

# C# and .NET Frameworks

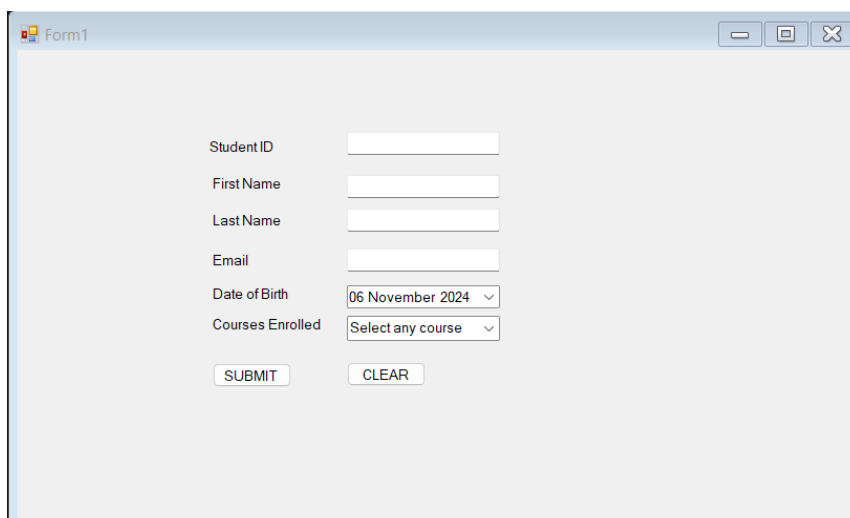
## Assignment 2

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". The form has a light gray background and contains the following controls:

- Student ID: Text input field
- First Name: Text input field
- Last Name: Text input field
- Email: Text input field
- Date of Birth: Date picker control showing "06 November 2024"
- Courses Enrolled: Dropdown menu showing "Select any course"
- SUBMIT: Button
- CLEAR: Button

Form1

Student ID: 73772226157

First Name: Varuna

Last Name: Sree

Email: varu1234@gmail.com

Date of Birth: 27 November 2004

Courses Enrolled: AI&DS

SUBMIT CLEAR

```

1  TextBox txtStudentID, txtFirstName, txtLastName, txtEmail,
    txtPhoneNumber;
2  ComboBox cmbCourseEnrolled;
3  DateTimePicker dtpDateOfBirth;
4  Button btnSubmit, btnClear;
5  DataGridView dgvStudents;
6  private void Form1_Load(object sender, EventArgs e) {
7      LoadStudentData();
8  }
9  private void btnSubmit_Click(object sender, EventArgs e) {
10     if (ValidateForm()) {
11         SaveStudentData();
12         ClearForm();
13         LoadStudentData();
14     }
15 }
16 private bool ValidateForm() {
17     if (string.IsNullOrEmpty(txtFirstName.Text) || string
        .IsNullOrEmpty(txtEmail.Text)) {
18         MessageBox.Show("First Name and Email are required.");
19         return false;
20     }
21     return true;
22 }
23 private void SaveStudentData() {

```

```

24     using (SqlConnection con = new SqlConnection
        ("your_connection_string")) {
25         con.Open();
26         SqlCommand cmd = new SqlCommand("INSERT INTO Students
            (FirstName, LastName, DateOfBirth, Email, PhoneNumber,
            CourseEnrolled) " +
27             "VALUES (@FirstName,
                @LastName, @DateOfBirth, @Email,
                @PhoneNumber, @CourseEnrolled)",
            con);
28         cmd.Parameters.AddWithValue("@FirstName", txtFirstName.Text
            );
29         cmd.Parameters.AddWithValue("@LastName", txtLastName.Text);
30         cmd.Parameters.AddWithValue("@DateOfBirth", dtpDateOfBirth
            .Value);
31         cmd.Parameters.AddWithValue("@Email", txtEmail.Text);
32         cmd.Parameters.AddWithValue("@PhoneNumber", txtPhoneNumber
            .Text);
33         cmd.Parameters.AddWithValue("@CourseEnrolled",
            cmbCourseEnrolled.SelectedItem.ToString());
34         cmd.ExecuteNonQuery();
35     }
36 }
37 private void ClearForm() {
38     txtFirstName.Clear();
39     txtLastName.Clear();

```

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:

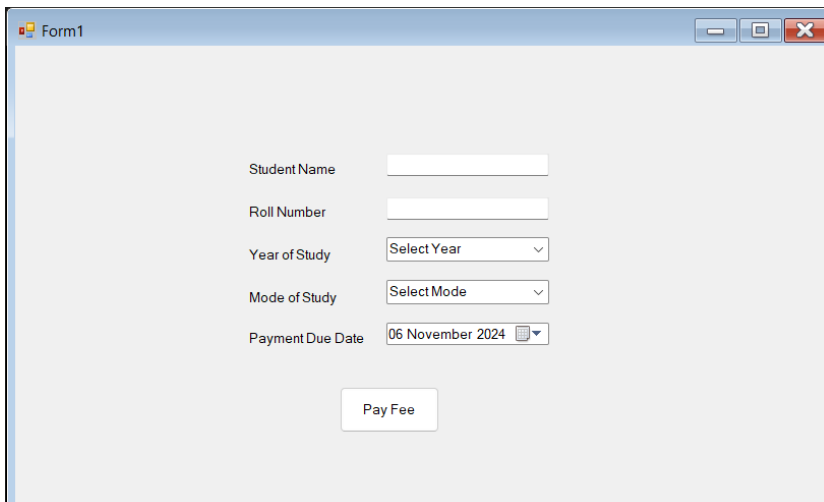
#### 1. Form Design:

- 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 five input fields arranged vertically, each with a label to its left: "Student Name" (text box), "Roll Number" (text box), "Year of Study" (dropdown menu showing "Select Year"), "Mode of Study" (dropdown menu showing "Select Mode"), and "Payment Due Date" (date picker showing "06 November 2024"). Below these fields is a button labeled "Pay Fee".

Form1

Student Name

Roll Number

Year of Study

Mode of Study

Payment Due Date

```

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

```

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:

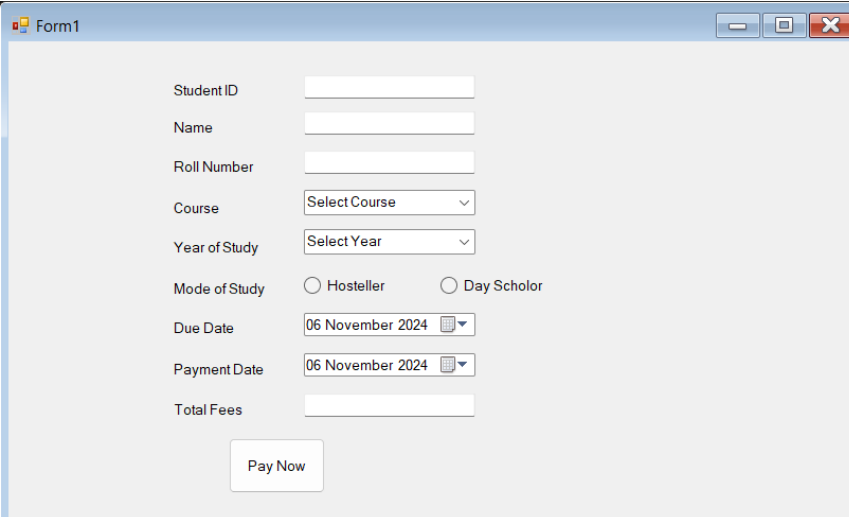
#### Define the Service Requirements:

- 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.Web.Services;
3- [WebService(Namespace = "http://yourdomain.com/")]
4- [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
5- public class StudentService : WebService {
6-     private static readonly Dictionary<int, Student> students = new Dictionary<int, Student> {
7-         { 1, new Student { StudentID = 1, Name = "John Doe", RollNumber = "1001", Course = "Computer Science", YearOfStudy = 2 } },
8-         { 2, new Student { StudentID = 2, Name = "Jane Smith", RollNumber = "1002", Course = "Electrical Engineering", YearOfStudy = 3 } }
9-     };
10-    public Student GetStudentDetails(int studentID) {
11-        if (students.ContainsKey(studentID)) {
12-            return students[studentID];
13-        }
14-        else {
15-            return null;
16-        }
17-    }
18-    public decimal CalculateFees(int studentID, bool isHostelResident, DateTime dueDate, DateTime paymentDate) {
19-        decimal baseFee = isHostelResident ? 50000 : 30000;
20-        decimal lateFee = 0;
21-        if (paymentDate > dueDate) {
22-            int daysLate = (paymentDate - dueDate).Days;
23-            lateFee = daysLate * 100;
24-        }
25-        return baseFee + lateFee;
26-    }
27- }
28- public class Student {
29-     public int StudentID { get; set; }
30-     public string Name { get; set; }
31-     public string RollNumber { get; set; }
32-     public string Course { get; set; }
33-     public int YearOfStudy { get; set; }
34- }

```

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:

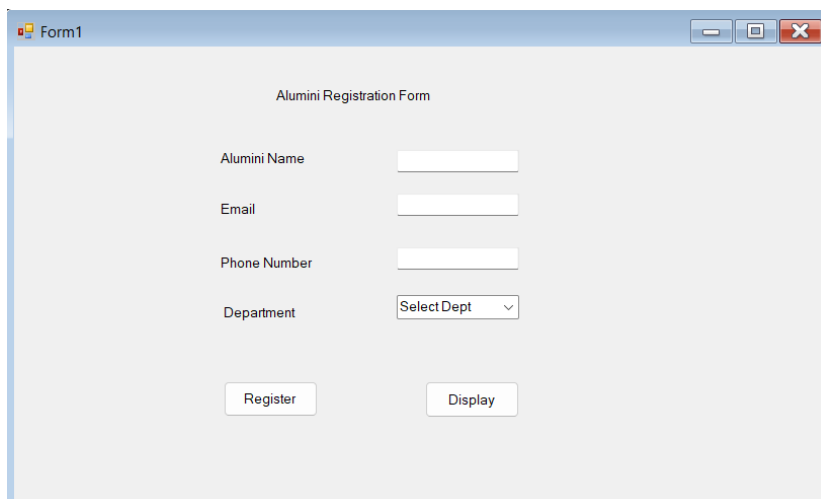
#### 1. Design the Registration Form:

- 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 four input fields: "Alumini Name", "Email", "Phone Number", and "Department". The "Department" field is a ComboBox with a dropdown arrow and the text "Select Dept". Below the input fields, there are two buttons: "Register" and "Display".

Form1

Alumini Registration Form

Alumini Name

Email

Phone Number

Department

```

1- using System;
2- using System.Data.SqlClient;
3- using System.Windows.Forms;
4- public partial class Form1 : Form {
5-     string connectionString = @"Data Source=YourServer;Initial
        Catalog=YourDatabase;Integrated Security=True;";
6-     public Form1() {
7-         InitializeComponent();
8-     }
9-     private void btnRegister_Click(object sender, EventArgs e) {
10-         if (ValidateForm()) {
11-             try {
12-                 using (SqlConnection con = new SqlConnection
                    (connectionString)) {
13-                     con.Open();
14-                     string query = "INSERT INTO Alumni (AlumniName,
                        Email, PhoneNumber, Department) " +
15-                         "VALUES (@AlumniName, @Email,
                            @PhoneNumber, @Department)";
16-
17-                     SqlCommand cmd = new SqlCommand(query, con);
18-                     cmd.Parameters.AddWithValue("@AlumniName",
                        txtAlumniName.Text);
19-                     cmd.Parameters.AddWithValue("@Email", txtEmail
                        .Text);

```

```

20-                     cmd.Parameters.AddWithValue("@PhoneNumber",
                        txtPhoneNumber.Text);
21-                     cmd.Parameters.AddWithValue("@Department",
                        cmbDepartment.SelectedItem.ToString());
22-
23-                     cmd.ExecuteNonQuery();
24-                     MessageBox.Show("Alumni registered
                        successfully!");
25-                     ClearForm();
26-                 }
27-             }
28-             catch (Exception ex) {
29-                 MessageBox.Show("Error: " + ex.Message);
30-             }
31-         }
32-     }
33-     private bool ValidateForm() {
34-         if (string.IsNullOrEmpty(txtAlumniName.Text) || string
            .IsNullOrEmpty(txtEmail.Text) ||
35-             string.IsNullOrEmpty(txtPhoneNumber.Text) ||
            cmbDepartment.SelectedItem == null) {
36-             MessageBox.Show("All fields are required!");
37-             return false;
38-         }
39-         return true;
40-     }

```



```
41 private void ClearForm() {  
42     txtAlumniName.Clear();  
43     txtEmail.Clear();  
44     txtPhoneNumber.Clear();  
45     cmbDepartment.SelectedIndex = -1;  
46 }  
47 }
```

**BY:**

RAGAVAN D

73772226138

III – B.TECH AI&DS