Assignment -II

1. Develop the windows form to get name, address and phone number from the user and display it on another form in label control during the form load event.

Form1

```
using System;
using System.Windows.Forms;

namespace UserInfoApp
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }

      private void buttonSubmit_Click(object sender, EventArgs e)
      {
         string name = textBoxName.Text;
         string address = textBoxAddress.Text;
         string phoneNumber = textBoxPhoneNumber.Text;

         Form2 form2 = new Form2(name, address, phoneNumber);
         form2.Show();
      }
   }
}
```

Form2

```
using System;
using System.Windows.Forms;

namespace UserInfoApp
{
   public partial class Form2 : Form
   {
      private string _name;
      private string _address;
      private string _phoneNumber;

      public Form2(string name, string address, string phoneNumber)
      {
         InitializeComponent();
         _name = name;
         _address = address;
         _phoneNumber = phoneNumber;
      }

      private void Form2_Load(object sender, EventArgs e)
      {
         labelDisplayName.Text = "Name: " + _name;
         labelDisplayAddress.Text = "Address: " + _address;
```

```
                labelDisplayPhoneNumber.Text = "Phone Number: " + _phoneNumber;
            }
        }
    }
```

2. Design a web form with a label and four checkboxes. The checkboxes should display the options to make the user select the skill sets he/she possess.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Skill Set Selection</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      margin: 0;
    }

    .form-container {
      background-color: white;
      padding: 20px;
      border-radius: 5px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      width: 300px;
    }

    h2 {
      text-align: center;
      margin-bottom: 20px;
    }

    .checkbox-container {
      margin-bottom: 15px;
    }

    .checkbox-container label {
      margin-left: 10px;
    }

    .checkbox-container input[type="checkbox"] {
      transform: scale(1.2);
      margin-right: 10px;
    }

    button {
      width: 100%;
      padding: 10px;
      background-color: #007BFF;
```

```
            color: white;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }

        button:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>

    <div class="form-container">
        <h2>Select Your Skill Set</h2>
        <form>
            <div class="checkbox-container">
                <input type="checkbox" id="skill1" name="skill" value="HTML">
                <label for="skill1">HTML</label>
            </div>
            <div class="checkbox-container">
                <input type="checkbox" id="skill2" name="skill" value="CSS">
                <label for="skill2">CSS</label>
            </div>
            <div class="checkbox-container">
                <input type="checkbox" id="skill3" name="skill" value="JavaScript">
                <label for="skill3">JavaScript</label>
            </div>
            <div class="checkbox-container">
                <input type="checkbox" id="skill4" name="skill" value="Python">
                <label for="skill4">Python</label>
            </div>
            <button type="submit">Submit</button>
        </form>
    </div>

</body>
</html>
```

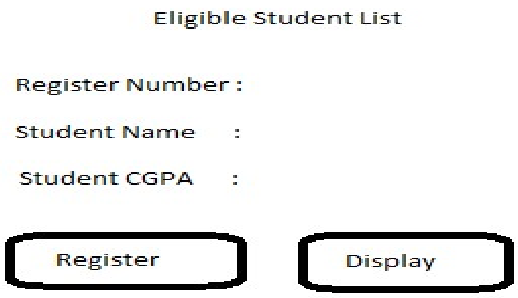
3. Develop the windows application to enable the students to pay their fee and generate the bill with bill number. Make use of windows form controls to design the form to get the information about the student such as name, roll number, year of study, hosteller/day scholar. Once the students enters these details and clicks on “Pay Fee” button, calculate the total fee and display and display bill along with the student details. If the student pay the fees after due date, add the late fee of Rs. 100 per day.

```
using System;
using System.Windows.Forms;

namespace StudentFeePaymentSystem
{
    public partial class Form1 : Form
    {
        private static int billCounter = 1000; // Starting bill number

        public Form1()
        {
            InitializeComponent();
        }

        private void buttonPayFee_Click(object sender, EventArgs e)
        {
            // Get student details
            string name = textBoxName.Text;
            string rollNumber = textBoxRollNumber.Text;
            int yearOfStudy = int.Parse(comboBoxYear.SelectedItem.ToString());
            bool isHosteller = radioButtonHosteller.Checked;
            DateTime dueDate = dateTimePickerDueDate.Value;
            DateTime paymentDate = DateTime.Now;

            // Base fees
            int baseFee = isHosteller ? 50000 : 30000;

            // Late fee calculation
            int lateFee = 0;
            if (paymentDate > dueDate)
            {
                lateFee = (paymentDate - dueDate).Days * 100;
            }

            // Total fee
            int totalFee = baseFee + lateFee;

            // Generate bill number
            int billNumber = GenerateBillNumber();

            // Display the bill
            textBoxBill.Text = GenerateBill(name, rollNumber, yearOfStudy, isHosteller, paymentDat
totalFee, billNumber);
        }

        private int GenerateBillNumber()
        {
            return billCounter++;
        }

        private string GenerateBill(string name, string rollNumber, int yearOfStudy, bool
isHosteller, DateTime paymentDate, int totalFee, int billNumber)
        {
            string studentType = isHosteller ? "Hosteller" : "Day Scholar";
            return $"Bill Number: {billNumber}\n" +
                   $"Name: {name}\n" +
                   $"Roll Number: {rollNumber}\n" +
                   $"Year of Study: {yearOfStudy}\n" +
                   $"Student Type: {studentType}\n" +
                   $"Payment Date: {paymentDate.ToShortDateString()}\n" +
                   $"Total Fee: Rs. {totalFee}";
        }
    }
}
```

4. Assume that your college has planned to conduct placement drive on 29-08-2024. As part of it, the placement cell is in process of creating a database that contains the list of registered eligible students who is going to attend the drive. For that, they have designed the registration form shown in the figure. Design this form using web controls and implement the fallowing task

using ADO.NET and relevant event handling.

- When the Register button is clicked, the entered details should be inserted into the database.
- When display button is clicked, all the registered eligible students should be displayed on a GridView control.



ASP Design

```
<asp:Label ID="lblName" runat="server" Text="Name:"></asp:Label>
<asp:TextBox ID="txtName" runat="server"></asp:TextBox><br />

<asp:Label ID="lblRollNumber" runat="server" Text="Roll Number:"></asp:Label>
<asp:TextBox ID="txtRollNumber" runat="server"></asp:TextBox><br />

<asp:Label ID="lblYear" runat="server" Text="Year of Study:"></asp:Label>
<asp:DropDownList ID="ddlYear" runat="server">
   <asp:ListItem Text="1" Value="1"></asp:ListItem>
   <asp:ListItem Text="2" Value="2"></asp:ListItem>
   <asp:ListItem Text="3" Value="3"></asp:ListItem>
   <asp:ListItem Text="4" Value="4"></asp:ListItem>
</asp:DropDownList><br />

<asp:Label ID="lblBranch" runat="server" Text="Branch:"></asp:Label>
<asp:TextBox ID="txtBranch" runat="server"></asp:TextBox><br />

<asp:Label ID="lblCGPA" runat="server" Text="CGPA:"></asp:Label>
<asp:TextBox ID="txtCGPA" runat="server"></asp:TextBox><br />

<asp:Button ID="btnRegister" runat="server" Text="Register"
OnClick="btnRegister_Click" />
<asp:Button ID="btnDisplay" runat="server" Text="Display"
OnClick="btnDisplay_Click" /><br />

<asp:GridView ID="gvStudents" runat="server"></asp:GridView>
```

Register Click

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
```

```
    protected void btnRegister_Click(object sender, EventArgs e)
    {
        string connectionString =
    ConfigurationManager.ConnectionStrings["PlacementDBConnectionString"].Connectio
    nString;

        using (SqlConnection con = new SqlConnection(connectionString))
        {
            string query = "INSERT INTO Students (Name, RollNumber, Year, Branch,
    CGPA) VALUES (@Name, @RollNumber, @Year, @Branch, @CGPA)";
            using (SqlCommand cmd = new SqlCommand(query, con))
            {
                cmd.Parameters.AddWithValue("@Name", txtName.Text);
                cmd.Parameters.AddWithValue("@RollNumber", txtRollNumber.Text);
                cmd.Parameters.AddWithValue("@Year", ddlYear.SelectedValue);
                cmd.Parameters.AddWithValue("@Branch", txtBranch.Text);
                cmd.Parameters.AddWithValue("@CGPA",
    Convert.ToDouble(txtCGPA.Text));

                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
            }
        }
    }
```

Display

```
    protected void btnDisplay_Click(object sender, EventArgs e)
    {
        string connectionString =
    ConfigurationManager.ConnectionStrings["PlacementDBConnectionString"].Connectio
    nString;

        using (SqlConnection con = new SqlConnection(connectionString))
        {
            string query = "SELECT * FROM Students";
            using (SqlDataAdapter sda = new SqlDataAdapter(query, con))
            {
                DataTable dt = new DataTable();
                sda.Fill(dt);
                gvStudents.DataSource = dt;
                gvStudents.DataBind();
            }
        }
}
```

5. Demonstrate the creation and invocation of web services using C# and ASP.NET with a sample code snippet.

Code for `CalculatorService.asmx.cs`

```
using System;
using System.Web.Services;

namespace SimpleCalculatorWebService
```

```
{
    [WebService(Namespace = "http://yourdomain.com/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    public class CalculatorService : WebService
    {
        [WebMethod]
        public int Add(int a, int b)
        {
            return a + b;
        }

        [WebMethod]
        public int Subtract(int a, int b)
        {
            return a - b;
        }

        [WebMethod]
        public int Multiply(int a, int b)
        {
            return a * b;
        }

        [WebMethod]
        public double Divide(int a, int b)
        {
            if (b == 0)
                throw new ArgumentException("Division by zero is not allowed.");
            return (double)a / b;
        }
    }
}
```

Invoke the Web Service in your client application.

```
using System;

namespace WebServiceClient
{
    class Program
    {
        static void Main(string[] args)
        {
            // Replace "ServiceReference1" with the namespace you used while adding the service
reference.
            var client = new ServiceReference1.CalculatorServiceSoapClient();

            int a = 10, b = 5;

            Console.WriteLine("Addition: " + client.Add(a, b));
            Console.WriteLine("Subtraction: " + client.Subtract(a, b));
            Console.WriteLine("Multiplication: " + client.Multiply(a, b));
            Console.WriteLine("Division: " + client.Divide(a, b));

            client.Close();
        }
    }
}
```

6. Illustrate the process of creating and calling a remote object with suitable sample code snippet.

1. **Service Contract**: This interface defines the methods that the remote object will expose.

```
using System.ServiceModel;

[ServiceContract]
public interface ICalculator
{
    [OperationContract]
    int Add(int a, int b);

    [OperationContract]
    int Subtract(int a, int b);

    [OperationContract]
    int Multiply(int a, int b);

    [OperationContract]
    double Divide(int a, int b);
}
```

2 **Service Implementation**: This class implements the service contract.

```
public class Calculator : ICalculator
{
    public int Add(int a, int b)
    {
        return a + b;
    }

    public int Subtract(int a, int b)
    {
        return a - b;
    }

    public int Multiply(int a, int b)
    {
        return a * b;
    }

    public double Divide(int a, int b)
    {
        if (b == 0)
            throw new ArgumentException("Division by zero is not allowed.");
        return (double)a / b;
    }
}
```

3 Host the WCF Service

```
using System;
using System.ServiceModel;

namespace RemoteCalculatorHost
{
  class Program
  {
    static void Main(string[] args)
    {
      using (ServiceHost host = new
ServiceHost(typeof(RemoteCalculatorService.Calculator)))
      {
        host.Open();
        Console.WriteLine("Service is running... Press Enter to stop.");
        Console.ReadLine();
        host.Close();
      }
    }
  }
}
```

4. **Configure the Service**: In `App.config`, define the service endpoint, binding, and address.

```
<configuration>
 <system.serviceModel>
  <services>
   <service name="RemoteCalculatorService.Calculator">
    <endpoint address="http://localhost:8000/CalculatorService"
         binding="basicHttpBinding"
         contract="RemoteCalculatorService.ICalculator" />
    <host>
     <baseAddresses>
      <add baseAddress="http://localhost:8000/CalculatorService"/>
     </baseAddresses>
    </host>
   </service>
  </services>
  <bindings>
   <basicHttpBinding>
    <binding name="BasicHttpBinding_ICalculator" />
   </basicHttpBinding>
  </bindings>
  <serviceHostingEnvironment multipleSiteBindingsEnabled="true" />
 </system.serviceModel>
</configuration>
```

4 Create the Client Application to Call the Remote Object

```
using System;

namespace RemoteCalculatorClient
{
  class Program
  {
        static void Main(string[] args)
        {
            // Replace "ServiceReference1" with the actual namespace of your service reference.
            var client = new ServiceReference1.CalculatorClient();

            int a = 10, b = 5;

            Console.WriteLine("Addition: " + client.Add(a, b));
            Console.WriteLine("Subtraction: " + client.Subtract(a, b));
            Console.WriteLine("Multiplication: " + client.Multiply(a, b));
            Console.WriteLine("Division: " + client.Divide(a, b));

            client.Close();
        }
    }
}
```