**C# and .NET Frameworks**

**Assignment 2**

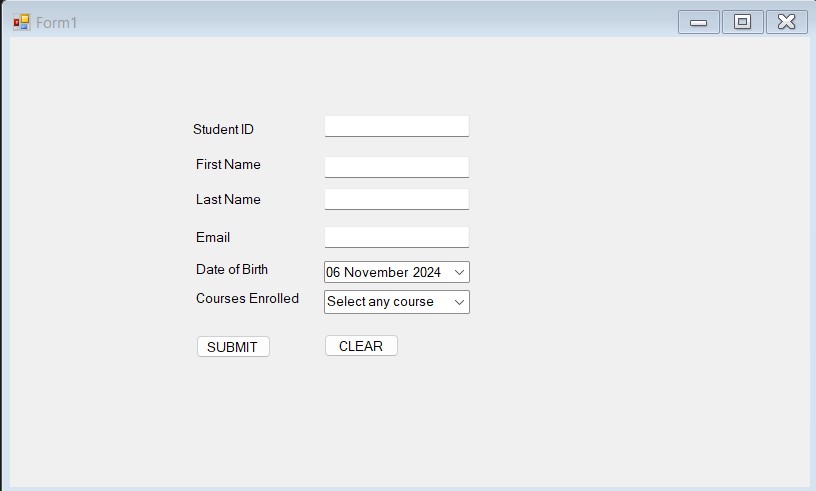
1. Design and implement a **Student Registration** form using C# and Windows Forms.

The form should allow users to input and save student details into a database.

**Requirements:**

1. The registration form should include the following fields:
   * **Student ID** (Auto-generated or entered manually)
   * **First Name** o **Last Name**
   * **Date of Birth** o **Email** o **Phone Number** o **Course Enrolled**
2. Implement the following features:
   * **Form Validation**: Ensure that all required fields (e.g., Student ID, First Name, Last Name, Email) are properly validated (e.g., email format, required fields, etc.).
   * **Save Data**: Connect the form to a database (SQL Server, MySQL, or any other relational database of your choice) using ADO.NET to save the student data.
   * **Reset Form**: Provide a "Clear" button to reset all the input fields.
   * **Display Students**: Optionally, include a DataGridView control to display all registered students after submission.

**FORM AND CODE:**



**Step 1: Set up the Database**

CREATE DATABASE StudentDB;

USE StudentDB;

CREATE TABLE Students (

StudentID INT PRIMARY KEY IDENTITY(1,1),

FirstName NVARCHAR(50) NOT NULL,

LastName NVARCHAR(50) NOT NULL,

DateOfBirth DATE,

Email NVARCHAR(100) NOT NULL,

PhoneNumber NVARCHAR(20),

CourseEnrolled NVARCHAR(100)

);

**Step 2: Windows Forms Application Code**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace StudentRegistrationApp

{

public partial class StudentRegistrationForm : Form

{

// Connection string to SQL Server database

private readonly string connectionString = "Server=YOUR\_SERVER\_NAME;Database=StudentDB;Integrated Security=True;";

public StudentRegistrationForm()

{

InitializeComponent();

LoadStudentData();

}

private void btnSave\_Click(object sender, EventArgs e)

{

if (ValidateForm())

{

SaveStudentData();

ClearForm();

LoadStudentData();

}

}

private void btnClear\_Click(object sender, EventArgs e)

{

ClearForm();

}

private bool ValidateForm()

{

if (string.IsNullOrWhiteSpace(txtFirstName.Text) ||

string.IsNullOrWhiteSpace(txtLastName.Text) ||

string.IsNullOrWhiteSpace(txtEmail.Text))

{

MessageBox.Show("Please fill in all required fields.", "Validation Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

if (!IsValidEmail(txtEmail.Text))

{

MessageBox.Show("Please enter a valid email address.", "Validation Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;

}

return true;

}

private bool IsValidEmail(string email)

{

try

{

var addr = new System.Net.Mail.MailAddress(email);

return addr.Address == email;

}

catch

{

return false;

}

}

private void SaveStudentData()

{

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

string query = "INSERT INTO Students (FirstName, LastName, DateOfBirth, Email, PhoneNumber, CourseEnrolled) " +

"VALUES (@FirstName, @LastName, @DateOfBirth, @Email, @PhoneNumber, @CourseEnrolled)";

using (SqlCommand cmd = new SqlCommand(query, conn))

{

cmd.Parameters.AddWithValue("@FirstName", txtFirstName.Text);

cmd.Parameters.AddWithValue("@LastName", txtLastName.Text);

cmd.Parameters.AddWithValue("@DateOfBirth", dtpDateOfBirth.Value.Date);

cmd.Parameters.AddWithValue("@Email", txtEmail.Text);

cmd.Parameters.AddWithValue("@PhoneNumber", txtPhoneNumber.Text);

cmd.Parameters.AddWithValue("@CourseEnrolled", txtCourse.Text);

cmd.ExecuteNonQuery();

}

}

}

private void LoadStudentData()

{

using (SqlConnection conn = new SqlConnection(connectionString))

{

conn.Open();

string query = "SELECT \* FROM Students";

SqlDataAdapter adapter = new SqlDataAdapter(query, conn);

DataTable dt = new DataTable();

adapter.Fill(dt);

dataGridViewStudents.DataSource = dt;

}

}

private void ClearForm()

{

txtFirstName.Clear();

txtLastName.Clear();

txtEmail.Clear();

txtPhoneNumber.Clear();

txtCourse.Clear();

dtpDateOfBirth.Value = DateTime.Now;

}

}

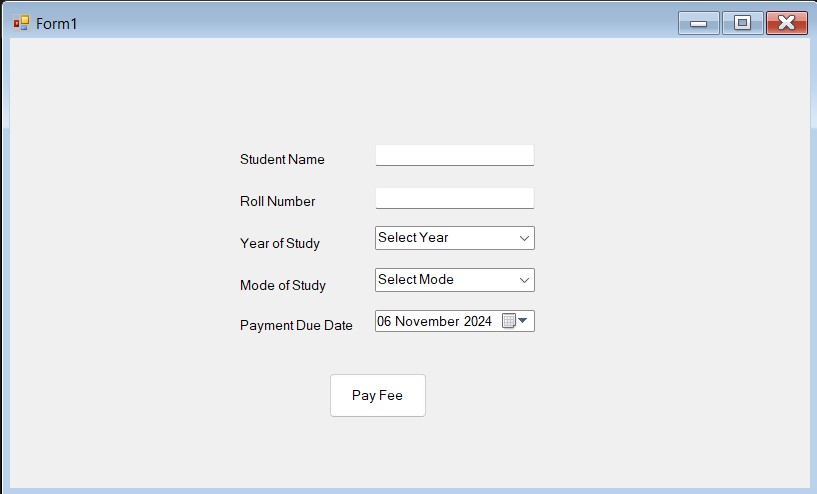
}

2.Design and implement a **Student Fee Payment System** using C# and Windows Forms. The application should allow students to enter their details, pay their fees, and generate a bill with a unique bill number.

**Requirements:**

* 1. **Form Design**:
     + The form should include the following fields:
       - **Student Name**
       - **Roll Number**
       - **Year of Study** (Dropdown or input field)
       - **Hostel/Day Scholar** (Radio buttons or dropdown selection)
       - **Due Date for Fee Payment** (Date Picker)
  2. **Fee Payment and Bill Generation**:
     + When the student fills in the required details and clicks the "Pay Fee" button:
       - **Calculate the total fees** based on whether the student is a **Hostel Resident** or a **Day Scholar** (apply appropriate fees based on your assumption).
       - If the payment is made **after the due date**, apply a **late fee of Rs. 100 per day**.
       - Generate a **Bill** with a unique bill number, and display it along with the student's details (name, roll number, year of study, total fees, and any late fees applied).

**FORM AND CODE:**



**Implement StudentFeePaymentForm.cs:**

using System;

using System.Windows.Forms;

namespace StudentFeePaymentSystem

{

public partial class FeePaymentForm : Form

{

const decimal HostelFee = 5000;

const decimal DayScholarFee = 3000;

const decimal LateFeePerDay = 100;

public FeePaymentForm()

{

InitializeComponent();

}

private void FeePaymentForm\_Load(object sender, EventArgs e)

{

cmbYearOfStudy.Items.Add("1st Year");

cmbYearOfStudy.Items.Add("2nd Year");

cmbYearOfStudy.Items.Add("3rd Year");

cmbYearOfStudy.Items.Add("4th Year");

cmbYearOfStudy.SelectedIndex = 0;

dtpDueDate.Value = DateTime.Now;

}

private void btnPayFee\_Click(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(txtStudentName.Text) ||

string.IsNullOrWhiteSpace(txtRollNumber.Text) ||

cmbYearOfStudy.SelectedIndex == -1 ||

(!rbtnHostel.Checked && !rbtnDayScholar.Checked))

{

MessageBox.Show("Please fill all the details.");

return;

}

string studentName = txtStudentName.Text;

string rollNumber = txtRollNumber.Text;

string yearOfStudy = cmbYearOfStudy.SelectedItem.ToString();

bool isHostelResident = rbtnHostel.Checked;

DateTime dueDate = dtpDueDate.Value.Date;

DateTime currentDate = DateTime.Now.Date;

decimal totalFee = isHostelResident ? HostelFee : DayScholarFee;

decimal lateFee = 0;

if (currentDate > dueDate)

{

int lateDays = (currentDate - dueDate).Days;

lateFee = lateDays \* LateFeePerDay;

}

// Calculate the final amount

decimal totalAmount = totalFee + lateFee;

// Generate a unique bill number (using a simple GUID approach)

string billNumber = Guid.NewGuid().ToString().Substring(0, 8); // First 8 characters of GUID

// Display the Bill details in the label

lblBillDetails.Text = $"Bill Number: {billNumber}\n" +

$"Student Name: {studentName}\n" +

$"Roll Number: {rollNumber}\n" +

$"Year of Study: {yearOfStudy}\n" +

$"Total Fee: Rs. {totalFee}\n" +

$"Late Fee: Rs. {lateFee}\n" +

$"Total Amount: Rs. {totalAmount}";

// Optionally clear the form fields after payment

txtStudentName.Clear();

txtRollNumber.Clear();

cmbYearOfStudy.SelectedIndex = 0;

rbtnHostel.Checked = false;

rbtnDayScholar.Checked = false;

dtpDueDate.Value = DateTime.Now;

}

}

}

3.Design and implement a **Web Service** using C# and ASP.NET to expose functionality for client applications to consume. The web service should provide a specific set of operations, such as retrieving data or performing a calculation.

**Implementation Steps:**

**Define the Service Requirements**:

* + - Identify the functionality that the web service will provide (e.g., retrieving student information, performing fee calculations, etc.). o Specify the input parameters and return types for each operation that the web service will expose.

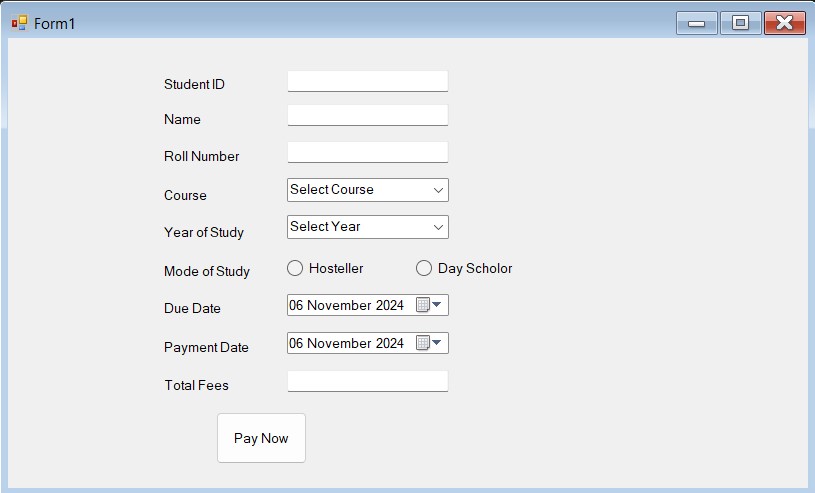
**Create the Web Service**:

* + - In Visual Studio, create a new **ASP.NET Web Service** project.
    - Define a service class by inheriting from System.Web.Services.WebService.
    - Decorate the class with the [WebService] attribute and each method with the

[WebMethod] attribute to expose them as web service operations.

* + - Implement the required service methods (e.g., retrieving student data or calculating fees).

**FORM AND CODE:**



**Implementation of Retrieving student information:**

using System;

using StudentFeeWebService; // Reference to the web service namespace

class Program

{

static void Main(string[] args)

{

// Create an instance of the web service

var service = new StudentFeeService();

// Get student information

string rollNumber = "1001"; // Example roll number

var student = service.GetStudentInfo(rollNumber);

if (student != null)

{

Console.WriteLine("Student Name: " + student.Name);

Console.WriteLine("Year of Study: " + student.YearOfStudy);

}

else

{

Console.WriteLine("Student not found.");

}

// Calculate fee

decimal fee = service.CalculateFee(rollNumber);

if (fee != -1)

{

Console.WriteLine("Total Fee: Rs. " + fee);

}

else

{

Console.WriteLine("Student not found.");

}

Console.ReadLine();

}

}

**Implementation of Calculating student fees:**

using System;

using System.Web.Services;

namespace StudentFeeWebService

{

public class StudentFeeService : WebService

{

// Constants for fees

private const decimal HostelFee = 5000;

private const decimal DayScholarFee = 3000;

private const decimal LateFeePerDay = 100;

private static readonly Student[] Students = new Student[]

{

new Student { RollNumber = "1001", Name = "John Doe", YearOfStudy = "1st Year", IsHostelResident = true, DueDate = new DateTime(2024, 9, 30) },

new Student { RollNumber = "1002", Name = "Jane Smith", YearOfStudy = "2nd Year", IsHostelResident = false, DueDate = new DateTime(2024, 9, 30) }

};

public Student GetStudentInfo(string rollNumber)

{

// Find student by roll number

foreach (var student in Students)

{

if (student.RollNumber == rollNumber)

{

return student;

}

}

// If not found, return null (or can throw an exception)

return null;

}

public decimal CalculateFee(string rollNumber)

{

// Find student by roll number

foreach (var student in Students)

{

if (student.RollNumber == rollNumber)

{

// Calculate total fee based on whether the student is a hostel resident or day scholar

decimal totalFee = student.IsHostelResident ? HostelFee : DayScholarFee;

// Check if the fee payment is overdue, apply late fee if necessary

DateTime currentDate = DateTime.Now;

if (currentDate > student.DueDate)

{

int lateDays = (currentDate - student.DueDate).Days;

totalFee += lateDays \* LateFeePerDay;

}

return totalFee;

}

}

// If student is not found, return -1 or throw an exception

return -1;

}

}

// Define a Student class for demonstration purposes

public class Student

{

public string RollNumber { get; set; }

public string Name { get; set; }

public string YearOfStudy { get; set; }

public bool IsHostelResident { get; set; }

public DateTime DueDate { get; set; }

}

}

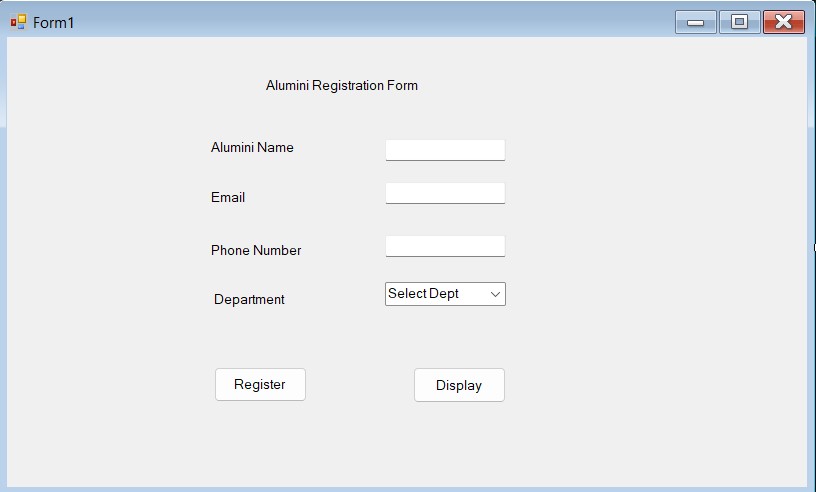
4.Our college is organizing an **Alumni Meet** on **May 5, 2024**. The alumni cell is in the process of creating a database to store a list of registered alumni who will attend the event.

You are tasked with designing a registration form and implementing it using ADO.NET.

**Requirements:**

* 1. **Design the Registration Form**:
     + Create a Windows Forms application that includes the following controls:
       - **TextBox** for entering the **Alumni Name**
       - **TextBox** for entering the **Email**
       - **TextBox** for entering the **Phone Number**
       - **ComboBox** for selecting the **Department** (e.g., Computer Science, Business, Arts)
       - **Button** to **Register** alumni
       - **Button** to **Display** registered alumni
       - **DataGridView** control to display the list of registered alumni from the selected department
  2. **Implement Functionality Using ADO.NET**:
     + **Register Button**:
       - When the **Register** button is clicked, validate the input fields.
       - If the inputs are valid, insert the entered details into the database using ADO.NET. Handle any database exceptions that may occur. o **Display Button**:
       - When the **Display** button is clicked, retrieve all registered alumni for the selected department from the ComboBox.
       - Display the results in the **DataGridView** control.

**FORM AND CODE:**



**Implementation of Design the Form:**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace AlumniRegistrationSystem

{

public partial class AlumniRegistrationForm : Form

{

// Connection string to the SQL Server (replace with your actual connection string)

private string connectionString = @"Server=your\_server\_name;Database=your\_database\_name;Integrated Security=True;";

public AlumniRegistrationForm()

{

InitializeComponent();

}

private void AlumniRegistrationForm\_Load(object sender, EventArgs e)

{

// Populate ComboBox with departments

cmbDepartment.Items.Add("Computer Science");

cmbDepartment.Items.Add("Business");

cmbDepartment.Items.Add("Arts");

cmbDepartment.SelectedIndex = 0;

}

// Register Button click handler

private void btnRegister\_Click(object sender, EventArgs e)

{

string alumniName = txtAlumniName.Text;

string email = txtEmail.Text;

string phone = txtPhone.Text;

string department = cmbDepartment.SelectedItem.ToString();

// Validate inputs

if (string.IsNullOrEmpty(alumniName) || string.IsNullOrEmpty(email) || string.IsNullOrEmpty(phone))

{

MessageBox.Show("Please fill in all the fields.");

return;

}

// Insert the alumni data into the database

try

{

using (SqlConnection conn = new SqlConnection(connectionString))

{

string query = "INSERT INTO Alumni (AlumniName, Email, Phone, Department) VALUES (@AlumniName, @Email, @Phone, @Department)";

SqlCommand cmd = new SqlCommand(query, conn);

cmd.Parameters.AddWithValue("@AlumniName", alumniName);

cmd.Parameters.AddWithValue("@Email", email);

cmd.Parameters.AddWithValue("@Phone", phone);

cmd.Parameters.AddWithValue("@Department", department);

conn.Open();

cmd.ExecuteNonQuery();

conn.Close();

MessageBox.Show("Alumni registered successfully!");

ClearInputs(); // Clear the input fields after successful registration

}

}

catch (SqlException ex)

{

MessageBox.Show("An error occurred while registering: " + ex.Message);

}

}

// Display Button click handler

private void btnDisplay\_Click(object sender, EventArgs e)

{

string selectedDepartment = cmbDepartment.SelectedItem.ToString();

DisplayAlumniList(selectedDepartment);

}

// Method to display alumni list based on selected department

private void DisplayAlumniList(string department)

{

try

{

using (SqlConnection conn = new SqlConnection(connectionString))

{

string query = "SELECT AlumniName, Email, Phone FROM Alumni WHERE Department = @Department";

SqlDataAdapter dataAdapter = new SqlDataAdapter(query, conn);

dataAdapter.SelectCommand.Parameters.AddWithValue("@Department", department);

DataTable alumniData = new DataTable();

dataAdapter.Fill(alumniData);

dgvAlumni.DataSource = alumniData;

}

}

catch (SqlException ex)

{

MessageBox.Show("An error occurred while retrieving alumni data: " + ex.Message);

}

}

private void ClearInputs()

{

txtAlumniName.Clear();

txtEmail.Clear();

txtPhone.Clear();

cmbDepartment.SelectedIndex = 0;

}

}

}

**Implementation of Database Table:**

CREATE TABLE Alumni (

AlumniId INT IDENTITY(1,1) PRIMARY KEY,

AlumniName NVARCHAR(100),

Email NVARCHAR(100),

Phone NVARCHAR(15),

Department NVARCHAR(50)

);

By:

R.Sujith

# 