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.

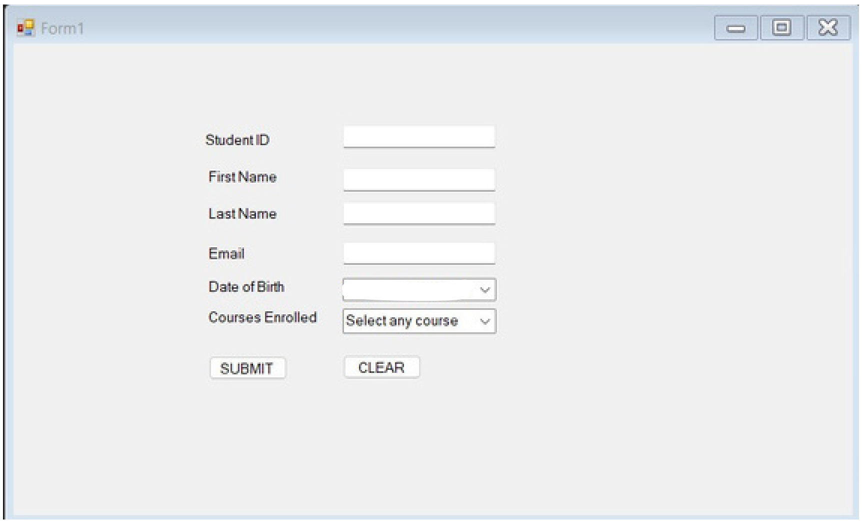
Requirements:

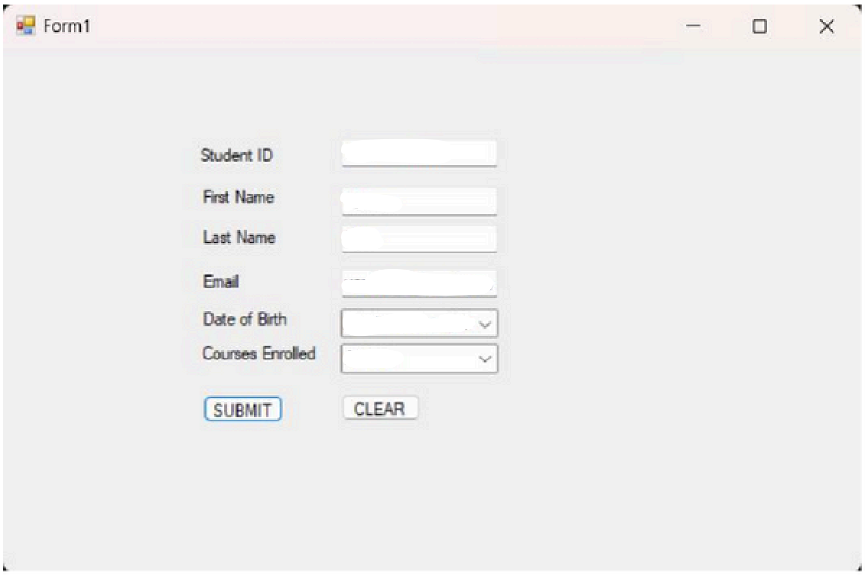
1. The registration form should include the following fields:

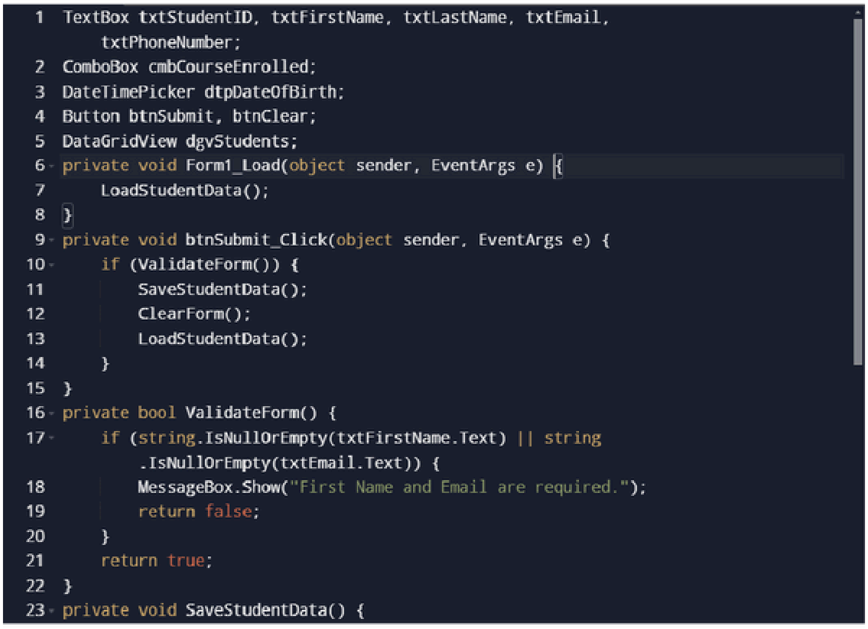
o **Student ID** (Auto-generated or entered manually)
o **First Name**
o **Last Name**
- o **Date of Birth**
- o **Email**
- o **Phone Number**
- o **Course Enrolled**

2. Implement the following features:
- o **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.).
- o **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.
- o **Reset Form**: Provide a "Clear" button to reset all the input fields.
- o **Display Students**: Optionally, include a DataGridView control to display all registered students after submission.

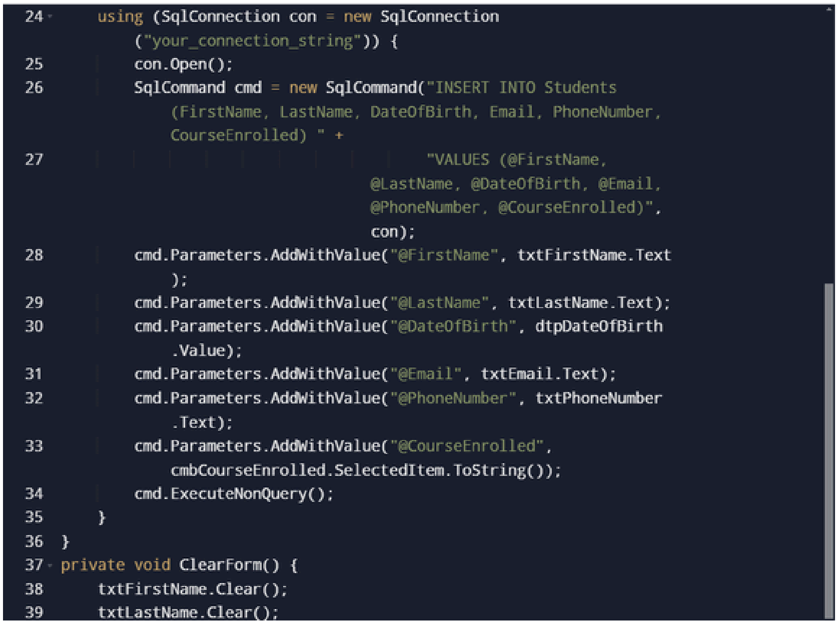
FORM AND CODE:







```
TextBox txtStudentID, txtFirstName, txtLastName, txtEmail, txtPhoneNumber;
ComboBox cmbCourseEnrolled;
DateTimePicker dtpDateOfBirth;
Button btnSubmit, btnClear;
DataGridView dgvStudents;
private void Form1_Load(object sender, EventArgs e) {
    LoadStudentData();
}
private void btnSubmit_Click(object sender, EventArgs e) {
    if (ValidateForm()) {
        SaveStudentData();
        ClearForm();
        LoadStudentData();
    }
}
private bool ValidateForm() {
    if (string.IsNullOrEmpty(txtFirstName.Text) || string.IsNullOrEmpty(txtEmail.Text)) {
        MessageBox.Show("First Name and Email are required.");
        return false;
    }
    return true;
}
private void SaveStudentData() {
```



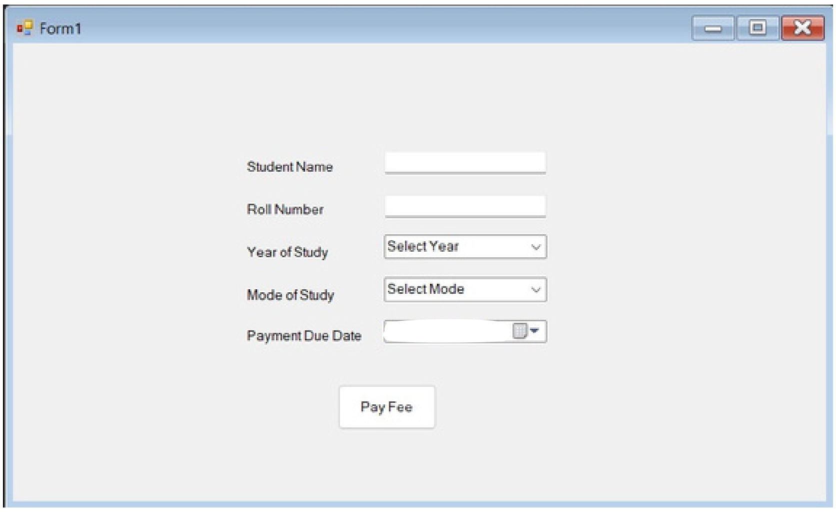
```
    using (SqlConnection con = new SqlConnection("your_connection_string")) {
        con.Open();
        SqlCommand cmd = new SqlCommand("INSERT INTO Students (FirstName, LastName, DateOfBirth, Email, PhoneNumber, CourseEnrolled) " +
                                        "VALUES (@FirstName, @LastName, @DateOfBirth, @Email, @PhoneNumber, @CourseEnrolled)", con);
        cmd.Parameters.AddWithValue("@FirstName", txtFirstName.Text);
        cmd.Parameters.AddWithValue("@LastName", txtLastName.Text);
        cmd.Parameters.AddWithValue("@DateOfBirth", dtpDateOfBirth.Value);
        cmd.Parameters.AddWithValue("@Email", txtEmail.Text);
        cmd.Parameters.AddWithValue("@PhoneNumber", txtPhoneNumber.Text);
        cmd.Parameters.AddWithValue("@CourseEnrolled", cmbCourseEnrolled.SelectedItem.ToString());
        cmd.ExecuteNonQuery();
    }
}
private void ClearForm() {
    txtFirstName.Clear();
    txtLastName.Clear();
```

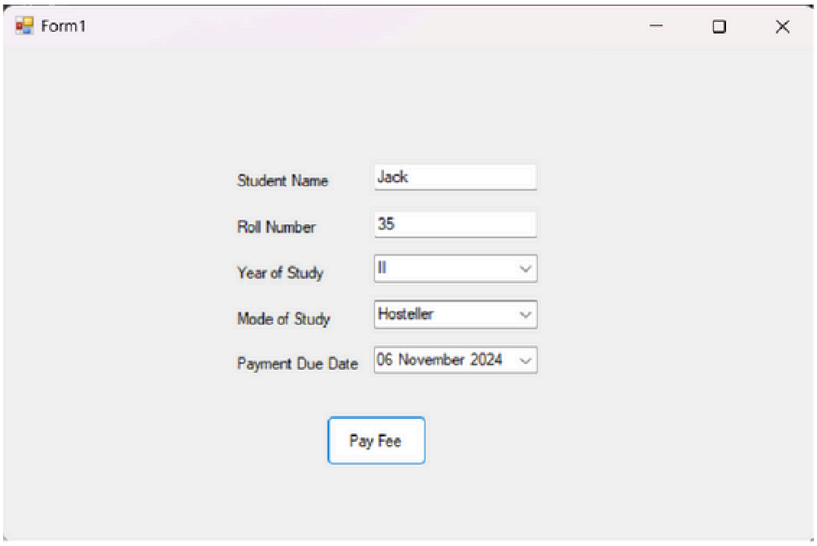
2. Design and implement a **Student Fee Payment System** using C# and Windows Forms. The application should allow students to enter their details, pay their fees, and generate a bill with a unique bill number.

Requirements:

1. **Form Design**:
   - o 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**:
   - o 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:





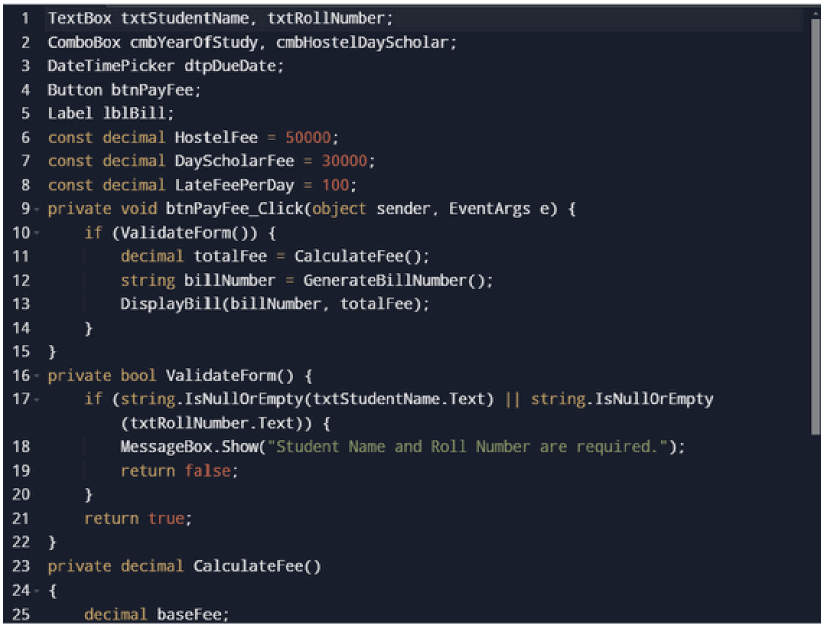
Student Name: Jack

Roll Number: 35

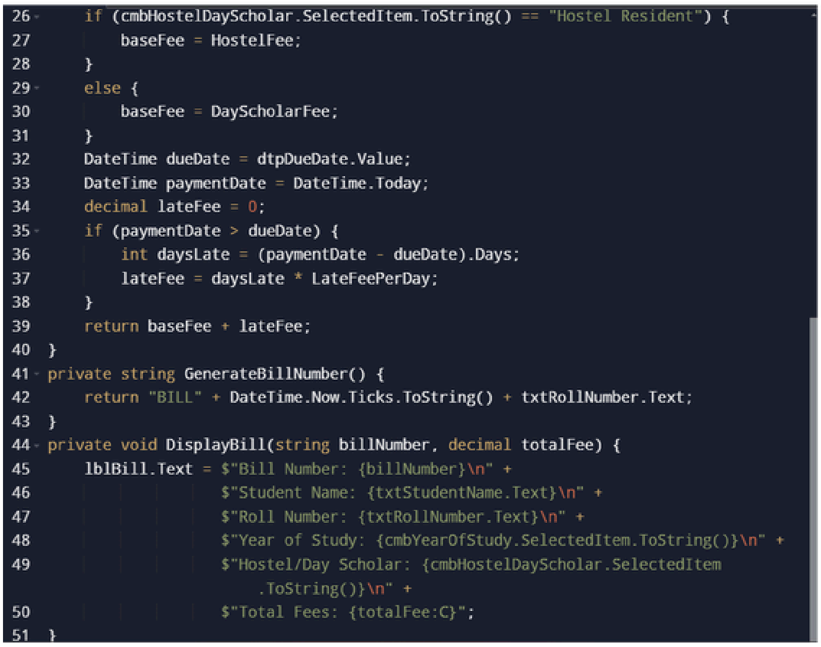
Year of Study: II

Mode of Study: Hosteller

Payment Due Date: 06 November 2024



```
TextBox txtStudentName, txtRollNumber;
ComboBox cmbYearOfStudy, cmbHostelDayScholar;
DateTimePicker dtpDueDate;
Button btnPayFee;
Label lblBill;
const decimal HostelFee = 50000;
const decimal DayScholarFee = 30000;
const decimal LateFeePerDay = 100;
private void btnPayFee_Click(object sender, EventArgs e) {
    if (ValidateForm()) {
        decimal totalFee = CalculateFee();
        string billNumber = GenerateBillNumber();
        DisplayBill(billNumber, totalFee);
    }
}
private bool ValidateForm() {
    if (string.IsNullOrEmpty(txtStudentName.Text) || string.IsNullOrEmpty
        (txtRollNumber.Text)) {
        MessageBox.Show("Student Name and Roll Number are required.");
        return false;
    }
    return true;
}
private decimal CalculateFee()
{
    decimal baseFee;
```



```
    if (cmbHostelDayScholar.SelectedItem.ToString() == "Hostel Resident") {
        baseFee = HostelFee;
    }
    else {
        baseFee = DayScholarFee;
    }
    DateTime dueDate = dtpDueDate.Value;
    DateTime paymentDate = DateTime.Today;
    decimal lateFee = 0;
    if (paymentDate > dueDate) {
        int daysLate = (paymentDate - dueDate).Days;
        lateFee = daysLate * LateFeePerDay;
    }
    return baseFee + lateFee;
}
private string GenerateBillNumber() {
    return "BILL" + DateTime.Now.Ticks.ToString() + txtRollNumber.Text;
}
private void DisplayBill(string billNumber, decimal totalFee) {
    lblBill.Text = $"Bill Number: {billNumber}\n" +
                   $"Student Name: {txtStudentName.Text}\n" +
                   $"Roll Number: {txtRollNumber.Text}\n" +
                   $"Year of Study: {cmbYearOfStudy.SelectedItem.ToString()}\n" +
                   $"Hostel/Day Scholar: {cmbHostelDayScholar.SelectedItem
                       .ToString()}\n" +
                   $"Total Fees: {totalFee:C}";
}
```

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:

o 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:

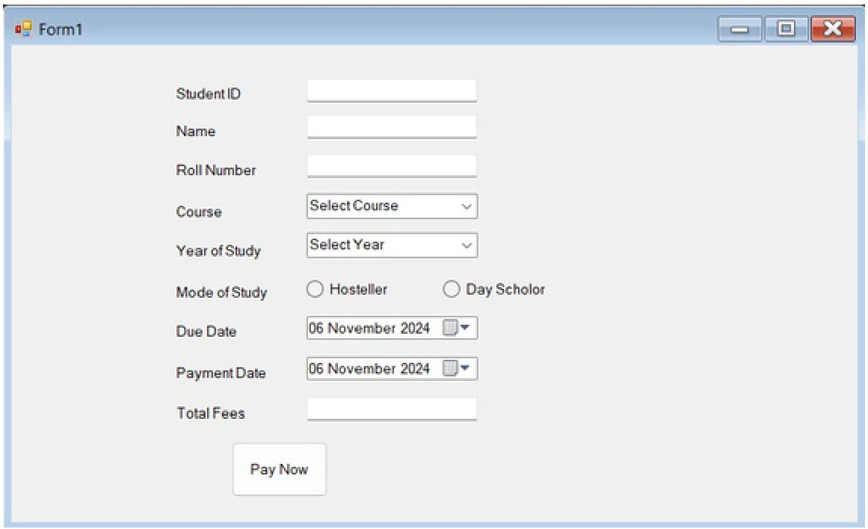
o In Visual Studio, create a new **ASP.NET Web Service** project.

o Define a service class by inheriting from System.Web.Services.WebService.

o Decorate the class with the [WebService] attribute and each method with the [WebMethod] attribute to expose them as web service operations.

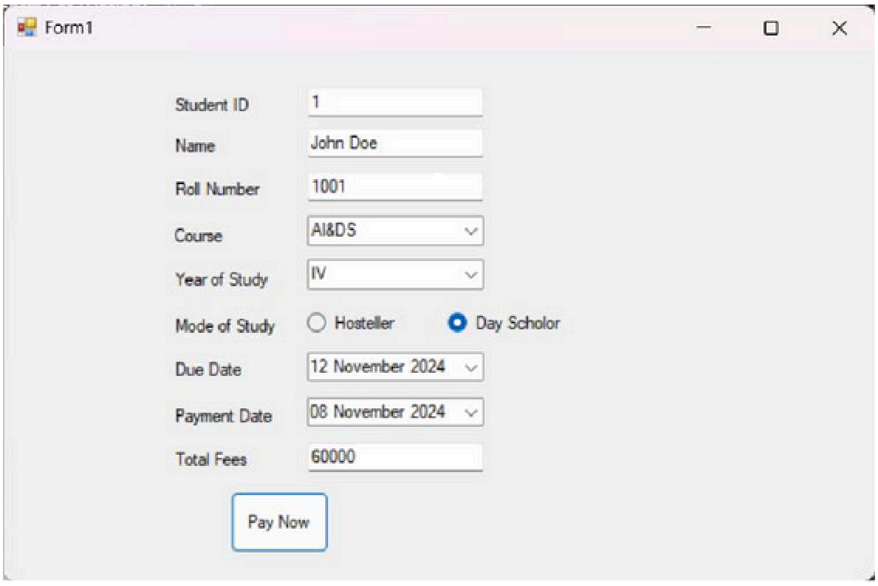
o Implement the required service methods (e.g., retrieving student data or calculating fees).

FORM AND CODE:



Form1

Student ID	
Name	
Roll Number	
Course	Select Course
Year of Study	Select Year
Mode of Study	○ Hosteller ○ Day Scholor
Due Date	06 November 2024
Payment Date	06 November 2024
Total Fees	



Student ID: 1

Name: John Doe

Roll Number: 1001

Course: AI&DS

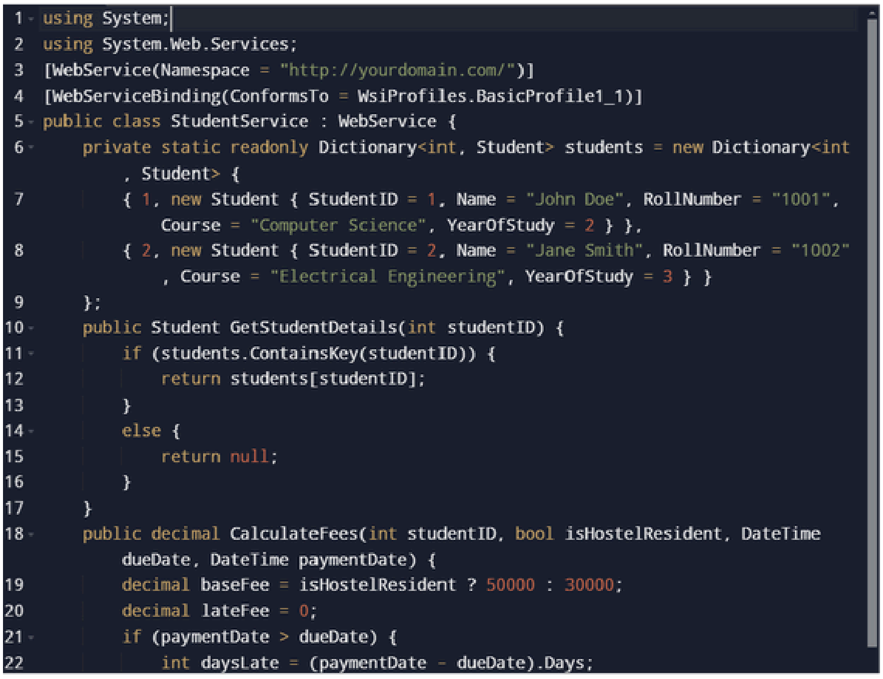
Year of Study: IV

Mode of Study: ○ Hosteller ● Day Scholor

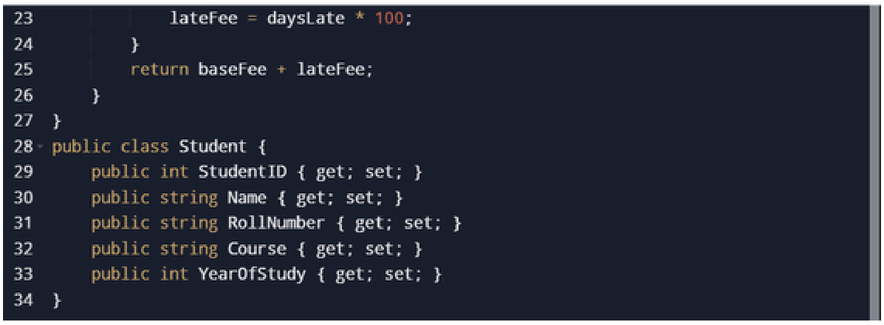
Due Date: 12 November 2024

Payment Date: 08 November 2024

Total Fees: 60000



```
using System;
using System.Web.Services;
[WebService(Namespace = "http://yourdomain.com/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class StudentService : WebService {
    private static readonly Dictionary<int, Student> students = new Dictionary<int, Student> {
        { 1, new Student { StudentID = 1, Name = "John Doe", RollNumber = "1001", Course = "Computer Science", YearOfStudy = 2 } },
        { 2, new Student { StudentID = 2, Name = "Jane Smith", RollNumber = "1002", Course = "Electrical Engineering", YearOfStudy = 3 } }
    };
    public Student GetStudentDetails(int studentID) {
        if (students.ContainsKey(studentID)) {
            return students[studentID];
        }
        else {
            return null;
        }
    }
    public decimal CalculateFees(int studentID, bool isHostelResident, DateTime dueDate, DateTime paymentDate) {
        decimal baseFee = isHostelResident ? 50000 : 30000;
        decimal lateFee = 0;
        if (paymentDate > dueDate) {
            int daysLate = (paymentDate - dueDate).Days;
```



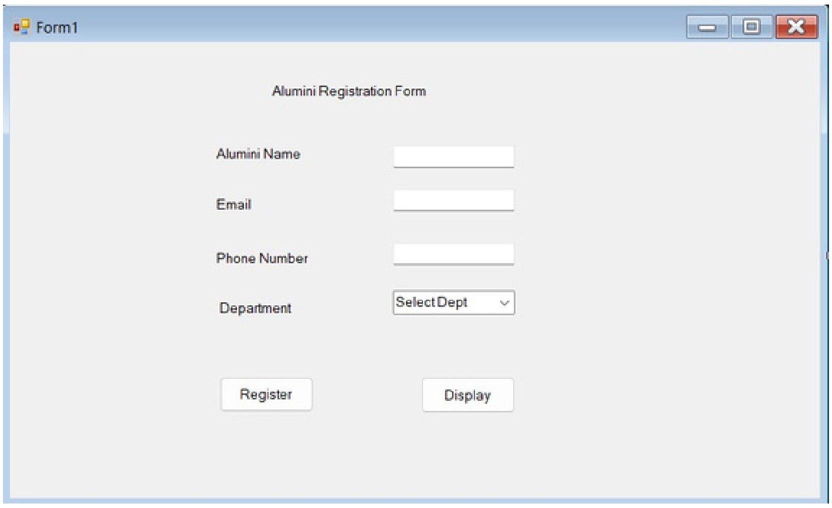
```
            lateFee = daysLate * 100;
        }
        return baseFee + lateFee;
    }
}
public class Student {
    public int StudentID { get; set; }
    public string Name { get; set; }
    public string RollNumber { get; set; }
    public string Course { get; set; }
    public int YearOfStudy { get; set; }
}
```

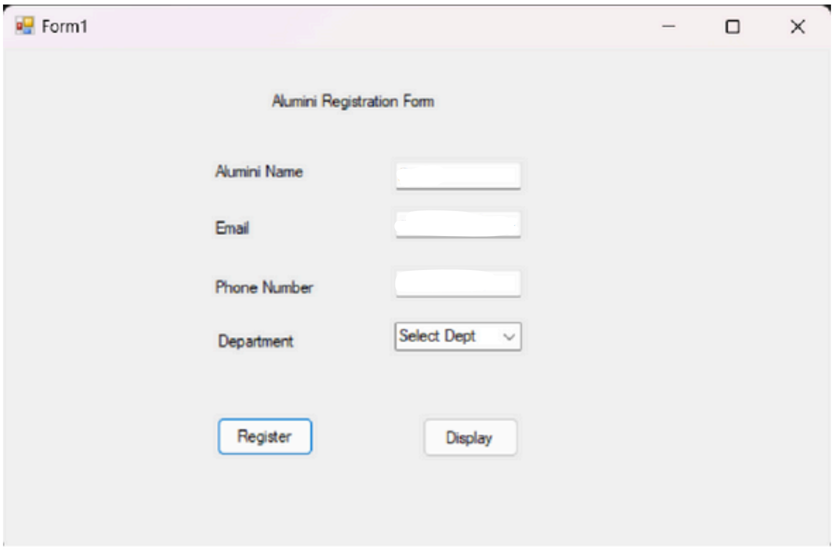
4. Our college is organizing an **Alumni Meet** on **May 5, 2024**. The alumni cell is in the process of creating a database to store a list of registered alumni who will attend the event. You are tasked with designing a registration form and implementing it using ADO.NET.

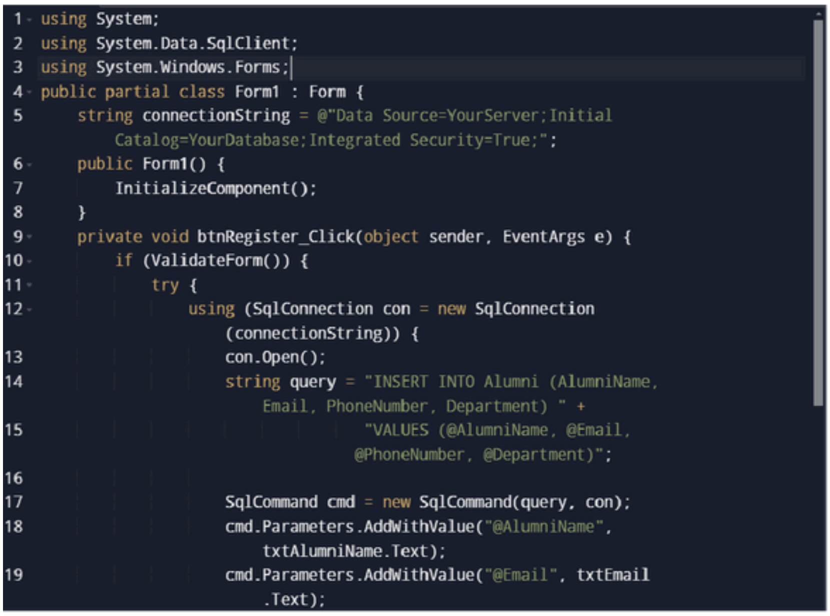
Requirements:

1. **Design the Registration Form**:
   o 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**:
   - o **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:

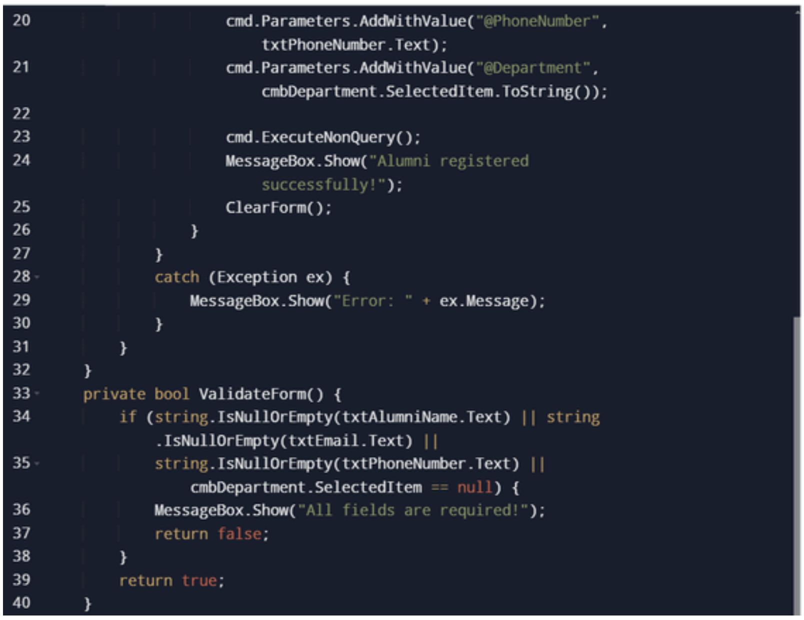






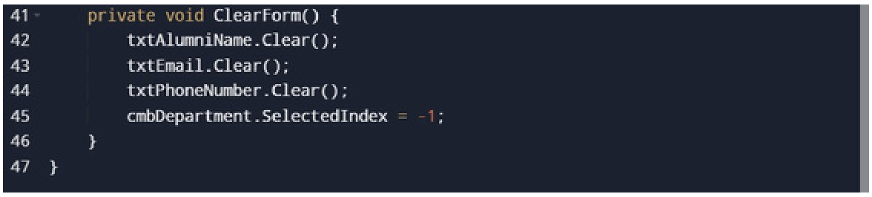
```
using System;
using System.Data.SqlClient;
using System.Windows.Forms;
public partial class Form1 : Form {
    string connectionString = @"Data Source=YourServer;Initial
        Catalog=YourDatabase;Integrated Security=True;";
    public Form1() {
        InitializeComponent();
    }
    private void btnRegister_Click(object sender, EventArgs e) {
        if (ValidateForm()) {
            try {
                using (SqlConnection con = new SqlConnection
                    (connectionString)) {
                    con.Open();
                    string query = "INSERT INTO Alumni (AlumniName,
                        Email, PhoneNumber, Department) " +
                                   "VALUES (@AlumniName, @Email,
                                  @PhoneNumber, @Department)";

                    SqlCommand cmd = new SqlCommand(query, con);
                    cmd.Parameters.AddWithValue("@AlumniName",
                        txtAlumniName.Text);
                    cmd.Parameters.AddWithValue("@Email", txtEmail
                        .Text);
```



```
                    cmd.Parameters.AddWithValue("@PhoneNumber",
                        txtPhoneNumber.Text);
                    cmd.Parameters.AddWithValue("@Department",
                        cmbDepartment.SelectedItem.ToString());

                    cmd.ExecuteNonQuery();
                    MessageBox.Show("Alumni registered
                        successfully!");
                    ClearForm();
                }
            }
            catch (Exception ex) {
                MessageBox.Show("Error: " + ex.Message);
            }
        }
    }
    private bool ValidateForm() {
        if (string.IsNullOrEmpty(txtAlumniName.Text) || string
            .IsNullOrEmpty(txtEmail.Text) ||
            string.IsNullOrEmpty(txtPhoneNumber.Text) ||
                cmbDepartment.SelectedItem == null) {
            MessageBox.Show("All fields are required!");
            return false;
        }
        return true;
    }
```



```
    private void ClearForm() {
        txtAlumniName.Clear();
        txtEmail.Clear();
        txtPhoneNumber.Clear();
        cmbDepartment.SelectedIndex = -1;
    }
}
```