

Font	Century Gothic, 13.2pt, style=Bold
Name	btnUP
Size	129, 42
Text	"Update"
Background color	255, 128, 0
Fore color	White
TabIndex	16

❖ Table 43: Button - Search

Font	Century Gothic, 13.2pt, style=Bold
Name	btnSE
Size	128, 42
Text	"Update"
Background color	255, 128, 0
Fore color	White
TabIndex	15

❖ Table 44: Button - Clear

Font	Century Gothic, 13.2pt, style=Bold
Name	btnCL
Size	124, 42
Text	"Clear"
Background color	255, 128, 0
Fore color	White
TabIndex	17

❖ Table 45: Button – Delete

Font	Century Gothic, 13.2pt, style=Bold
Name	btnDEL
Size	125, 42
Text	"Delete"
Background color	255, 128, 0
Fore color	White
TabIndex	18

❖ Table 46: Radio button – Male

Font	Century Gothic, 12pt, style=Italic
Name	rbtM
Size	80, 27
Text	"Male"
Background color	Gold
Fore color	Chocolate
TabIndex	5

❖ Table 47: Radio button – Female

Font	Century Gothic, 12pt, style=Italic
Name	rbtF
Size	103, 27
Text	"Female"
Background color	Gold
Fore color	Chocolate
TabIndex	6

❖ Table 48: Group box 01

Font	Century Gothic, 12pt, style=Italic
Name	groupbox1
Size	847, 660
Text	"Registration"
Background color	LightGray
Fore color	Chocolate

❖ Table 49: Group box 02

Font	Century Gothic, 12pt, style=Italic
Name	Groupbox2
Size	785, 178
Text	"Basic Details"
Background color	OldLace
Fore color	Black

❖ Table 50: Group box 03

Font	Century Gothic, 12pt, style=Italic
Name	Groupbox3
Size	785, 149
Text	"Parent Details"
Background color	OldLace
Fore color	Black

Table 51: Group box 04

Font	Century Gothic, 12pt, style=Italic
Name	Groupbox4
Size	785, 212
Text	"Contact Details"
Background color	OldLace
Fore color	Black

❖ Table 52: Link label - Exit

Font	Century Gothic, 13.8pt, style=Bold
Name	linkEXIT
Size	50, 28
Text	"Exit"
Background color	LightGray
Link color	0, 0, 255

❖ Table 53; Date and Time Picker

Format	Short
Name	dateTimePickerDOB
Size	181, 32
Custom Format	dd-MM-yyyy
TabIndex	4

❖ Table 54; Combo box – Register

Font	Century Gothic, 12pt, style=Italic
Name	cmbREG
Size	133, 31
Background color	Window
Fore color	Window Text

c. Source Code

```csharp
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;
using System.Data.SqlClient;
namespace DITEC_110_E134048_final_project
{
    public partial class form2 : Form
    {
        public form2()
        {
            InitializeComponent();
        }
          SqlDataAdapter sqlDa = new SqlDataAdapter();
        SqlCommand cmd = new SqlCommand();
        SqlConnection con = new SqlConnection(@"Data Source=DESKTOP-6C9LP04;Initial
Catalog=Student;Integrated Security=True");
        String gen, RegNo;
        private void RegNoGen()
        {
            try
            {
                String Reccheck = "SELECT* FROM registration";
                con.Open();
                sqlDa = new SqlDataAdapter(Reccheck, con);
                DataTable Dt = new DataTable();
                sqlDa.Fill(Dt);
                int RegId = 0;
                if (Dt.Rows.Count > 0)
                {
                    String MaxNo = "SELECT MAX (SerialNo) FROM registration ";
                    SqlCommand cmd = new SqlCommand(MaxNo, con);
                    SqlDataReader R = cmd.ExecuteReader();
                    if (R.Read())
                    {
                        RegId = int.Parse(R.GetValue(0).ToString());
                    }
                    RegId = RegId + 1;
                }
                else
                {
                    RegId = 1;
                }
                lblregno.Text = "SKI-" + RegId.ToString();
                con.Close();
                lblregno.Text = RegNo;
            }
            catch (Exception GenErr)
            {
                MessageBox.Show("Error while Reg No Generator..." +
                Environment.NewLine + GenErr);
            }
        }
        private void Ali()
        {
```

```
            lblregno.Text = "";
            txtFN.Text = "";
            txtLN.Text = "";
            dateTimePickerDOB.Text = "";
            txtADDRESS.Text = "";
            txtEMAIL.Text = "";
            txtMP.Text = "";
            txtHP.Text = "";
            txtPARENT.Text = "";
            txtNIC.Text = "";
            textBoxCN.Text = "";
            rbtF.Checked = false;
            rbtM.Checked = false;
            cmbREG.Items.Clear();
            btnREG.Visible = true;
            btnDEL.Visible = false;
            btnUP.Visible = false;
            cmbREG.Visible = true;
            lblregno.Visible = true;
            cmbREG.Text="";
            cmbREG.Focus();
        }


        private void linkLabel2_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
        {
            var result = MessageBox.Show("Are you sure,Do you realy want to
exit....?", "EXIT", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
            if (result == DialogResult.Yes)
            {
                Application.Exit();
            }
        }
        private void button1_Click(object sender, EventArgs e)
        {

            try
            {
                if (rbtM.Checked == true) { gen = "Male"; }
                else if (rbtF.Checked == true) { gen = "Female"; }
                String queReg = "INSERT INTO registration (RegNo, FirstName, LastName,
DateofBirth, Gender, Address, Email, MobilePhone, HomePhone, ParentName, NIC,
ContactNo)VALUES('" + cmbREG.Text + "','" + txtFN.Text + "','" + txtLN.Text + "','" +
dateTimePickerDOB.Text + "','" + gen + "','" + txtADDRESS.Text + "','" + txtEMAIL.Text
+ "','" + txtMP.Text + "','" + txtHP.Text + "','" + txtPARENT.Text + "','" +
txtNIC.Text + "','" + textBoxCN.Text + "')";
                con.Open();
                cmd = new SqlCommand(queReg, con);
                cmd.ExecuteNonQuery();
                con.Close();
                MessageBox.Show( "Record Added Succesfully", "Register Student",
MessageBoxButtons.OK,  MessageBoxIcon.Information);
                Ali();

            }
            catch (Exception RegErr)
            {
                MessageBox.Show("Error while Register..." +
                Environment.NewLine + RegErr);
            }
        }
```

```csharp
        private void linkLOGOUT_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
        {
            FormLogin obj = new FormLogin();
            obj.Show();
            this.Close();

        }


        private void btnDEL_Click(object sender, EventArgs e)
        {
            try
            {
                DialogResult resDel = MessageBox.Show("Are you sure,Do  you really
want to Delete this Record...? ", "Delete",
                MessageBoxButtons.YesNo, MessageBoxIcon.Question);
                if (resDel == DialogResult.Yes)
                {
                    String queDel = "DELETE FROM registration WHERE  RegNo = '" +
cmbREG.SelectedItem + "'";
                    con.Open();
                    SqlCommand cmd = new SqlCommand(queDel, con);
                    cmd.ExecuteNonQuery();
                    con.Close();
                    MessageBox.Show("Record Deleted Succesfully","Delete Student",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                    Ali();
                }
            }
            catch (Exception DeLErr)
            {
                MessageBox.Show("Error while Delete..." +Environment.NewLine +
DeLErr);
            }

        }

        private void cmbREG_SelectedIndexChanged(object sender, EventArgs e)
        {
            {
                try
                {
                    if (cmbREG.SelectedIndex > 0)
                    {
                        RegNo = cmbREG.SelectedItem.ToString();
                        String selData = "SELECT* FROM registration WHERE RegNO = '" +
RegNo + "'";
                        con.Open();
                        cmd = new SqlCommand(selData, con);
                        SqlDataReader R = cmd.ExecuteReader();
                        if (R.Read())
                        {
                            txtFN.Text = R.GetValue(2).ToString();
                            txtLN.Text = R.GetValue(3).ToString();
                            dateTimePickerDOB.Text = R.GetValue(4).ToString();
                            gen = R.GetValue(5).ToString();
                            if (gen.Equals("Male"))  { rbtM.Checked = true; }
                            else if (gen.Equals("Female")) { rbtF.Checked = true;}
                            txtADDRESS.Text = R.GetValue(6).ToString();
                            txtEMAIL.Text = R.GetValue(7).ToString();
                            txtMP.Text = R.GetValue(8).ToString();
```

```
                        txtHP.Text = R.GetValue(9).ToString();
                        txtPARENT.Text = R.GetValue(10).ToString();
                        txtNIC.Text = R.GetValue(11).ToString();
                        textBoxCN.Text = R.GetValue(12).ToString();
                    }
                    con.Close();
                    }
                     else
                {
                    txtFN.Text = "";
                    txtLN.Text = "";
                    dateTimePickerDOB.Text = "";
                    txtADDRESS.Text = "";
                    txtEMAIL.Text = "";
                    txtHP.Text = "";
                    txtMP.Text = "";
                    txtPARENT.Text = "";
                    txtNIC.Text = "";
                    textBoxCN.Text = "";
                    rbtM.Checked = false;
                    rbtF.Checked = false;
                }
                    }
                    catch (Exception SelErr)
                  {
                    MessageBox.Show("Error while selecting from ComboBox..." +
Environment.NewLine + SelErr);
                    }
                  }
                }

        private void btnUP_Click(object sender, EventArgs e)
        {

            try
          {
                DialogResult resup = MessageBox.Show("Do you want to  Update the
record ? ", "Update", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
                if (resup == DialogResult.Yes)
                {
                    if (rbtM.Checked == true) { gen = "Male"; }
                    else if (rbtF.Checked == true) { gen = "Female"; }
                    string queup = "UPDATE  registration SET FirstName  = '" +
txtFN.Text + "', LastName = '" + txtLN.Text + "',DateofBirth = '" +
dateTimePickerDOB.Text + "', Gender = '" + gen + "', Address = '" + txtADDRESS.Text +
"', Email = '" + txtEMAIL.Text + "', MobilePhone = '" + txtMP.Text + "', HomePhone =
'" + txtHP.Text + "',ParentName = '" + txtPARENT.Text + "', NIC = '" + txtNIC.Text +
"', ContactNo = '" + textBoxCN.Text + "' WHERE RegNO = '" + cmbREG.SelectedItem + "'";
                    con.Open();
                    cmd = new SqlCommand(queup, con);
                    cmd.ExecuteNonQuery();
                    con.Close();
                    MessageBox.Show( "Record Updated Successfully", "Update Student",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                    Ali();
                    cmbREG.Text = ToString();
                    txtMP.Text = ToString();
                    txtHP.Text = ToString();
                }
              }
                catch (Exception UpErr)
              {
```

```csharp
                    MessageBox.Show("Error while Update..." + Environment.NewLine +
UpErr);
                }
            }

        private void btnSE_Click(object sender, EventArgs e)
        {
            try
            {
                String GetRegNo = "SELECT RegNO FROM registration";
                con.Open();
                sqlDa = new SqlDataAdapter(GetRegNo, con);
                DataTable Dts = new DataTable();
                sqlDa.Fill(Dts);

                cmbREG.Items.Clear();
                cmbREG.Items.Add("--SELECT--");
                foreach (DataRow row in Dts.Rows)
                {
                    cmbREG.Items.Add(row["RegNo"]);
                }
                cmbREG.SelectedIndex = 0;
                con.Close();
                cmbREG.Visible = true;
                btnREG.Visible = false;
                btnDEL.Visible = true;
                btnUP.Visible = true;
                lblregno.Visible = true;
            }
            catch (Exception SeaErr)
            {
                MessageBox.Show("Error while search..." +Environment.NewLine +
SeaErr);
            }
        }

        private void form2_Load(object sender, EventArgs e)
        {
            RegNoGen();
        }

        private void btnCl_Click(object sender, EventArgs e)
        {
            Ali();

        }
    }
}
```

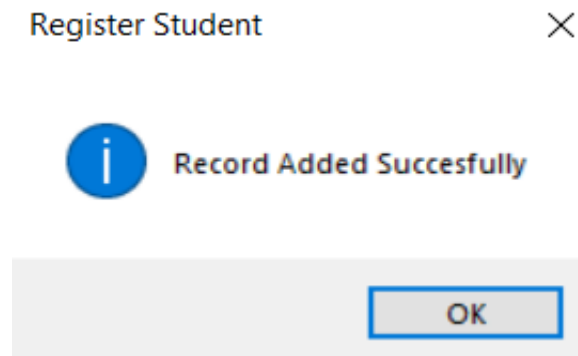# d. Message Boxes

❖ Register Message Box -
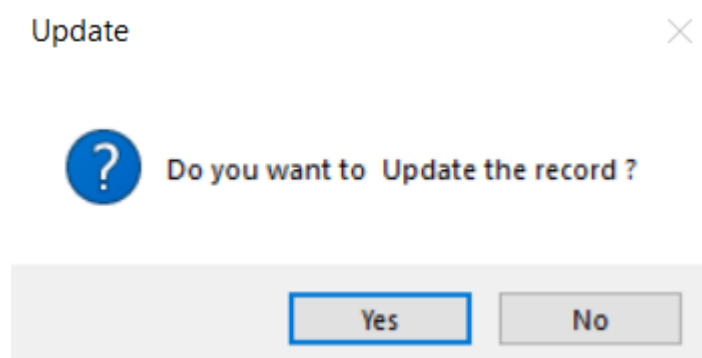

Figure 6

❖ Update Message Box -

Figure 07

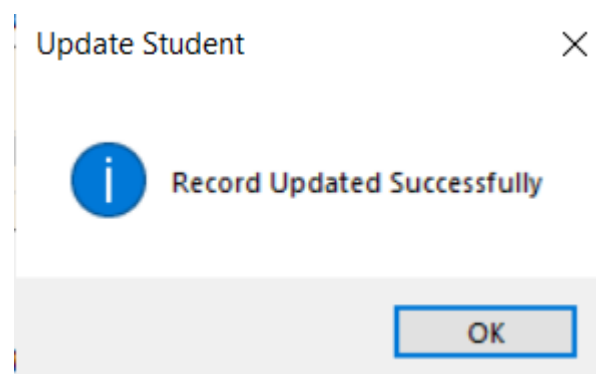❖ Update Successful Message Box -
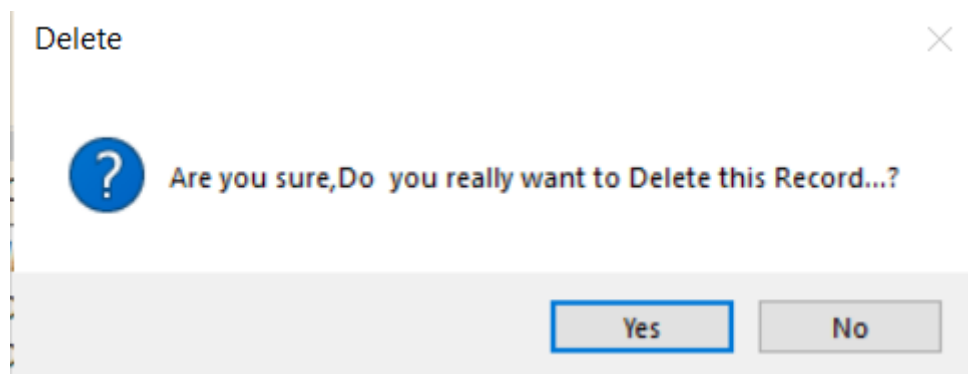

Figure 08

❖ Delete Message Box -
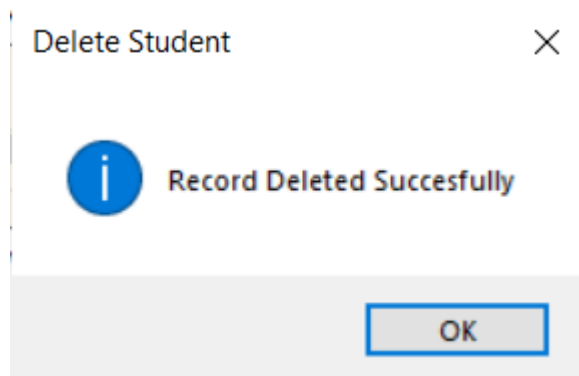

Figure 09

❖ Delete Successful Message Box -

Figure 10

❖ Exit Message Box -
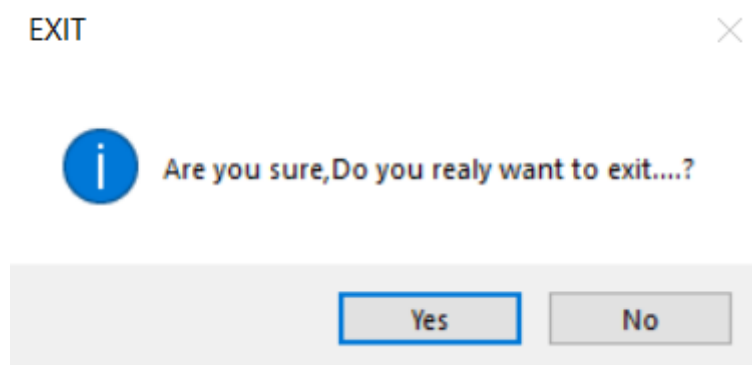
Figure 11

04. Database Table Design

DESKTOP-6C9LP04.St...- dbo.registration ✕		
Column Name	Data Type	Allow Nulls
▶ SerialNo	int	☐
🔑 RegNo	int	☐
FirstName	varchar(50)	☑
LastName	varchar(50)	☑
DateofBirth	varchar(50)	☑
Gender	varchar(50)	☑
Address	varchar(50)	☑
Email	varchar(50)	☑
MobilePhone	int	☑
HomePhone	int	☑
ParentName	varchar(50)	☑
NIC	varchar(50)	☑
ContactNo	int	☑
		☐

figure 12

05. Database Table View -

	SerialNo	RegNo	FirstName	LastName	DateofBirth	Gender	Address	Email	MobilePho...	HomePhone	ParentName	NIC	ContactNo
▶	11	0	maduri	heshana	10/25/1997	Female	hikkaduwag...	maduri1997...	754438901	112456783	Gayani	1977300144...	742114568
	12	1	pramadu ch...	Shehan	11/15/2001	Male	334/Dgalle ...	kprmaduchi...	785553667	917921496	ramani	2001320013...	771867045
	17	2	ramani	palihepitiye	5/1/1967	Female	Halvitigala ...		771234567	0	P.G Saiman	1967200134...	0
	19	4	Shanuka	Sanjeewa	6/15/2001	Male	udaragama...	Sanjeewa20...	774987051	0	T.L Sami	1970300248...	0
	14	5	maduri	heshana	10/25/1997	Female	hikkaduwag...	maduri1997...	754438901	112456783	Gayani	1977300144...	742114568
	21	6	Sahan	Kaushalya	2/25/2004	Male	new kandy r...	kaushalyaSa...	0	114300200	Jayalath	1964532044v	78446567
	23	7	Ushan	Liyanage	1/31/2002	Male	galle roadN...	UshanLi200...	754598211	914598765	Kumari	197188910v	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 13

Project link -:

https://drive.google.com/file/d/1gGAhGEovZtaod2VYdr_ TojUl2OxWJb84/view?usp=sharing