



Final Project

ESOFT METRO CAMPUS

Shaping Lives, Creating Futures.

Contents

Table of Figures	2
Acknowledgement	3
Introduction.....	4
Development Process.....	4
Requirement Gathering	4
Limitations.....	5
Planning.....	6
Flow chart for its functionalities.....	6
Illustrative Diagram and Respective Codes	12
Login Page.....	12
Login Page: Illustrative Diagram	12
Login Page: Code	12
Dashboard Page.....	14
Dashboard Page: Illustrative Diagram.....	14
Dashboard Page: Code.....	14
Student Menu	16
Student Menu: Illustrative Diagram	16
Student Menu: Code	16
Registration Page.....	17
Registration Page: Illustration Diagram	17
Registration Page: Code	17
Search Page	23
Search Page: Illustration Page	23
Search Page: Code	23
Add Member Page.....	24
Add Member Page: Illustration Diagram	24
Add Member Page: Code	24
References.....	25

Table of Figures

Figure 1:Login Form Flowchart	6
Figure 2: Registration Flowchart.....	7
Figure 3:Update Record Flowchart	8
Figure 4:Delete Records Flowchart.....	9
Figure 5: Search Record Flowchart	10
Figure 6:ER Diagram.....	11
Figure 7:Login Page Illustrative Diagram	12
Figure 8: Login Page Sign-In Button	12
Figure 9: Login Page Clear Button.....	13
Figure 10: Login Page Close Button	13
Figure 11: Login Page Password Show/Hide Button.....	13
Figure 12: Login Page Close Button	13
Figure 13: Dashboard Illustrative Diagram	14
Figure 14: Dashboard Page Dashboard Menu Button	14
Figure 15: Dashboard Page Student Menu Button	14
Figure 16: Dashboard Page Minimize Button	15
Figure 17: Dashboard Page Exit Button	15
Figure 18: Dashboard Page Sign-Out Button	15
Figure 19: Student Menu Illustrative Diagram.....	16
Figure 20: Student Menu Registration Button.....	16
Figure 21: Student Menu Search Button	16
Figure 22: Registration Page Illustrative Diagram.....	17
Figure 23: Registration Page Delete Button.....	17
Figure 24: Registration Page Clear Button.....	18
Figure 25: Registration Page Search Button 1	19
Figure 26: Registration Page Search Button 2	19
Figure 27: Registration Page Update Button 1	20
Figure 28: Registration Page Update Button 2	20
Figure 29: Registration Page Register Button 1	21
Figure 30: Registration Page Register Button 2	21
Figure 31: Registration Page Register Button 3	21
Figure 32: Registration Page Register Button 4	22
Figure 33: Search Page Illustrative Diagram	23
Figure 34: Search Page Display Button	23
Figure 35: Add Member Page Illustrative Diagram.....	24
Figure 36: Add Member Page Add Button.....	24
Figure 37: Add Member Page Clear Button	24
Figure 38:Login Page invalid Credentials	26
Figure 39: Registration Successful	26
Figure 40: Registration was unsuccessful	26
Figure 41: Update was Successful.....	27
Figure 42:Record Deleted Successful.....	27
Figure 43:Searching record is Successful	27
Figure 44:Searching Table is Successful	28
Figure 45: Adding Member is Successful	28

Acknowledgement

I want to thank my lecturer, Ms. Tiyetra, for creating a good atmosphere that helped me finish this work. She gave me lots of good advice and support and I really appreciate it.

I also want to thank everyone else who helped me especially my teammates. Their help was important for getting this work done.

Best regards,

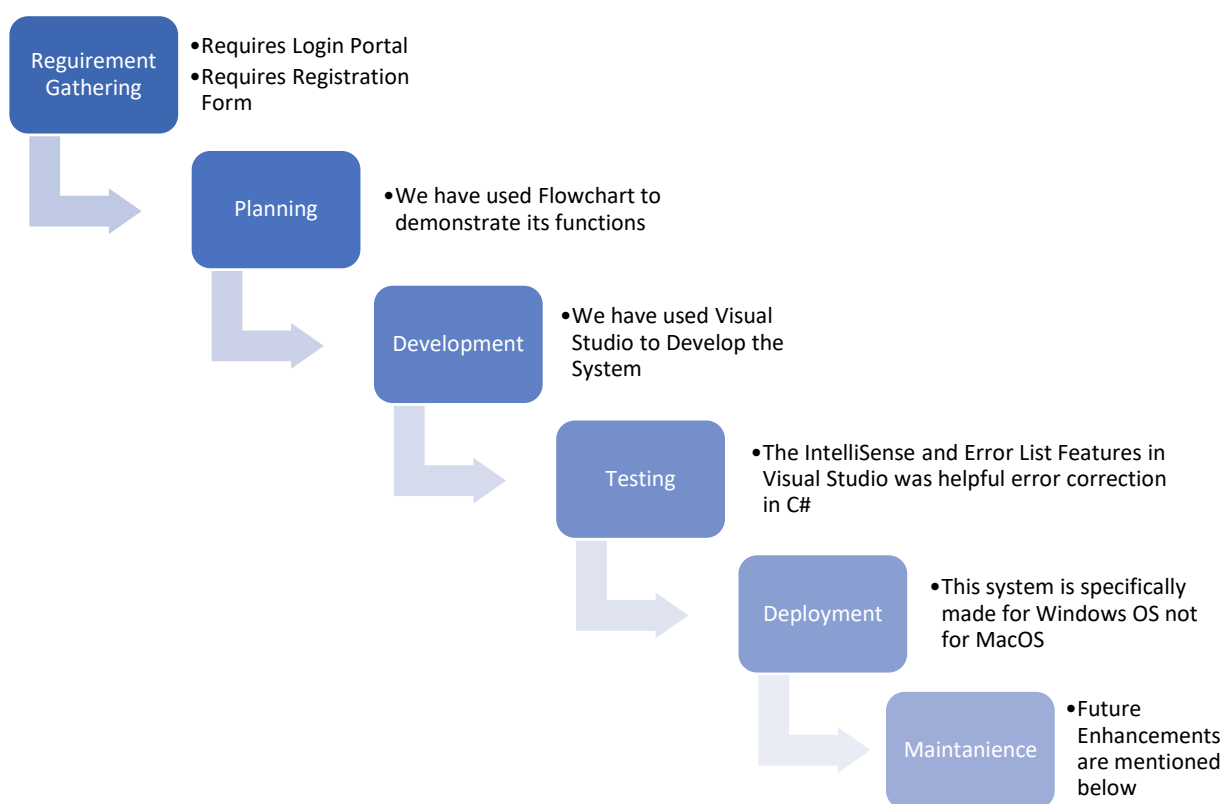
K. Gynorrj.

Introduction

This is an Official Document to our new Student Management System made for Student Management System for Skills Management System. The Student Management System is an application created to reduce the work load of Registration Staff. This documentation provides a fully detailed explanation of how system works and its functionalities.

Development Process

I have used waterfall model to explain the Development.



Requirement Gathering

The requirement required by the Institutes by the Student Management System are:

1. Should have a functioning login portal.
2. Ability to Save a student's data.
3. Ability to Update a student's details.
4. Ability to Delete a student detail.
5. Ability to Clear the details entered.

We are able to develop a system to fulfil their requirement and some additional feature to Student management system what we felt it would be useful to Registration Staff to increase their productivity:

1. Ability to navigate through dashboard and have more menus on it.
2. Ability to view the Table.
3. Ability to add registration member to the system.

Limitations

Even though we are able to fulfil the Clients requirement our team have found few limitations and vulnerabilities of the system:

1. Compatibilities Issues.
2. Security Risk (Risk of Being attacked by SQL Injection).
3. Unable to Filter the details in Data view.

Planning

Flow chart for its functionalities

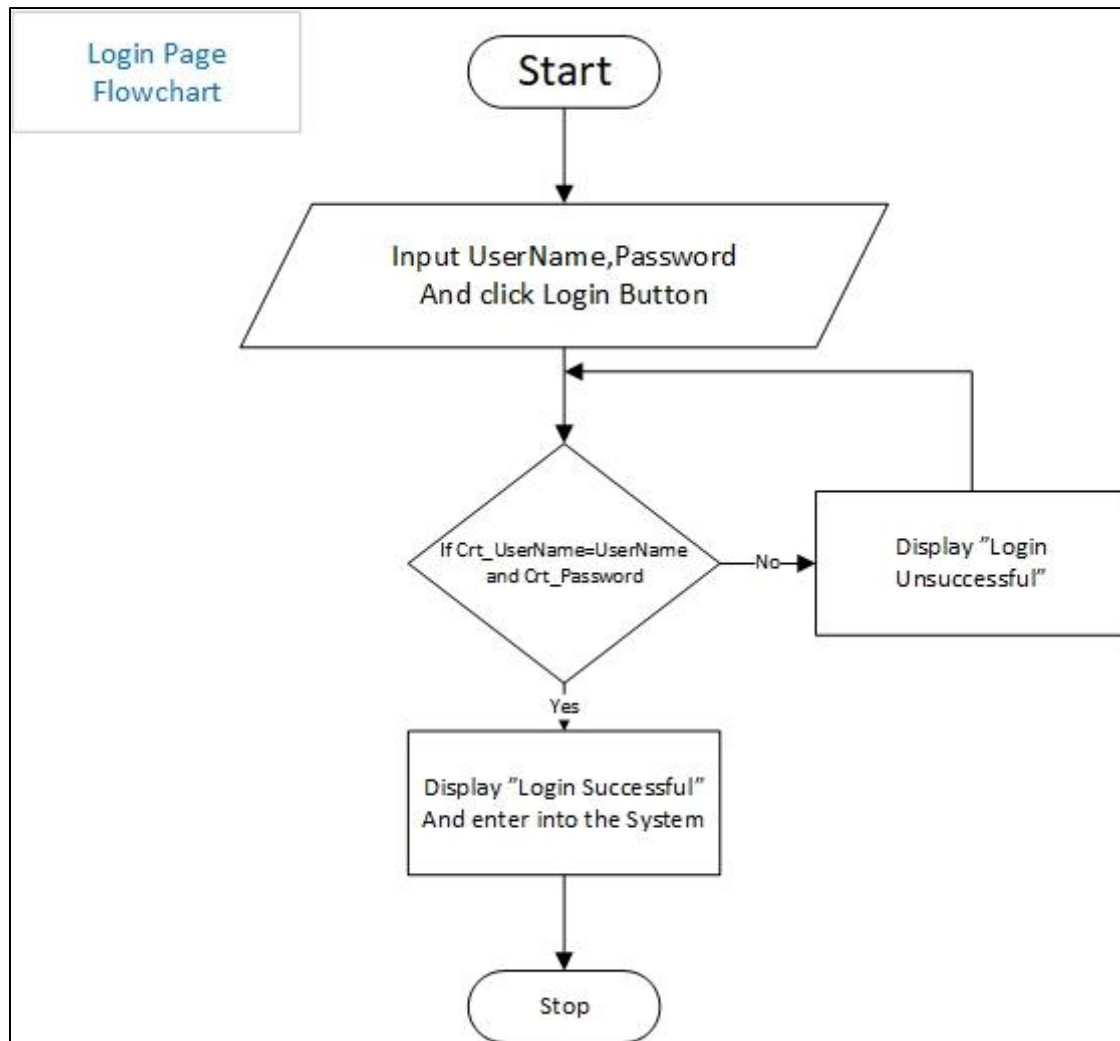


Figure 1:Login Form Flowchart

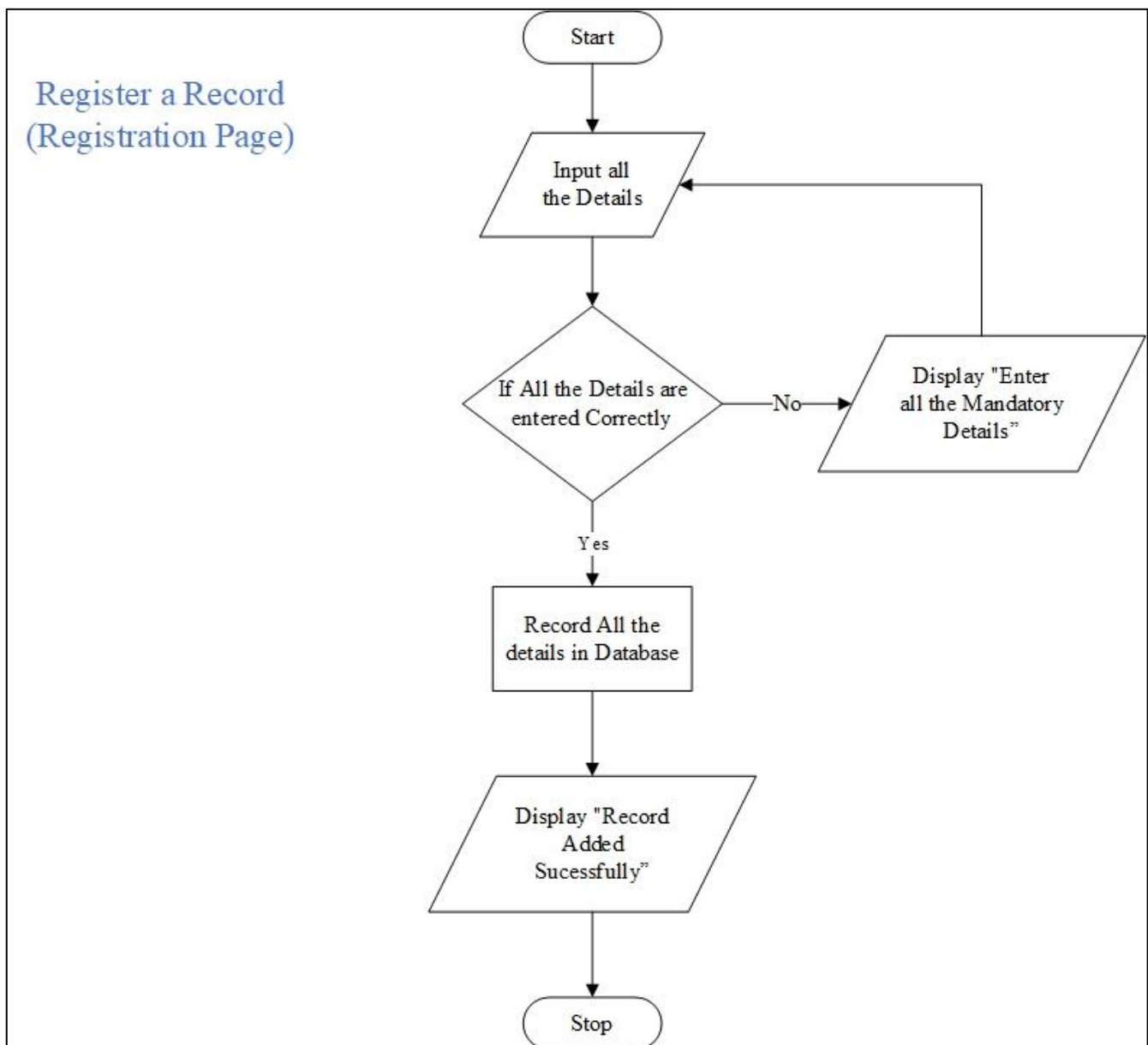


Figure 2: Registration Flowchart

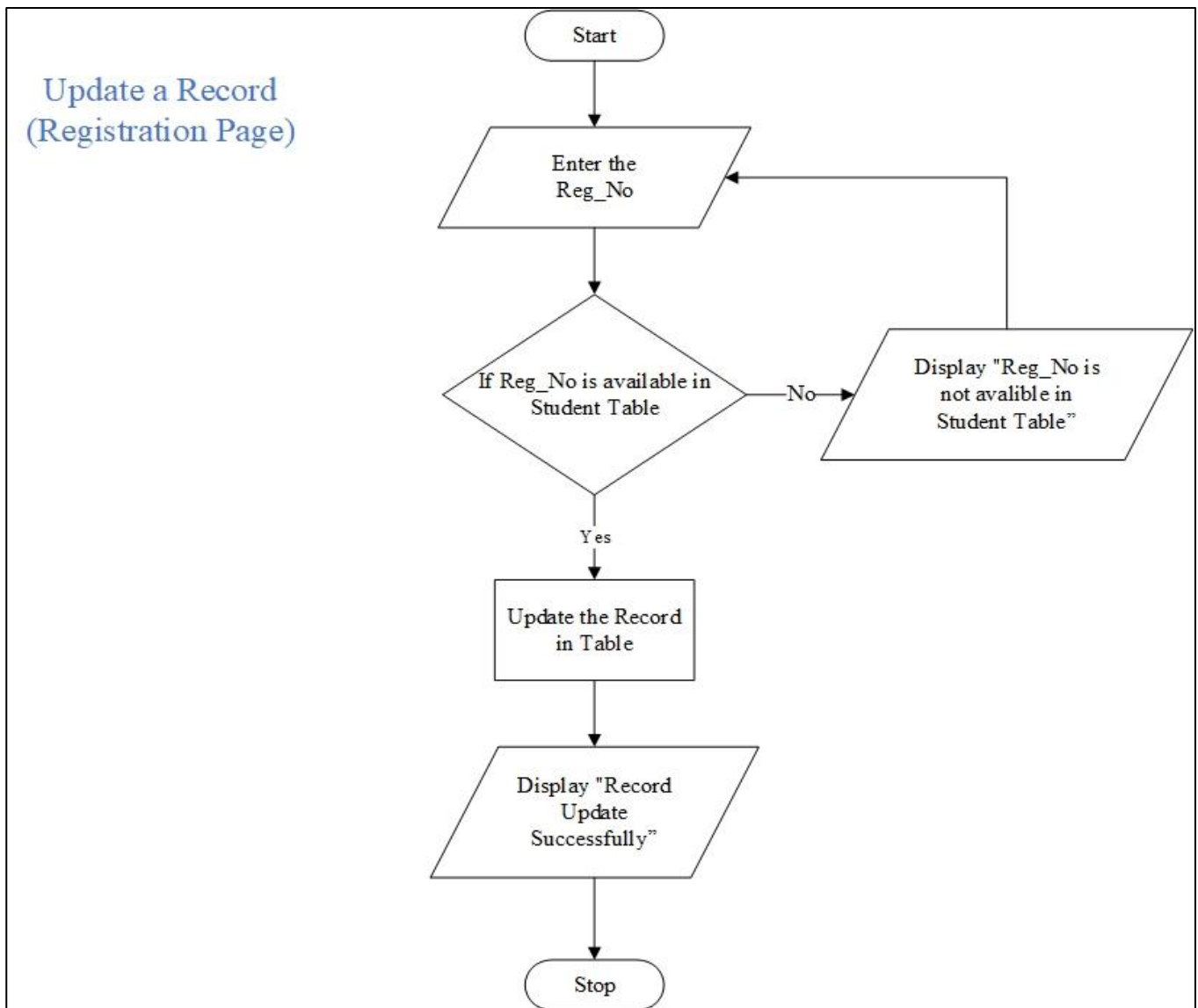


Figure 3:Update Record Flowchart

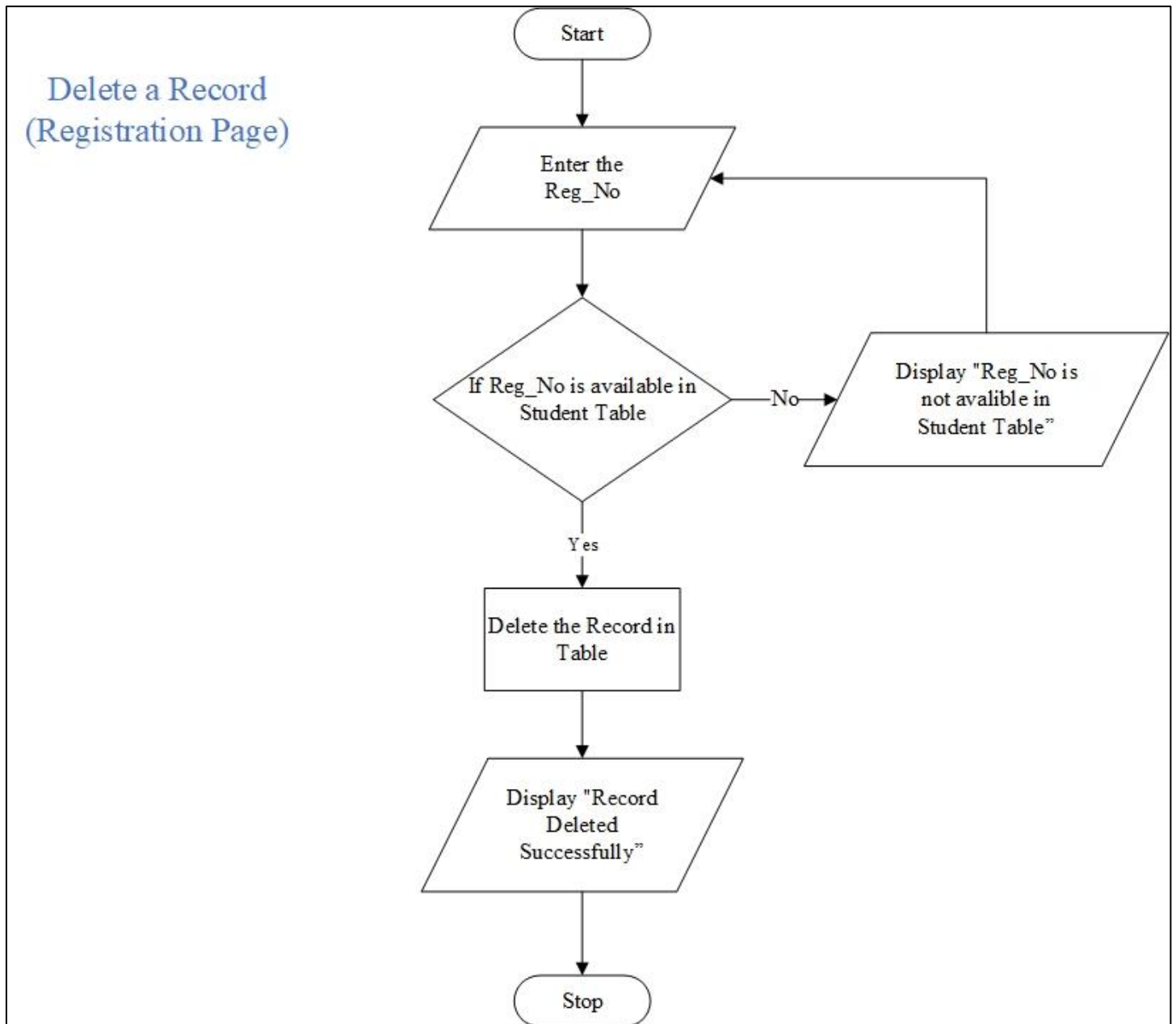


Figure 4:Delete Records Flowchart

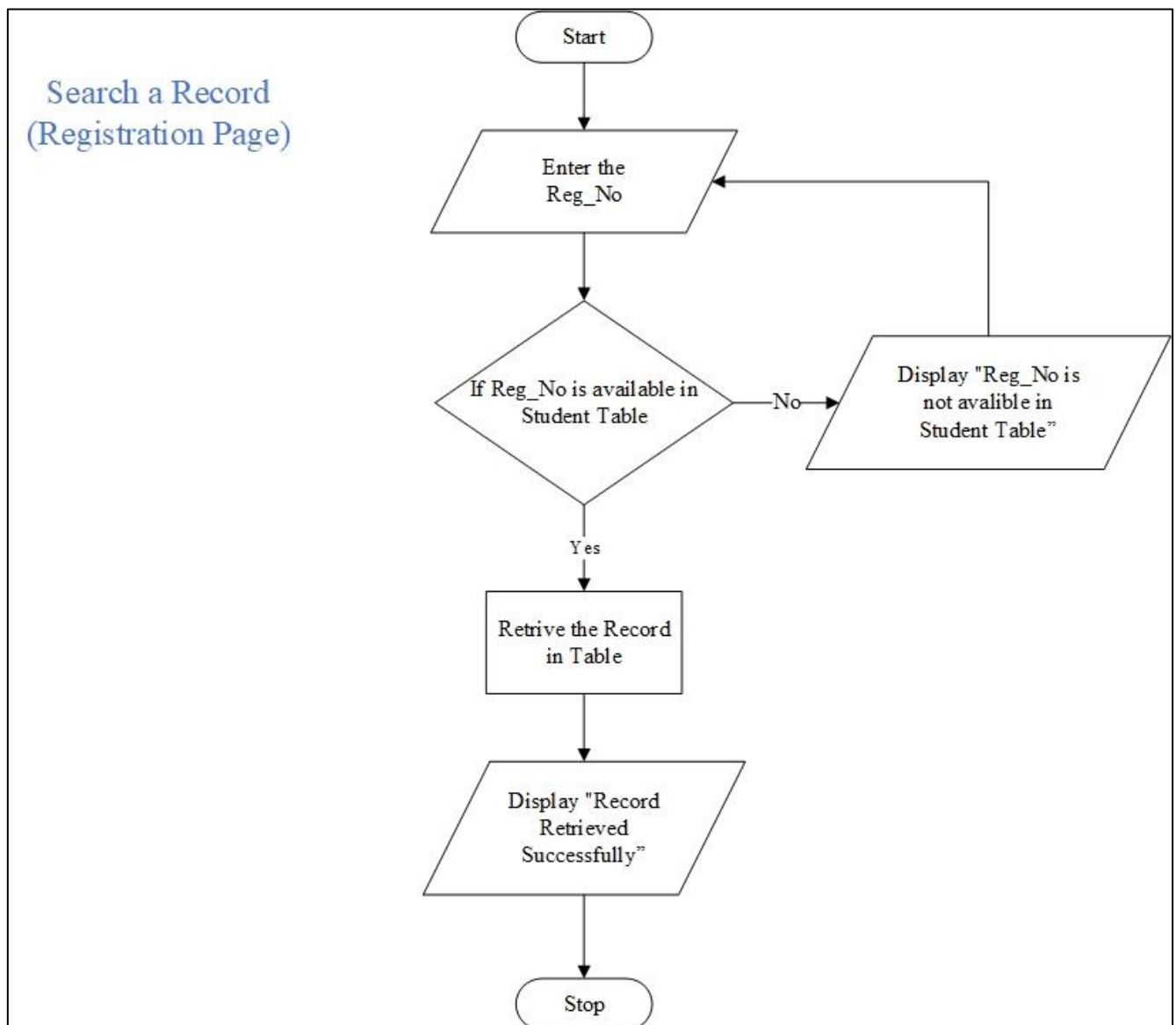


Figure 5: Search Record Flowchart

ER Diagram

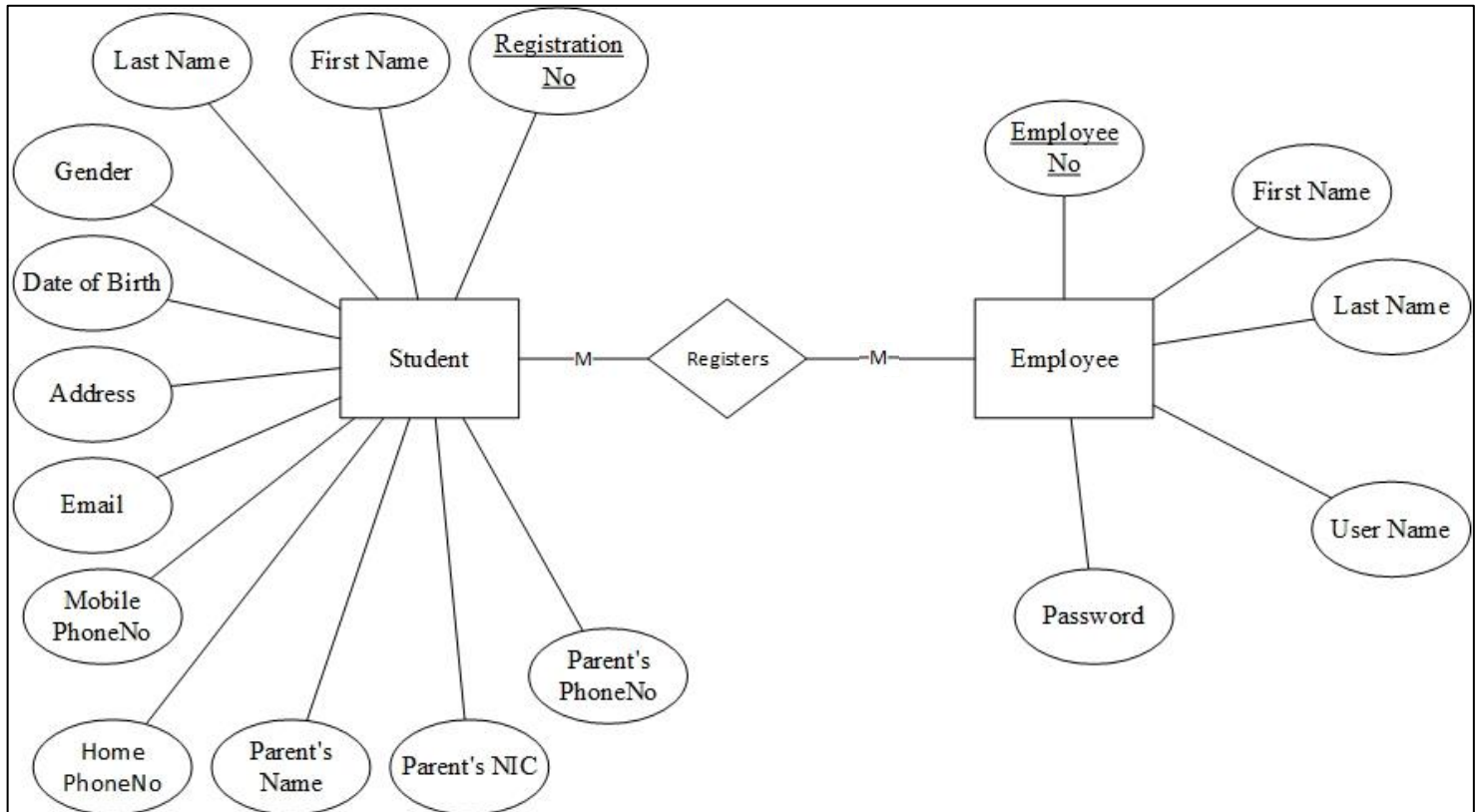


Figure 6:ER Diagram

Illustrative Diagram and Respective Codes

Login Page

Objective: To allow only Authorized people to access into system

Login Page: Illustrative Diagram

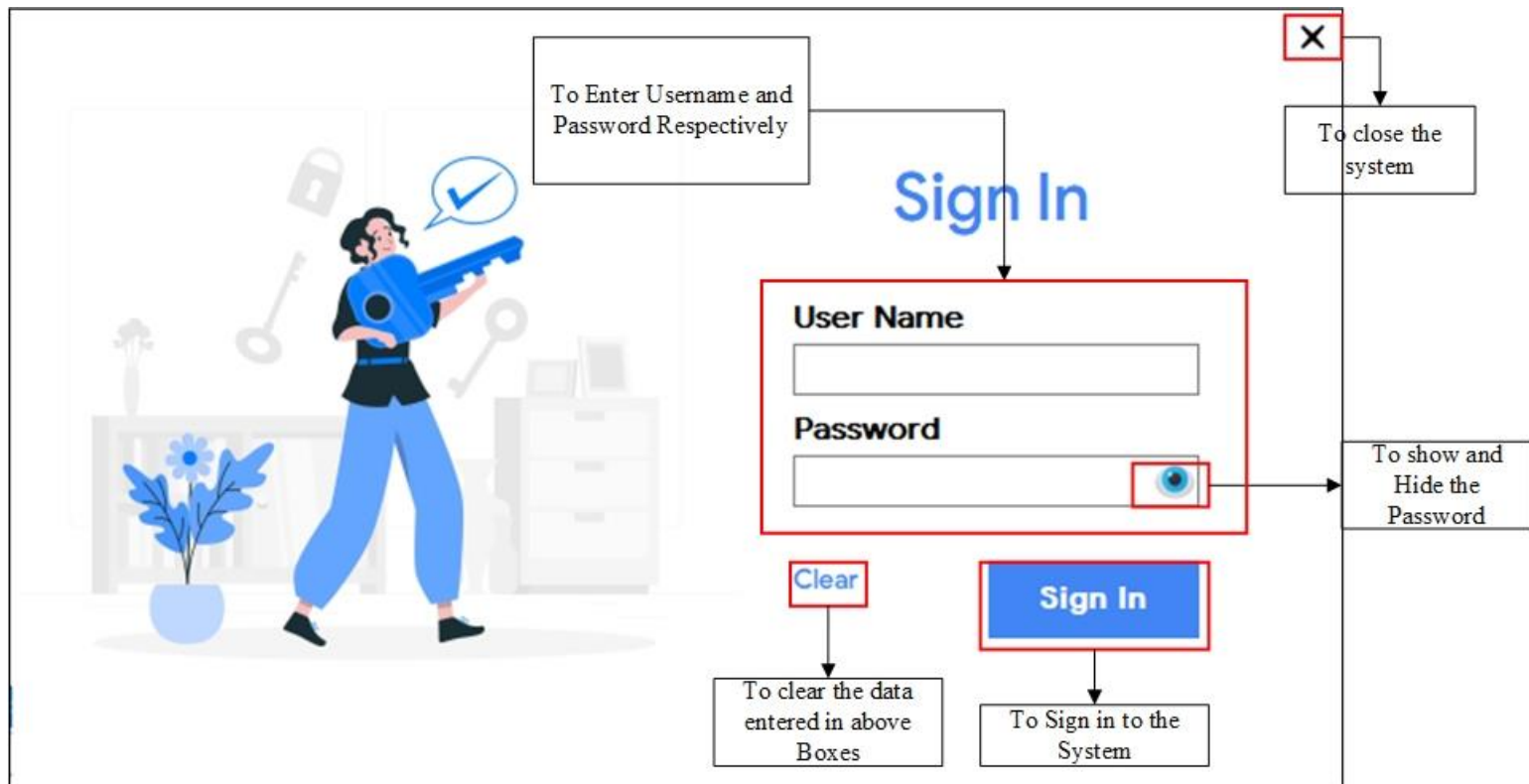


Figure 7: Login Page Illustrative Diagram

Login Page: Code

```
1 reference
private void SignIn_Button_Click(object sender, EventArgs e)
{
    string Entered_Display_Name = UserName_TextBox.Text;
    string Entered_Password = Password_Textbox.Text;
    string connectionString = @"Data Source=Code-Maestro; Initial Catalog = Student; Integrated Security=True;";
    SqlConnection Connect = new SqlConnection(connectionString);
    Connect.Open();
    string search_query = "Select * from employee where Display_Name COLLATE SQL_Latin1_General_CP1_CS_AS = '" + Entered_Display_Name + "' and Password COLLA";
    SqlCommand Fetch = new SqlCommand(search_query, Connect);
    SqlDataReader row = Fetch.ExecuteReader();

    if (row.HasRows)
    {
        //To send user name to DashBoard_Form
        UserName = UserName_TextBox.Text;
        DateTime CTime = DateTime.Now;
        Time = CTime.ToString("HH:mm:ss");
        Hour = CTime.ToString("HH");
        this.Hide();
        Dashboard Main_Page = new Dashboard();
        Main_Page.Show();

        //
    }
    else
    {
        MessageBox.Show("Invalid Login Credentials, \nPlease Try Again", "Invalid Login Details", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Figure 8: Login Page Sign-In Button

```
1 reference
private void Clear_Click(object sender, EventArgs e)
{
    UserName_TextBox.Clear();
    Password_Textbox.Clear();
    UserName_TextBox.Focus();
}
```

Figure 9: Login Page Clear Button

```
1 reference
private void Close_Button_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();
    }
}
```

Figure 10: Login Page Close Button

```
1 reference
private void Hide_Button_Click(object sender, EventArgs e)
{
    Password_Textbox.UseSystemPasswordChar = true;
    Show_Button.BringToFront();
}

1 reference
private void Show_Button_Click(object sender, EventArgs e)
{
    Password_Textbox.UseSystemPasswordChar = false;
    Hide_Button.BringToFront();
}
```

Figure 11: Login Page Password Show/Hide Button

```
1 reference
private void Close_Button_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();
    }
}
```

Figure 12: Login Page Close Button

Dashboard Page

Objective: To easily access Multiple Menu at ease

Dashboard Page: Illustrative Diagram

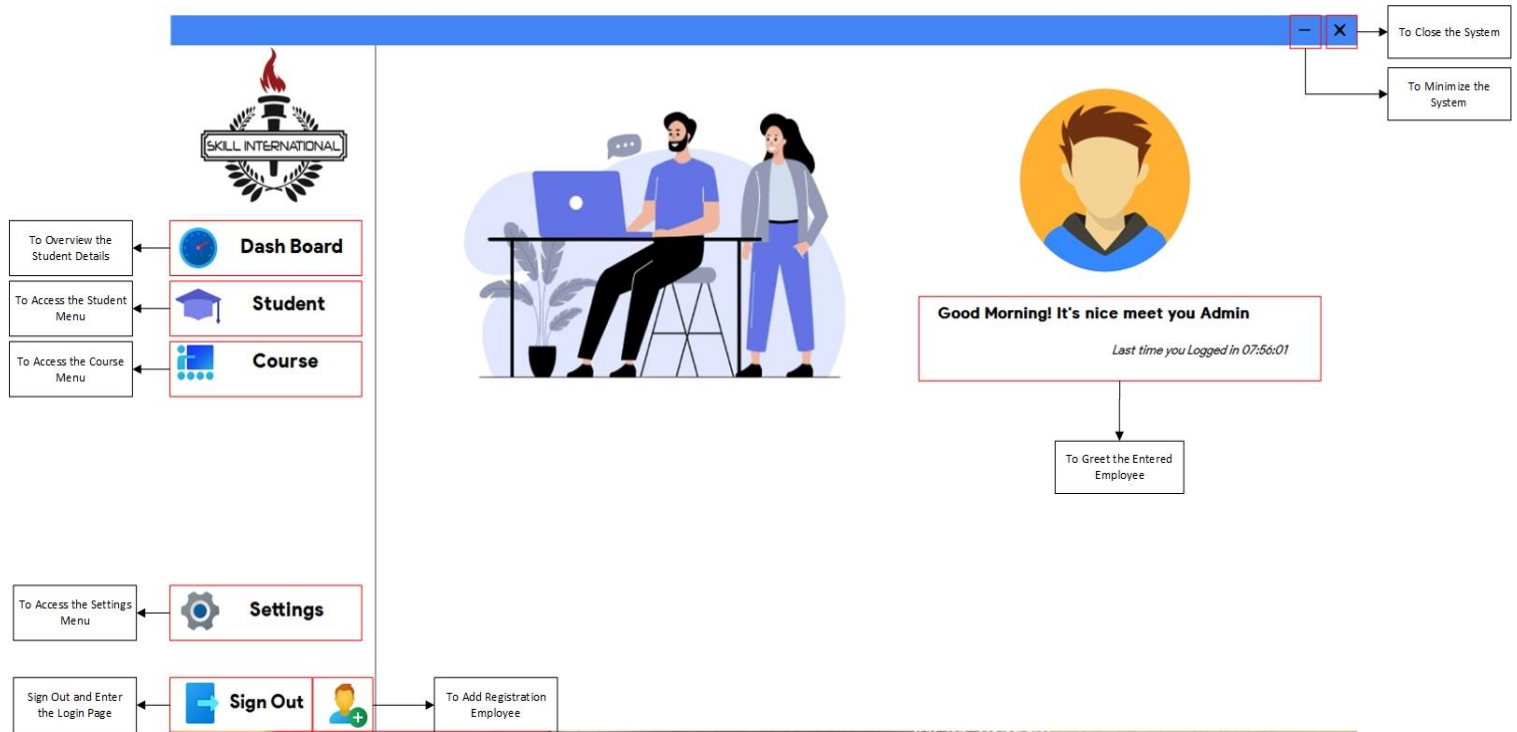


Figure 13: Dashboard Illustrative Diagram

Dashboard Page: Code

```
1 reference
private void Dashboard_Button_Click_1(object sender, EventArgs e)
{
    this.Form_Loader.Controls.Clear();
    Dashboard_Form_Vrb = new Dashboard_Form() { Dock = DockStyle.Fill, TopLevel = false, TopMost = true };
    Dashboard_Form_Vrb.FormBorderStyle = FormBorderStyle.None;
    this.Form_Loader.Controls.Add(Dashboard_Form_Vrb);
    Dashboard_Form_Vrb.Show();
}
```

Figure 14: Dashboard Page Dashboard Menu Button

```
1 reference
private void StudentMenu_Button_Click_1(object sender, EventArgs e)
{
    this.Form_Loader.Controls.Clear();
    StudentMenu_Vrb = new StudentMenu() { Dock = DockStyle.Fill, TopLevel = false, TopMost = true };
    StudentMenu_Vrb.FormBorderStyle = FormBorderStyle.None;
    this.Form_Loader.Controls.Add(StudentMenu_Vrb);
    StudentMenu_Vrb.Show();
}
```

Figure 15: Dashboard Page Student Menu Button

```
1 reference
private void Minimize_Button_Click(object sender, EventArgs e)
{
    WindowState = FormWindowState.Minimized;
}
```

Figure 16: Dashboard Page Minimize Button

```
1 reference
private void SignOut_Button_Click(object sender, EventArgs e)
{
    Login_Form SignUp_Page = new Login_Form();

    var result = MessageBox.Show("Are you sure, Do you really need to Sign Out....?", "Sign Out", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (result == DialogResult.Yes)
    {
        this.Close();
        SignUp_Page.Show();
    }
}
```

Figure 18: Dashboard Page Sign-Out Button

```
1 reference
private void Close_Button_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();
    }
}
```

Figure 17: Dashboard Page Exit Button

Student Menu

Objective: To access all the students details and Student Table Related Functions.

Student Menu: Illustrative Diagram



Figure 19: Student Menu Illustrative Diagram

Student Menu: Code

```
1 reference
private void Registration_Button_Click(object sender, EventArgs e)
{
    Registration_Button.Visible = false;
    Search_Button.Visible = false;
    this.Student_Loader.Controls.Clear();
    Registration_Form Registration_Form_New = new Registration_Form() { Dock = DockStyle.Fill, TopLevel = false, TopMost = true };
    Registration_Form_New.FormBorderStyle = FormBorderStyle.None;
    this.Student_Loader.Controls.Add(Registration_Form_New);
    Registration_Form_New.Show();
}
```

Figure 20: Student Menu Registration Button

```
1 reference
private void Search_Button_Click(object sender, EventArgs e)
{
    Registration_Button.Visible = false;
    Search_Button.Visible = false;
    this.Student_Loader.Controls.Clear();
    Search_Form Search_Form_New = new Search_Form() { Dock = DockStyle.Fill, TopLevel = false, TopMost = true };
    Search_Form_New.FormBorderStyle = FormBorderStyle.None;
    this.Student_Loader.Controls.Add(Search_Form_New);
    Search_Form_New.Show();
}
```

Figure 21: Student Menu Search Button

Registration Page

Objective: To register, update and delete the records of students.

Registration Page: Illustration Diagram

Figure 22: Registration Page Illustrative Diagram

Registration Page: Code

```
1 reference
private void Delete_Button_Click(object sender, EventArgs e)
{
    var result = MessageBox.Show("Are You Sure You Need To Delete This Record...?", "Delete", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (result == DialogResult.Yes)
    {
        string Registration_No = Registration_ComboBox.Text;
        if (string.IsNullOrEmpty(Registration_No))
        {
            MessageBox.Show("Enter a Registration No to Delete the record", "Enter Registration No", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        }
        else
        {
            string Registration_No_Store = Registration_ComboBox.Text;
            string Delete_Query = "DELETE from Registration WHERE Registration_No = " + Registration_No + " ";
            con.Open();
            SqlCommand command = new SqlCommand(Delete_Query, con);
            Command.ExecuteNonQuery();
            SqlCommand command2 = new SqlCommand("Select (Registration_No)from Registration", con);
            SqlDataReader reader;
            reader = command2.ExecuteReader();
            DataTable table = new DataTable();
            table.Columns.Add("Registration_No", typeof(int));
            table.Load(reader);
            Registration_ComboBox.ValueMember = "Registration_No";
            Registration_ComboBox.DataSource = table;
            con.Close();
            Registration_ComboBox.Text = Registration_No_Store.ToString();
            MessageBox.Show("Record Have Been Deleted", "Student Deleted", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}
```

Figure 23: Registration Page Delete Button

```
1 reference
private void Clear_Button_Click(object sender, EventArgs e)
{
    Registration_ComboBox.Text = "";
    FName_TextBox.Clear();
    LName_TextBox.Clear();
    DateTime This_Day = DateTime.Today;
    DOB_Picker.Text = This_Day.ToString();
    Male_RBtn.Checked = false;
    Female_RBtn.Checked = false;
    Address_TextBox.Clear();
    EMail_Textbox.Clear();
    PhoneNo_Textbox.Clear();
    HomeNo_Textbox.Clear();
    PName_Textbox.Clear();
    PNIC_Textbox.Clear();
    PPhoneNo_Textbox.Clear();
    FName_TextBox.Focus();
}
```

Figure 24: Registration Page Clear Button

```

1 reference
private void Search_Button_Click(object sender, EventArgs e)
{
    string Registration_No = Registration_ComboBox.Text;

    if (string.IsNullOrEmpty(Registration_No))
    {
        MessageBox.Show("Please enter a registration number.", "Search Record", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        return;
    }

    using (SqlConnection connection = new SqlConnection(@"Data Source=Code-Maestro;Initial Catalog=Student;Integrated Security=True; "))
    {
        connection.Open();

        string query = "SELECT * FROM Registration WHERE Registration_No = @Registration_No";
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@Registration_No", Registration_No);

            using (SqlDataReader reader = command.ExecuteReader())
            {
                if (reader.Read())
                {
                    Registration_ComboBox.Text = reader["Registration_No"].ToString();
                    FName_TextBox.Text = reader["First_Name"].ToString();
                    LName_TextBox.Text = reader["Last_Name"].ToString();
                    DOB_Picker.Format = DateTimePickerFormat.Custom;
                    DOB_Picker.CustomFormat = "yyyy/MM/dd";
                    DOB_Picker.Text = reader["Date_Of_Birth"].ToString();
                    if (reader["Gender"].ToString() == "Male")
                    {

```

Figure 25: Registration Page Search Button 1

```

                    DOB_Picker.CustomFormat = "yyyy/MM/dd";
                    DOB_Picker.Text = reader["Date_Of_Birth"].ToString();
                    if (reader["Gender"].ToString() == "Male")
                    {
                        Male_RBtn.Checked = true;
                    }
                    else
                    {
                        Female_RBtn.Checked = true;
                    }

                    Address_TextBox.Text = reader["Address"].ToString();
                    Email_Textbox.Text = reader["Email"].ToString();
                    PhoneNo_Textbox.Text = reader["MobilePhoneNo"].ToString();
                    HomeNo_Textbox.Text = reader["HomePhoneNo"].ToString();
                    PName_Textbox.Text = reader["Parent_Name"].ToString();
                    PNIC_Textbox.Text = reader["Parent_NIC"].ToString();
                    PPhoneNo_Textbox.Text = reader["Parent_ContactNo"].ToString();
                }
            }
        }
    }
}

```

Figure 26: Registration Page Search Button 2

```
string Query_update = "UPDATE Registration SET " +
    "First_Name = '" + First_Name + "', " +
    "Last_Name = '" + Last_Name + "', " +
    "Date_Of_Birth = '" + Date_Of_Birth + "', " +
    "Gender = '" + Gender + "', " +
    "Address = '" + Address + "', " +
    "Email = '" + Email + "', " +
    "MobilePhoneNo = '" + MobilePhoneNo + "', " +
    "HomePhoneNo = '" + HomePhoneNo + "', " +
    "Parent_Name = '" + Parent_Name + "', " +
    "Parent_NIC = '" + Parent_NIC + "', " +
    "Parent_ContactNo = '" + Parent_ContactNo + "' " +
    "WHERE Registration_No = '" + Registration_No + "'";
if (string.IsNullOrEmpty(Registration_No))
{
    MessageBox.Show("Enter a Registration No to update the record", "Enter Registration No", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}
else
{
    if (First_Name == "" || Last_Name == "" || Date_Of_Birth == "" || Gender == "" || Address == "" || MobilePhoneNo == "" || Parent_Name == "" || Parent_NI
    {
        MessageBox.Show("Enter All The Mandatory Details", "Enter Details", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
    else
    {
        con.Open();
        SqlCommand Update = new SqlCommand(Query_update, con);
        Update.ExecuteNonQuery();
        con.Close();
        MessageBox.Show("Record Updated Successfully", "Update Student", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
```

Figure 27: Registration Page Update Button 1

```
1 reference
private void Update_Button_Click(object sender, EventArgs e)
{
    try
    {
        string Registration_No = Registration_ComboBox.Text;
        string First_Name = FName_TextBox.Text;
        string Last_Name = LName_TextBox.Text;
        DOB_Picker.Format = DateTimePickerFormat.Custom;
        DOB_Picker.CustomFormat = "yyyy/MM/dd";
        string Date_Of_Birth = DOB_Picker.Text;
        string Gender;
        if (Male_RBbtn.Checked)
        {
            Gender = "Male";
        }
        else if (Female_RBbtn.Checked)
        {
            Gender = "Female";
        }
        else
        {
            Gender = "";
        }

        string Address = Address_TextBox.Text;
        string Email = EMail_Textbox.Text;
        string MobilePhoneNo = PhoneNo_Textbox.Text;
        string HomePhoneNo = HomeNo_Textbox.Text;
        string Parent_Name = PName_Textbox.Text;
        string Parent_NIC = PNIC_Textbox.Text;
        string Parent_ContactNo = PPhoneNo_Textbox.Text;
```

Figure 28: Registration Page Update Button 2

Figure 29: Registration Page Register Button 1

Figure 30: Registration Page Register Button 2

Figure 31: Registration Page Register Button 3

```

//-----
// Disable IDENTITY_INSERT
string disableIdentityInsertQuery = $"SET IDENTITY_INSERT Registration OFF;";
using (SqlCommand disableIdentityInsertCommand = new SqlCommand(disableIdentityInsertQuery, con))
{
    disableIdentityInsertCommand.ExecuteNonQuery();
}
string Registration_No_Store = Registration_ComboBox.Text;
SqlCommand command = new SqlCommand("Select (Registration_No)from Registration", con);
SqlDataReader reader;
reader = command.ExecuteReader();
DataTable table = new DataTable();
table.Columns.Add("Registration_No", typeof(int));
table.Load(reader);
Registration_ComboBox.ValueMember = "Registration_NO";
Registration_ComboBox.DataSource = table;
con.Close();
Registration_ComboBox.Text = Registration_No_Store.ToString();
}

catch (SqlException ex)
{
    string message = "Insert Error:";
    message += ex.Message;
    MessageBox.Show(message);
    con.Close();
}
}

```

Figure 32: Registration Page Register Button 4

Search Page

Objective: To View the Student's Table

Search Page: Illustration Page

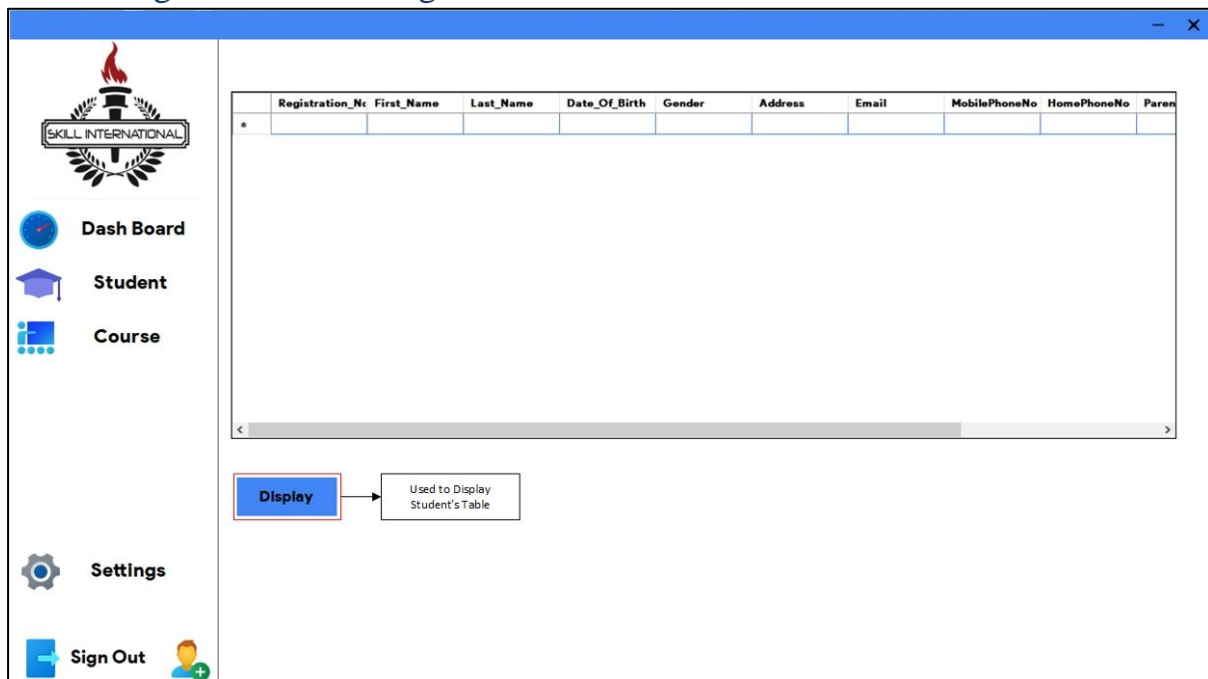


Figure 33: Search Page Illustrative Diagram

Search Page: Code

```
1 reference
private void Display_Click(object sender, EventArgs e)
{
    try
    {
        using (SqlConnection sqlCon = new SqlConnection(connectionString))
        {
            sqlCon.Open();
            SqlDataAdapter sqlDa = new SqlDataAdapter("SELECT * FROM Registration", sqlCon);
            DataTable dtbl = new DataTable();
            sqlDa.Fill(dtbl);

            Student_DGV.DataSource = dtbl;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error: " + ex.Message);
    }
}
```

Figure 34: Search Page Display Button

Add Member Page

Objective: Ability to Add Staff Members to Student Management System.

Add Member Page: Illustration Diagram

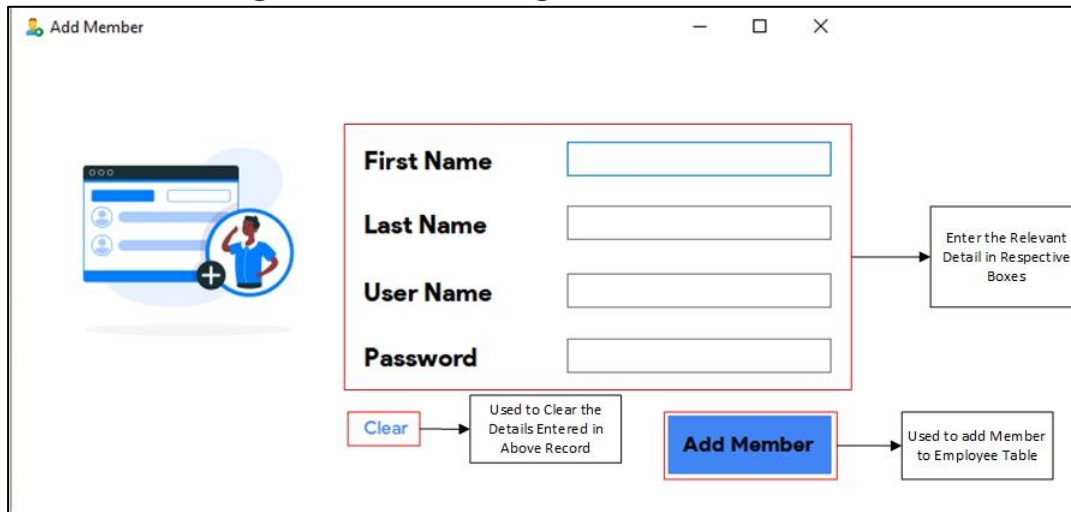


Figure 35: Add Member Page Illustrative Diagram

Add Member Page: Code

```
1 reference
private void Add_Button_Click(object sender, EventArgs e)
{
    string First_Name = FirstName_TextBox.Text;
    string Last_Name = LastName_TextBox.Text;
    string User_Name = UserName_Textbox.Text;
    string Password = Password_Textbox.Text;
    if (First_Name == "" || Last_Name == "" || User_Name == "" || Password == "")
    {
        MessageBox.Show("Enter All The Details", "Enter Details", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
    else
    {
        string Query_Add = "Insert into Employee ( First_Name, Last_Name, Display_Name , Password)" +
            "Values('" + First_Name + "','" + Last_Name + "','" + User_Name + "','" + Password + "')";
        con.Open();
        SqlCommand Add = new SqlCommand(Query_Add, con);
        Add.ExecuteNonQuery();
        MessageBox.Show("Record Added Successfully", "Add Employee", MessageBoxButtons.OK, MessageBoxIcon.Information);
        con.Close();
    }
}
```

Figure 36: Add Member Page Add Button

```
1 reference
private void Clear_Click(object sender, EventArgs e)
{
    FirstName_TextBox.Clear();
    LastName_TextBox.Clear();
    UserName_Textbox.Clear();
    Password_Textbox.Clear();
    FirstName_TextBox.Focus();
}
```

Figure 37: Add Member Page Clear Button

Development Process

Why did we used Visual Studio?

To build the Student Management System our team members have used Visual Studio because it has support to multiple programming language and supports drag and drop feature to create an GUI windows form with less effort.

Testing Process

The System have been tested and received response in form of messages

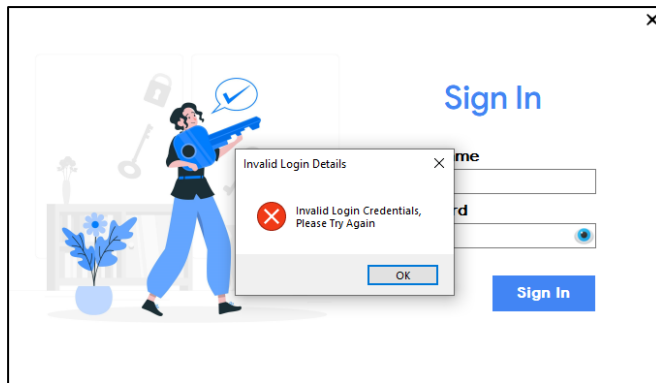


Figure 38: Login Page invalid Credentials

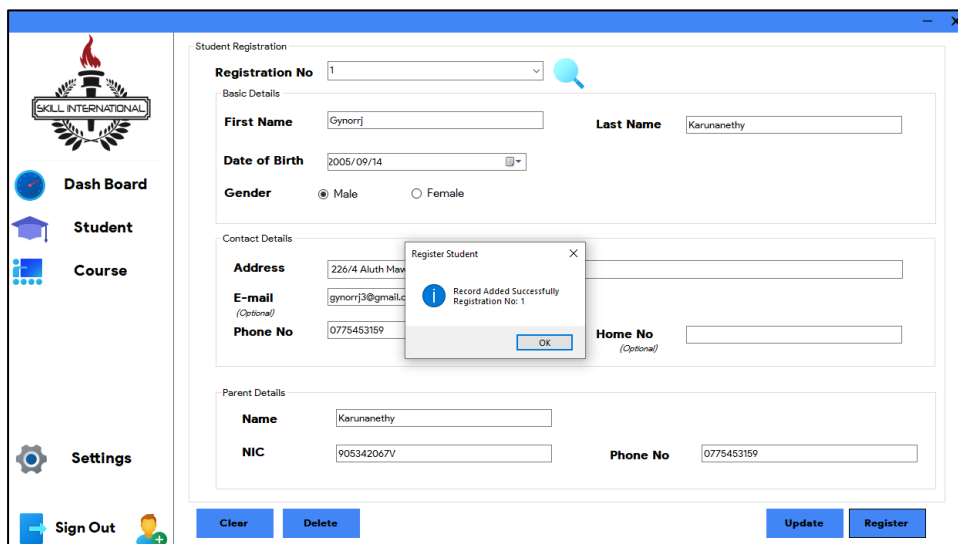


Figure 39: Registration Successful

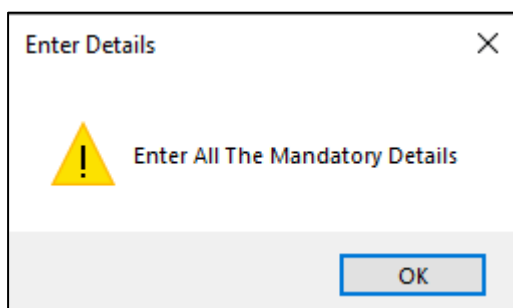


Figure 40: Registration was unsuccessful

The screenshot shows the 'Student Registration' form with a sidebar on the left containing links for 'Dash Board', 'Student', 'Course', 'Settings', and 'Sign Out'. The form is divided into three sections: 'Basic Details', 'Contact Details', and 'Parent Details'. In the 'Basic Details' section, 'First Name' is 'Gynorj' and 'Last Name' is 'Karunanethy'. 'Date of Birth' is '2005/09/14' and 'Gender' is 'Male'. In the 'Contact Details' section, 'Address' is '226/4 Aluth Mawatha Road Colombo -15', 'E-mail' is 'gynorj3@gmail.com', 'Phone No' is '775453159', and 'Home No' is '0'. In the 'Parent Details' section, 'Name' is 'Karunanethy', 'NIC' is '905342067V', and 'Phone No' is '775453159'. A modal dialog titled 'Update Student' is displayed in the center, showing a message 'Record Updated Successfully' and an 'OK' button. At the bottom of the form, there are buttons for 'Clear', 'Delete', 'Update', and 'Register'.

Figure 41: Update was Successful

The screenshot shows the 'Student Registration' form with the same sidebar and form sections as Figure 41. A modal dialog titled 'Student Deleted' is displayed in the center, showing a message 'Record Have Been Deleted' and an 'OK' button. The 'Update' button at the bottom of the form is now disabled, and the 'Delete' button is active.

Figure 42:Record Deleted Successful

The screenshot shows the 'Student Registration' form with the same sidebar and form sections. The 'Registration No' dropdown is set to '1'. The 'Basic Details' section shows 'First Name' as 'Gynorj' and 'Last Name' as 'Karunanethy'. The 'Date of Birth' is '2005/09/14' and 'Gender' is 'Male'. The 'Contact Details' section shows 'Address' as '226/4 Aluth Mawatha Road Colombo -15', 'E-mail' as 'gynorj3@gmail.com', 'Phone No' as '775453159', and 'Home No' as '0'. The 'Parent Details' section shows 'Name' as 'Karunanethy', 'NIC' as '905342067V', and 'Phone No' as '775453159'. The 'Update' and 'Register' buttons at the bottom are active.

Figure 43:Searching record is Successful

Registration_No	First_Name	Last_Name	Date_Of_Birth	Gender	Address	Email	MobilePhoneNo	HomePhoneNo	Parent
*									

Display

Figure 44: Searching Table is Successful

Add Member

First Name: Gynorj

Last Name: Gynanethy

Email: gynorj

Address: englers

Clear

Add Member

Add Employee

Record Added Successfully

OK

Figure 45: Adding Member is Successful

Future Implementation

1. Better Login Portal:

- Better Login Portal with restricted entry for each role (Principal, Teachers and Registers).

2. Access More Option:

- Ability to access more option like Print, Filter Option in Database View.

3. More Menus:

- Ability to access more Menus like Setting and Course.

4. More Customizations:

- Ability to change the Font Size, Font Color.
- The Ability to Switch between Light and Night Mode.

5. Del_RegistrationNo Table:

- Additional Table to record all the deleted Registration no (in case if the record is deleted that registration number will be empty so to reuse it, we have to implement another table for it.)