

# C# and .NET Frameworks Assignment

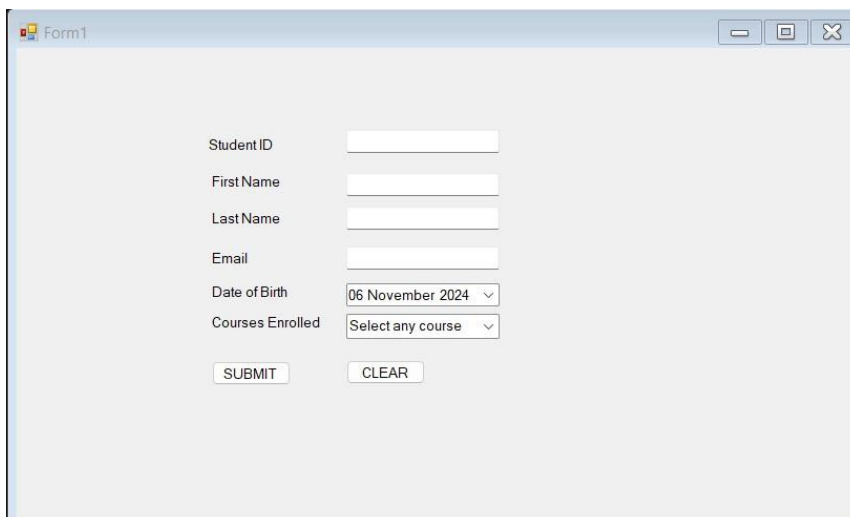
## 2

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** ○ **Last Name**
  - **Date of Birth** ○ **Email** ○ **Phone Number** ○ **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:



The screenshot shows a Windows Forms application window titled "Form1". Inside the window, there is a registration form with the following elements:

- Labels and input fields for: StudentID, First Name, Last Name, Email, Date of Birth, and Courses Enrolled.
- The "Date of Birth" field is a date picker showing "06 November 2024".
- The "Courses Enrolled" field is a dropdown menu showing "Select any course".
- At the bottom of the form, there are two buttons: "SUBMIT" and "CLEAR".

Form1

Student ID

First Name

Last Name

Email

Date of Birth

Courses Enrolled

```

1  using System;
2  using System.Data.SqlClient;
3  using System.Windows.Forms;
4  public partial class StudentForm : Form
5  {
6      public StudentForm()
7      {
8          InitializeComponent();
9      }
10     private void Form1_Load(object sender, EventArgs e)
11     {
12         LoadStudentData();
13     }
14     private void btnSubmit_Click(object sender, EventArgs e)
15     {
16         if (ValidateForm())
17         {
18             SaveStudentData();
19             ClearForm();
20             LoadStudentData();
21         }
22     }
23     private bool ValidateForm()
24     {
25         if (string.IsNullOrEmpty(txtFirstName.Text) || string.IsNullOrEmpty(txtEmail.Text))
26         {
27             MessageBox.Show("First Name and Email are required.");
28             return false;
29         }
30         return true;
31     }
32     private void SaveStudentData()
33     {
34         string connectionString = "your connection string";
35         using (SqlConnection con = new SqlConnection(connectionString))
36         {
37             con.Open();
38             string query = "INSERT INTO Students (FirstName, LastName, DateOfBirth, PhoneNumber, Email, CourseEnrolled) " +
39                             "VALUES (@FirstName, @LastName, @DateOfBirth, @PhoneNumber, @Email, @CourseEnrolled)";
40             using (SqlCommand cmd = new SqlCommand(query, con))
41             {
42                 cmd.Parameters.AddWithValue("@FirstName", txtFirstName.Text);
43                 cmd.Parameters.AddWithValue("@LastName", txtLastName.Text);
44                 cmd.Parameters.AddWithValue("@DateOfBirth", dtpDateOfBirth.Value);
45                 cmd.Parameters.AddWithValue("@PhoneNumber", txtPhoneNumber.Text);
46                 cmd.Parameters.AddWithValue("@Email", txtEmail.Text);
47                 cmd.Parameters.AddWithValue("@CourseEnrolled", cabCourseEnrolled.SelectedItem.ToString());
48                 cmd.ExecuteNonQuery();
49             }
50         }
51     }
52     private void ClearForm()
53     {
54         txtFirstName.Clear();
55         txtLastName.Clear();
56         txtEmail.Clear();
57         txtPhoneNumber.Clear();
58         cabCourseEnrolled.SelectedIndex = -1;
59         dtpDateOfBirth.Value = DateTime.Now;
60     }
61     private void LoadStudentData()
62     {
63         // Your logic to load student data into the DataGridView
64     }
65 }
66
67

```

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:

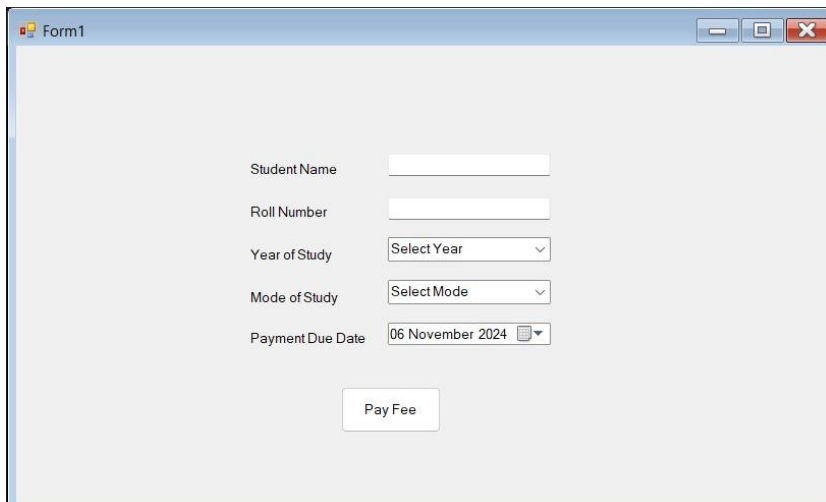
#### 1. Form Design:

- 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:



The screenshot shows a Windows Form titled "Form1" with a light gray background. It contains the following controls:

- Student Name:** A text input field.
- Roll Number:** A text input field.
- Year of Study:** A dropdown menu with "Select Year" and a downward arrow.
- Mode of Study:** A dropdown menu with "Select Mode" and a downward arrow.
- Payment Due Date:** A date picker showing "06 November 2024" with a calendar icon and a downward arrow.
- Pay Fee:** A button located below the input fields.

Form1

Student Name

Roll Number

Year of Study

Mode of Study

Payment Due Date

```

1  using System;
2  using System.Windows.Forms;
3  public partial class FeePaymentForm : Form
4  {
5      const decimal HostelFee = 50000;
6      const decimal DayScholarFee = 30000;
7      const decimal LateFeePerDay = 100;
8      public FeePaymentForm()
9      {
10         InitializeComponent();
11     }
12     private void btnPayFee_Click(object sender, EventArgs e)
13     {
14         if (ValidateForm())
15         {
16             decimal totalFee = CalculateFee();
17             string billNumber = GenerateBillNumber();
18             DisplayBill(billNumber, totalFee);
19         }
20     }
21     private bool ValidateForm()
22     {
23         if (string.IsNullOrEmpty(txtStudentName.Text) || string.IsNullOrEmpty(txtRollNumber.Text))
24         {
25             MessageBox.Show("Student Name and Roll Number are required.");
26             return false;
27         }
28         return true;
29     }
30     private decimal CalculateFee()
31     {
32         decimal baseFee = 0;
33         if (cmbHostelDayScholar.SelectedItem.ToString() == "Hostel Resident")
34         {
35             baseFee = HostelFee;
36         }
37         else
38         {
39             baseFee = DayScholarFee;
40         }
41         DateTime dueDate = dtpDueDate.Value;
42         DateTime paymentDate = DateTime.Today;
43         decimal lateFee = 0;
44         if (paymentDate > dueDate)
45         {
46             int daysLate = (paymentDate - dueDate).Days;
47             lateFee = daysLate * LateFeePerDay;
48         }
49         return baseFee + lateFee;
50     }
51     private string GenerateBillNumber()
52     {
53         return "BILL-" + DateTime.Now.ToString("yyyyMMddHmss") + "-" + txtRollNumber.Text;
54     }
55     private void DisplayBill(string billNumber, decimal totalFee)
56     {
57         lblBill.Text = $"Bill Number: {billNumber}\n" +
58             $"Student Name: {txtStudentName.Text}\n" +
59             $"Roll Number: {txtRollNumber.Text}\n" +
60             $"Year of Study: {cmbYearOfStudy.SelectedItem.ToString()}\n" +
61             $"Hostel/Day Scholar: {cmbHostelDayScholar.SelectedItem.ToString()}\n" +
62             $"Total Fees: {totalFee:C}";
63     }
64 }
65
66

```

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:

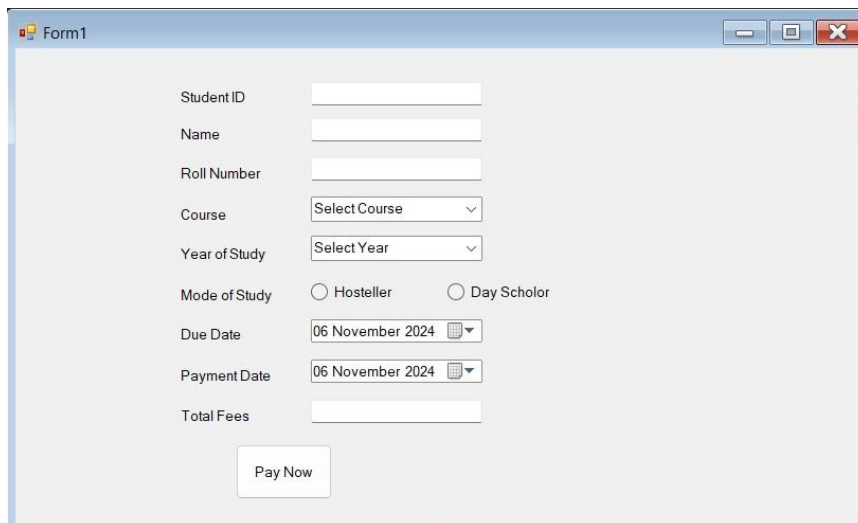
#### Define the Service Requirements:

- Identify the functionality that the web service will provide (e.g., retrieving student information, performing fee calculations, etc.).
- 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:



The screenshot shows a web form titled "Form1" with the following fields and controls:

- Student ID: Text input field
- Name: Text input field
- Roll Number: Text input field
- Course: Dropdown menu with "Select Course" text
- Year of Study: Dropdown menu with "Select Year" text
- Mode of Study: Radio buttons for "Hosteller" and "Day Scholor" (note the typo in the image)
- Due Date: Date picker showing "06 November 2024"
- Payment Date: Date picker showing "06 November 2024"
- Total Fees: Text input field
- Pay Now: Button

Form1

Student ID: 1

Name: John Doe

Roll Number: 1001

Course: AI&DS

Year of Study: IV

Mode of Study: ☐ Hosteller ☒ Day Scholar

Due Date: 12 November 2024

Payment Date: 08 November 2024

Total Fees: 60000

Pay Now

```

1  using System;
2  using System.Collections.Generic;
3  using System.Web.Services;
4  [WebService(Namespace = "http://yurdum.com")]
5  [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
6  public class StudentService : WebService
7  {
8      // Dictionary to store student data
9      private static readonly Dictionary<int, Student> students = new Dictionary<int, Student>
10     {
11         { 1001, new Student { StudentID = 1001, Name = "John Doe", RollNumber = "CS1001", Course = "Computer Science", YearOfStudy = 1 } },
12         { 1002, new Student { StudentID = 1002, Name = "Jane Smith", RollNumber = "EE1002", Course = "Electrical Engineering", YearOfStudy = 3 } }
13     };
14     // Web method to get student details by student ID
15     [WebMethod]
16     public Student GetStudentDetails(int studentID)
17     {
18         if (students.ContainsKey(studentID))
19         {
20             return students[studentID];
21         }
22         else
23         {
24             return null; // Return null if student not found
25         }
26     }
27     // Web method to calculate the total fees (including late fee if applicable)
28     [WebMethod]
29     public decimal CalculateFees(int studentID, bool isHostelResident, DateTime dueDate, DateTime paymentDate)
30     {
31         decimal baseFee = isHostelResident ? 75000 : 30000; // Hostel fee or Day Scholar fee
32
33         decimal lateFee = 0;
34         if (paymentDate > dueDate)
35         {
36             int daysLate = (paymentDate - dueDate).Days;
37             lateFee = daysLate * 100; // 100 per day late fee
38         }
39         return baseFee + lateFee;
40     }
41 }
42 // Student class to represent student data
43 public class Student
44 {
45     public int StudentID { get; set; }
46     public string Name { get; set; }
47     public string RollNumber { get; set; }
48     public string Course { get; set; }
49     public int YearOfStudy { get; set; }
50 }
51

```

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:

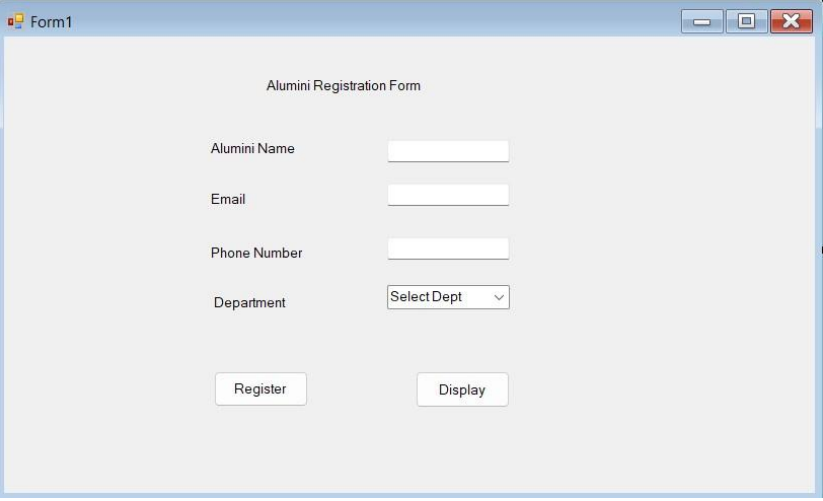
#### 1. Design the Registration Form:

- 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.
- **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:



The screenshot shows a Windows Forms application window titled "Form1". Inside the window, there is a form titled "Alumini Registration Form". The form contains the following controls:

- Labels for "Alumini Name", "Email", "Phone Number", and "Department".
- Three text boxes for entering the Name, Email, and Phone Number.
- A dropdown menu for the Department, currently showing "Select Dept" with a downward arrow.
- Two buttons at the bottom: "Register" and "Display".

Form1

Alumini Registration Form

Alumini Name: Smithi

Email: smithi@gmail.com

Phone Number: 9876543210

Department: Select Dept

Register Display

```

1  using System;
2  using System.Data.SqlClient;
3  using System.Windows.Forms;
4  public partial class Form1 : Form
5  {
6      // Connection string for SQL Server (update it with your actual database details)
7      string connectionString = "Data Source=YourServer;Initial Catalog=YourDatabase;Integrated Security=True;";
8      public Form1()
9      {
10         InitializeComponent();
11     }
12     // Event handler for Register button click
13     private void btnRegister_Click(object sender, EventArgs e)
14     {
15         if (ValidateForm())
16         {
17             try
18             {
19                 using (SqlConnection con = new SqlConnection(connectionString))
20                 {
21                     con.Open();
22                     string query = "INSERT INTO Alumni (AlumniName, Email, PhoneNumber, Department) " +
23                                     "VALUES (@AlumniName, @Email, @PhoneNumber, @Department)";
24
25                     SqlCommand cmd = new SqlCommand(query, con);
26                     cmd.Parameters.AddWithValue("@AlumniName", txtAlumniName.Text);
27                     cmd.Parameters.AddWithValue("@Email", txtEmail.Text);
28                     cmd.Parameters.AddWithValue("@PhoneNumber", txtPhoneNumber.Text);
29                     cmd.Parameters.AddWithValue("@Department", cmbDepartment.SelectedItem.ToString());
30
31                     cmd.ExecuteNonQuery();
32
33                     MessageBox.Show("Alumni registered successfully!");
34                     ClearForm();
35                 }
36             }
37             catch (Exception ex)
38             {
39                 MessageBox.Show("Error: " + ex.Message);
40             }
41         }
42     }
43     // Method to validate form inputs
44     private bool ValidateForm()
45     {
46         if (string.IsNullOrEmpty(txtAlumniName.Text) ||
47             string.IsNullOrEmpty(txtEmail.Text) ||
48             string.IsNullOrEmpty(txtPhoneNumber.Text) ||
49             cmbDepartment.SelectedItem == null)
50         {
51             MessageBox.Show("All fields are required!");
52             return false;
53         }
54         return true;
55     }
56     // Method to clear the form after successful registration
57     private void ClearForm()
58     {
59         txtAlumniName.Clear();
60         txtEmail.Clear();
61         txtPhoneNumber.Clear();
62         cmbDepartment.SelectedIndex = -1; // Reset combo box
63     }
64 }
65

```



**BY:**

JANANI S 73772226118 III – B.TECH AI&DS