C# and .Net framework

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.

AIM:

To design and implement a Student Registration Form using C# and Windows Forms, allowing users toinput student details and save them to a database.

PROGRAM:

Database Setup:

```
CREATE DATABASE StudentDB;

USE StudentDB;

CREATE TABLE Students (
   StudentID INT PRIMARY KEY
   IDENTITY,Name NVARCHAR(50),
   Age INT,
   Gender
   NVARCHAR(10),
   Email
   NVARCHAR(50),
   Contact NVARCHAR(15)
);
```

Code for Student Registration Form:

```
Using System;
Using System.Data.SqlClient; using System.Windows.Forms;
Namespace StudentRegistrationApp
{
Public partial class StudentRegistrationForm : Form
{
Public StudentRegistrationForm()
{
InitializeComponent();

}
String connectionString = “Data Source=YourServerName;Initial Catalog=StudentDB;Integrated
Security=True”;
```

```
        private void btnSave_Click(object sender, EventArgs e)
        {
            string name = txtName.Text;
            int age = int.Parse(txtAge.Text);
            string gender =
            comboGender.SelectedItem.ToString();string
            email = txtEmail.Text;
            string contact = txtContact.Text;
            using (SqlConnection conn = new SqlConnection(connectionString))
            {
                string query = "INSERT INTO Students (Name, Age, Gender, Email, Contact) VALUES
(@Name,@Age, @Gender, @Email, @Contact)";
                SqlCommand cmd = new SqlCommand(query,
                conn);
                cmd.Parameters.AddWithValue("@Name",
                name); cmd.Parameters.AddWithValue("@Age",
                age);
                cmd.Parameters.AddWithValue("@Gender",
                gender);
                cmd.Parameters.AddWithValue("@Email",
                email);
                cmd.Parameters.AddWithValue("@Contact",
                contact);conn.Open();
                int result = cmd.ExecuteNonQuery();

                if (result > 0)
                {
                    MessageBox.Show("Student registered successfully.");
                }
                else
                {
                    MessageBox.Show("Failed to register student.");
                }
            }
        }
    }
}
```

INPUT:

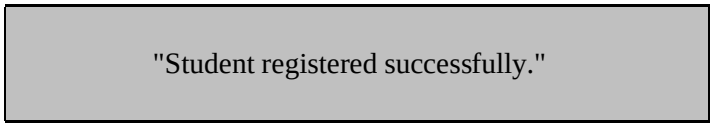
- **Name**: John Doe
- **Age**: 21
- **Gender**: Male
- **Email**: johndoe@example.com
- **Contact**: 1234567890

OUTPUT:

1. **User Input Form**
   The form would look like this in Windows Forms:

2. **Confirmation Message** After clicking the **Save** button, the form displays this message:

Message Box:



Database Output:

Query: SELECT * FROM Students;

SQL Table Output:

Student Id	Name	Age	Gender	Email	Contact
1	John Doe	21	Gender	johndoe@example.com	1234567890

2. DesignandimplementaStudentFeePaymentSystemusingC#andWindowsForms.The application should allow students to enter their details, pay their fees, and generate a bill with a unique bill number.

AIM:

To design and implement a Student Fee Payment System using C# and Windows Forms, allowing students to enter their details, pay their fees, and generate a bill with a unique bill number.

PROGRAM:

```
-- Create the database
CREATE DATABASE StudentFeesDB;

USE StudentFeesDB;

-- Create the Students table
CREATE TABLE Students (
    StudentID INT PRIMARY KEY,
    Name NVARCHAR(50),
    Course NVARCHAR(50)
);

-- Create the Payments table
CREATE TABLE Payments (
    PaymentID INT PRIMARY KEY IDENTITY,
    StudentID INT FOREIGN KEY REFERENCES Students(StudentID),
    Amount DECIMAL(10, 2),
    PaymentDate DATETIME,
    BillNumber UNIQUEIDENTIFIER DEFAULT NEWID()
);
```

Code for Fee Payment System Form:

```
using System;
    using System.Data.SqlClient;
    using System.Windows.Forms;
    namespace StudentFeePaymentApp
    {
       public partial class FeePaymentForm : Form
       {
          public FeePaymentForm(){
             InitializeComponent();
          }
          string connectionString = "Data Source=YourServerName;Initial
    Catalog=StudentFeesDB;Integrated Security=True";
          private void btnPay_Click(object sender, EventArgs e){
             int studentID = int.Parse(txtStudentID.Text);
             string name = txtName.Text;
             string course = txtCourse.Text;
             decimal amount = decimal.Parse(txtAmount.Text);
             DateTime paymentDate = DateTime.Now;
             using (SqlConnection conn = new SqlConnection(connectionString)){
                conn.Open();
                string insertStudent = "IF NOT EXISTS (SELECT * FROM Students WHERE
    StudentID = @StudentID) " +
                                       "INSERT INTO Students (StudentID, Name, Course) VALUES
    (@StudentID, @Name, @Course)";
```

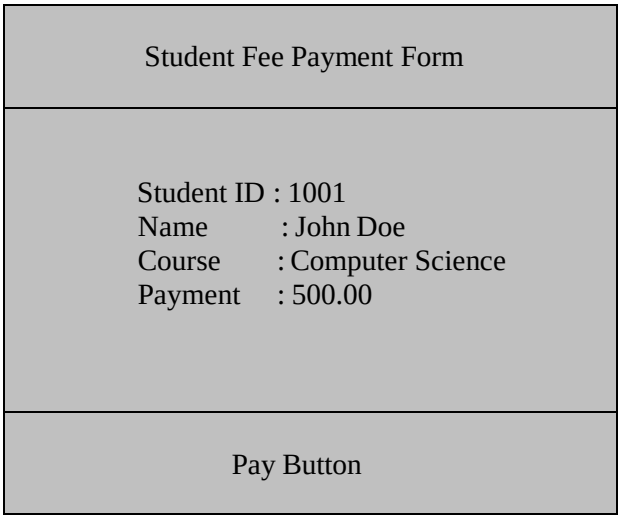
```
            SqlCommand cmdStudent = new SqlCommand(insertStudent, conn);

            cmdStudent.Parameters.AddWithValue("@StudentID", studentID);

            cmdStudent.Parameters.AddWithValue("@Name", name);

            cmdStudent.Parameters.AddWithValue("@Course", course);

            cmdStudent.ExecuteNonQuery();

            string insertPayment = "INSERT INTO Payments (StudentID, Amount,
PaymentDate) VALUES (@StudentID, @Amount, @PaymentDate); " +

                                   "SELECT SCOPE_IDENTITY();";

            SqlCommand cmdPayment = new SqlCommand(insertPayment, conn);

            cmdPayment.Parameters.AddWithValue("@StudentID", studentID);

            cmdPayment.Parameters.AddWithValue("@Amount", amount);

            cmdPayment.Parameters.AddWithValue("@PaymentDate", paymentDate);

            int paymentID = Convert.ToInt32(cmdPayment.ExecuteScalar());

            MessageBox.Show("Payment successful. Bill number will be generated.");

            string getBillNumber = "SELECT BillNumber FROM Payments WHERE
PaymentID = @PaymentID";

            SqlCommand cmdBill = new SqlCommand(getBillNumber, conn);

            cmdBill.Parameters.AddWithValue("@PaymentID", paymentID);

            Guid billNumber = (Guid)cmdBill.ExecuteScalar();

            MessageBox.Show($"Bill Generated Successfully!\n\nBill Number:
{billNumber}\nStudent ID: {studentID}\nName: {name}\nCourse: {course}\nAmount Paid:
{amount:C}\nDate: {paymentDate}");

        }

    }

  }

}
```

INPUT:

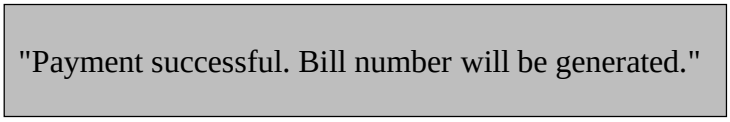
- **Student ID**: 1001
- **Name**: John Doe
- **Course**: Computer Science
- **Fee Amount**: 500.00

OUTPUT:

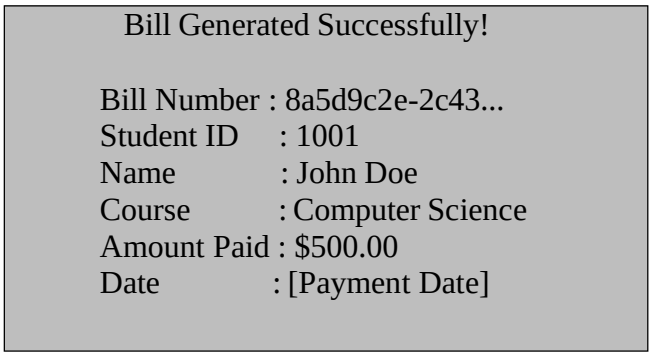
1. Form Layout:



2. Confirmation Message:
Message Box:



3. Bill Display:
Message Box:



3. DesignandimplementaWebServiceusingC#andASP.NETtoexposefunctionalityfor client applicationsto consume. Theweb serviceshould providea specificset ofoperations, such as retrieving data or performing a calculation.

AIM:

To design and implement a Web Service using C# and ASP.NET that exposes specific functionality for clientapplications to consume, such as retrieving data or performing calculations.

PROGRAM:

1. **Project Structure**: Create an ASP.NET Web API project with controllers and models.
2. **Models**: Define `Student` and `Grade` models.

```
namespace StudentWebService.Models
{
   public class Student
   {
      public int StudentID { get; set; }
      public string Name { get; set; }
      public List<int> Grades { get;
      set; }
   }
}
```

3. **Controller**: Create a `StudentController` to define the Web API endpoints.

```
using
Microsoft.AspNetCore.Mvc;
using
StudentWebService.Models;
using
System.Collections.Generic;
using System.Linq;
namespace StudentWebService.Controllers
{
```

```
    [ApiController]
    [Route("api/[controller]
    ")]
    public class StudentController : ControllerBase
    {

        private static readonly List<Student> students = new List<Student>
        {
            new Student { StudentID = 1, Name = "John Doe", Grades = new List<int> { 85, 90, 78
            } }, new Student { StudentID = 2, Name = "Jane Smith", Grades = new List<int> { 92,
            88, 94 } }
        };


        [HttpGet]
        public ActionResult<IEnumerable<Student>> GetStudents()
        {
            return Ok(students);
        }


        [HttpGet("{id}/average")]
        public ActionResult<double> GetAverageGrade(int id)
        {
            var student = students.FirstOrDefault(s => s.StudentID ==
            id);if (student == null)
            {
                return NotFound("Student not found");
            }
            double averageGrade =
            student.Grades.Average();return
            Ok(averageGrade);
        }
    }
}
```

4.Startup Configuration: Configure the API services in `Startup.cs`.

```
using
Microsoft.AspNetCore.Builder;
using
Microsoft.AspNetCore.Hosting;
using
Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;

namespace StudentWebService
{
    public class Startup
    {
        public void ConfigureServices(IServiceCollection services)
        {
```

```
    services.AddControllers();
}
```

```
        public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
        {
            if (env.IsDevelopment())
            {
                app.UseDeveloperExceptionPage();
            }

            app.UseRouting();

            app.UseEndpoints(endpoints =>
            {
                endpoints.MapControllers();
            });
        }
    }
}
```

INPUT:

1. **Get All Students**:
    - URL: `http://localhost:5000/api/student`
    - Method: GET

2. **Get Average Grade for a Student**:
    - URL: `http://localhost:5000/api/student/1/average`
    - Method: GET

OUTPUT:

Get All Students:

- **Request**: `GET http://localhost:5000/api/student`
- **Response** (JSON):


```
[
  {
    "StudentID": 1,
    "Name": "John Doe",
    "Grades": [85, 90, 78]
  },
  {
    "StudentID": 2,
    "Name": "Jane
    Smith",
    "Grades": [92, 88, 94]
  }
```


Get Average Grade for a Student:

- **Request**: `GET http://localhost:5000/api/student/1/average`
- **Response** (JSON):

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.

AIM:

To design a Windows Forms application for alumni registration for the Alumni Meet and implement functionality to store and display registered alumni details using ADO.NET.

PROGRAM:

Database Setup:

Create the AlumniDB database CREATE DATABASE AlumniDB;
USE AlumniDB;
Create the Alumni table CREATE TABLE Alumni (
AlumniID INT PRIMARY KEY IDENTITY, Name NVARCHAR(50),
Email NVARCHAR(50),
PhoneNumber NVARCHAR(15), Department NVARCHAR(50)
);

C# Code for Alumni Registration Form:

```
Using System; using System.Data;
Using System.Data.SqlClient; using System.Windows.Forms;
Namespace AlumniRegistrationApp
{
Public partial class AlumniForm : Form
{
Private string connectionString = “Data Source=YourServerName;Initial
Catalog=AlumniDB;Integrated Security=True”;
Public AlumniForm()
{

InitializeComponent();
 LoadDepartments();

}
Private void LoadDepartments()
{
comboBoxDepartment.Items.AddRange(new string[] { “Computer Science”, “Business”, “Arts” });
}
```

```
private void btnRegister_Click(object sender, EventArgs e)
        {
            string name = txtName.Text;
            string email = txtEmail.Text;
            string phoneNumber = txtPhoneNumber.Text;
            string department = comboBoxDepartment.SelectedItem?.ToString();

            if (string.IsNullOrWhiteSpace(name) || string.IsNullOrWhiteSpace(email) ||
                string.IsNullOrWhiteSpace(phoneNumber) || string.IsNullOrWhiteSpace(department))
            {
                MessageBox.Show("All fields are required.");
                return;
            }

            using (SqlConnection conn = new SqlConnection(connectionString))
            {
                try
                {
                    conn.Open();
                    string query = "INSERT INTO Alumni (Name, Email, PhoneNumber, Department)
    VALUES (@Name, @Email, @PhoneNumber, @Department)";
                    using (SqlCommand cmd = new SqlCommand(query, conn))
                    {
                        cmd.Parameters.AddWithValue("@Name", name);
                        cmd.Parameters.AddWithValue("@Email", email);
                        cmd.Parameters.AddWithValue("@PhoneNumber", phoneNumber);
                        cmd.Parameters.AddWithValue("@Department", department);
                        cmd.ExecuteNonQuery();
                    }
                    MessageBox.Show("Alumni registered successfully.");
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Error: {ex.Message}");
                }
            }
        }

        private void btnDisplay_Click(object sender, EventArgs e)
        {
            string department = comboBoxDepartment.SelectedItem?.ToString();
            if (string.IsNullOrWhiteSpace(department))
            {
                MessageBox.Show("Please select a department.");
                return;
            }
            using (SqlConnection conn = new SqlConnection(connectionString))
            {
                try
                {
                    conn.Open();
```

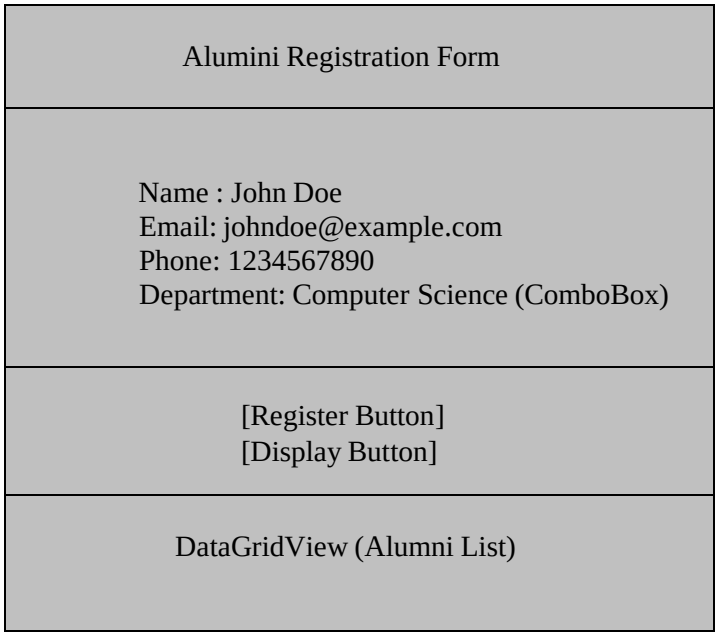
```
                string query = "SELECT AlumniID, Name, Email, PhoneNumber, Department FROM
Alumni WHERE Department = @Department";
                using (SqlCommand cmd = new SqlCommand(query, conn))
                {
                    cmd.Parameters.AddWithValue("@Department", department);
                    using (SqlDataAdapter adapter = new SqlDataAdapter(cmd))
                    {
                        DataTable dt = new DataTable();
                        adapter.Fill(dt);
                        dataGridViewAlumni.DataSource = dt;
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Error: {ex.Message}");
            }
        }
    }
}
```

INPUT:

- **Alumni Name**: John Doe
- **Email**: johndoe@example.com
- **Phone Number**: 1234567890
- **Department**: Computer Science

DISPLAY:

Form Layout:



OUTPUT:

After Registering Alumni:

- Message Box:

"Alumni registered successfully."

Displaying Registered Alumni for Selected Department:

- On clicking **Display** with "Computer Science" selected, the `DataGridView` displays all registered alumni in the Computer Science department:

DataGridView (Alumni List):

Alumini Id	Name	Email	Phone	Dept
1	John Doe	johndoe@example.com	1234567890	CS