

**Advanced Web Technologies(AWT) Lab (MCAL25)****INDEX****Name of the faculty: Ganesh Bhagwat**

| <b>Experiment Number</b> | <b>Name of the experiment</b>  | <b>Date</b> | <b>CO</b> | <b>Sign</b> |
|--------------------------|--|-------------|-----------|-------------|
| 1                        | <b>Design a Web Application for an Organization with Registration forms and advanced controls.</b>         |             | CO<br>1   |             |
| 2                        | <b>Create a website using the master page concept.</b>   |             | CO<br>1   |             |
| 3                        | <b>Design a Web Application using advanced controls.</b>   |             | CO<br>1   |             |
| 4                        | <b>Webpage Demonstrating Connection-Oriented Architecture (ASP.NET Web Forms with SQL Server Database)</b> |             | CO<br>2   |             |
| 5                        | <b>Webpage Demonstrating Disconnected Architecture (ASP.NET Web Forms with SQL Server Database)</b>        |             | CO<br>2   |             |
| 6                        | <b>Create a webpage that demonstrates the use of data bound controls of ASP.NET.</b>                       |             | CO<br>2   |             |

|    |   |  |             |  |
|----|---|--|-------------|--|
| 7  | <b>Design a webpage to demonstrate the working of a simple stored procedure.</b>      |  | <b>CO 2</b> |  |
| 8  | <b>Design a webpage to demonstrate the working of parameterized stored procedure.</b> |  | <b>CO 2</b> |  |
| 9  | <b>Design a webpage to display the use of LINQ.</b>                                   |  | <b>CO 2</b> |  |
| 10 | <b>Build websites to demonstrate the working of entity frameworks in dot net.</b>     |  | <b>CO 3</b> |  |
| 11 | <b>Design Web Applications using Client Side Session Management</b>                   |  | <b>C03</b>  |  |
| 12 | <b>Design Web Applications using Server Side Session Management Techniques</b>        |  | <b>CO 3</b> |  |
| 13 | <b>Build a web page using AJAX Controls.</b>  |  | <b>CO 3</b> |  |
| 14 | <b>Build a web application to create and use web service in ASP.net</b>               |  | <b>CO 3</b> |  |
| 15 | <b>Build a web application to create and WCF service in ASP.net</b>                   |  | <b>CO 3</b> |  |

|         |          |                    |
|---------|----------|--------------------|
| C-24004 | AWT(LAB) | MCA SEM II 2024-25 |
|---------|----------|--------------------|

|    |   |  |                 |  |  |
|----|---|--|-----------------|--|--|
|    |   |  |                 |  |  |
| 16 | <b>Design web application using MVC framework</b> |  | <b>CO<br/>4</b> |  |  |

## PRACTICAL-1

**Design a Web Application for an Organization with Registration forms and advanced controls.**

### WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="PR_1.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <strong>REGISTRATION FORM</strong><br />
            <br />
        </div>
        <p>
            FIRST NAME<asp:TextBox ID="TextBox1" runat="server"/>
        </p>
        <p>
            LAST NAME<asp:TextBox ID="TextBox2" runat="server"/>
        </p>
        <p>
            EMAIL<asp:TextBox ID="TextBox3" runat="server"/>
        </p>
        <p>
            DATE OF BIRTH<asp:TextBox ID="DOB" runat="server"/>
        </p>
        <p>
            GENDER<asp:DropDownList ID="DropDownList1" runat="server">
                <asp:ListItem>FEMALE</asp:ListItem>
                <asp:ListItem>MALE</asp:ListItem>
            </asp:DropDownList>
        </p>
        <p>
            DEPARTMENT<asp:RadioButtonList ID="RadioButtonList1" runat="server">
                <asp:ListItem>IT</asp:ListItem>
                <asp:ListItem>HR</asp:ListItem>
                <asp:ListItem>FINANCE</asp:ListItem>
            </asp:RadioButtonList>
        </p>
        <p>
            <asp:CheckBox ID="CheckBox1" runat="server" />
        </p>
    </form>
</body>
</html>
```

&nbsp;I ACCEPT THE TERMS AND CONDITIONS

```
<p>
    <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="REGISTER" />
</p>
<p>
    &nbsp;</p>
<p>
    &nbsp;</p>
</form>
</body>
</html>
```

### WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

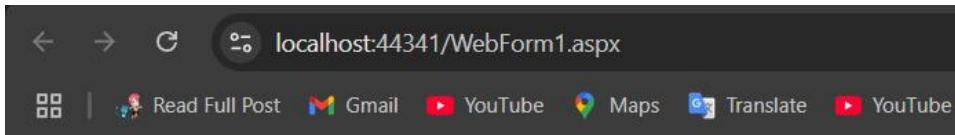
```
namespace PR_1
```

```
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
}
```

```
protected void Button1_Click(object sender, EventArgs e)
{
    if (CheckBox1.Checked)
```

```
{  
    string FIRSTNAME = TextBox1.Text;  
    string LASTNAME = TextBox2.Text;  
    string EMAIL = TextBox3.Text;  
    string DATEOFBIRTH = TextBox4.Text;  
    string GENDER = DropDownList1.SelectedValue;  
    string DEPARTMENT = RadioButtonList1.SelectedValue;  
  
    // Display confirmation message  
    Response.Write($"<h3>Registration Successful!</h3>");  
    Response.Write($"<p>Name: {FIRSTNAME} {LASTNAME}</p>");  
    Response.Write($"<p>Email: {EMAIL}</p>");  
    Response.Write($"<p>Date of Birth: {DATEOFBIRTH}</p>");  
    Response.Write($"<p>Gender: {GENDER}</p>");  
    Response.Write($"<p>Department: {DEPARTMENT}</p>");  
}  
else  
    Response.Write("<h3 style='color:red'>Please accept the terms and conditions.</h3>");  
}  
}  
}
```

**OUTPUT:**



### REGISTRATION FORM

FIRST NAME :

LAST NAME :

EMAIL :

DATE OF BIRTH :

GENDER :

DEPARTMENT :

- IT
- HR
- FINANCE

I ACCEPT THE TERMS AND CONDITIONS

---

## Registration Form

First Name:

Last Name:

Email:

Date of Birth:

Gender:

HR

Department:  IT  
 Finance

I accept the terms and conditions

**Registration Successful!**

Name: ADIFA C24004

Email: ADIFA@GMAIL.COM

Date of Birth: 2025-05-28

Gender: Female

Department: IT

**Registration Form**

First Name:

Last Name:

Email:

Date of Birth:

Gender:

HR

IT

Finance

I accept the terms and conditions

## PRACTICAL-2

### Create a website using the master page concept

```

<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site1.master.cs" Inherits="practical_2.Site1" %>

<!DOCTYPE html>

<html>
<head runat="server">
<title>REALME </title>
<link rel="stylesheet" href="styles.css" />
</head>
<body>
<div class ="header">
    <h1> WELCOME TO REALME</h1>
    <nav>
        <a href="Home.aspx">Home</a>
        <a href="About.aspx">About</a>
        <a href="Contact.aspx">Contact</a>
        <a href="Login.aspx">Login</a>
    </nav>
</div>

<asp:ContentPlaceHolder ID="head" runat="server">
</asp:ContentPlaceHolder>

<form id="form1" runat="server">
<div>
    <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
        </asp:ContentPlaceHolder>
    </div>
    <div class ="footer">
        <p>&copy; REALME All Rights are Reserved </p>
    </div>
</form>
</body>
</html>

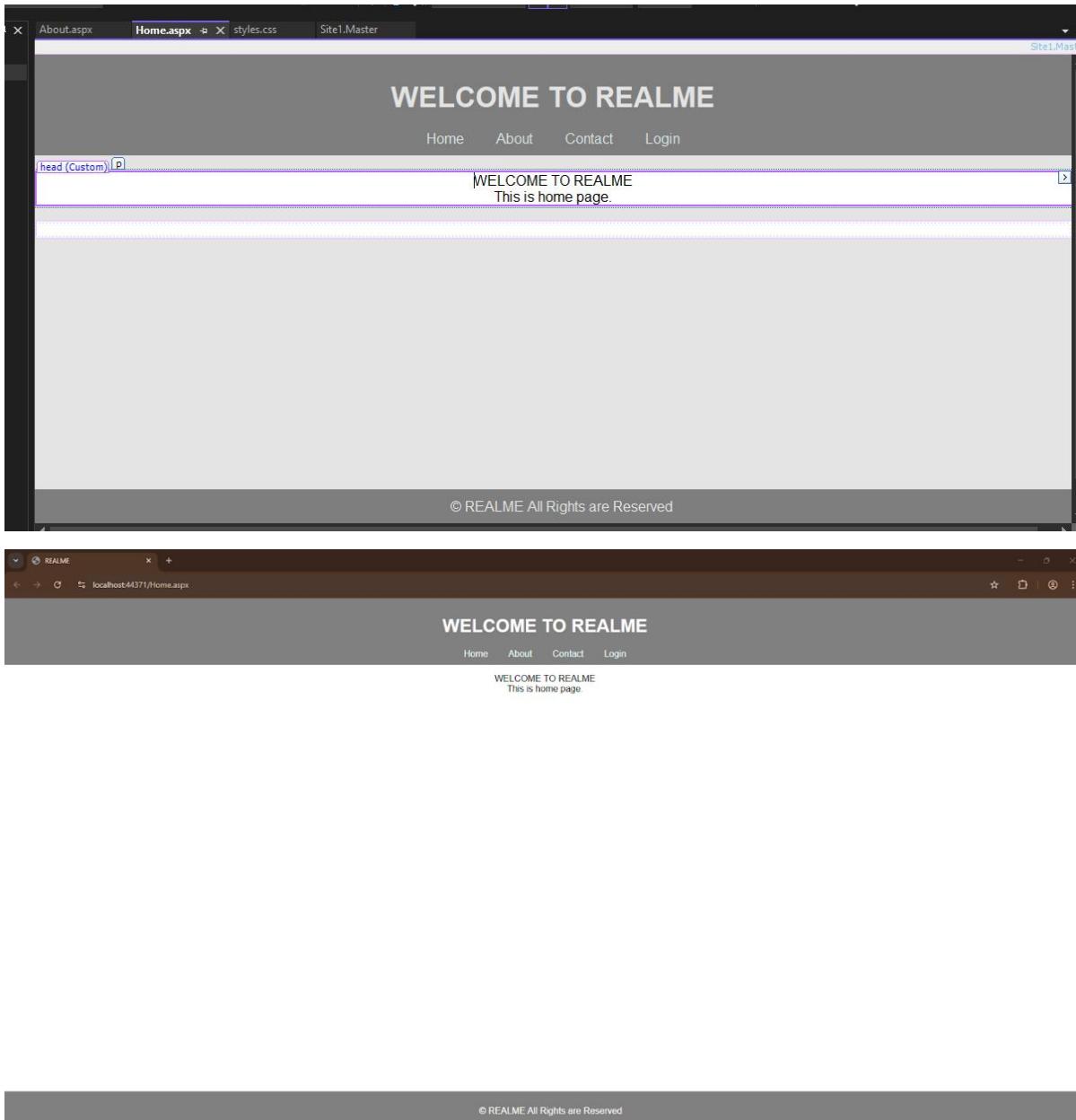
```

#### Home.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master" AutoEventWireup="true" CodeBehind="Home.aspx.cs"
Inherits="practical_2.WebForm1" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
    <p style="text-align:center" >
        WELCOME TO REALME
    <br />
        This is home page.
    </p>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
</asp:Content>

```



## About.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master" AutoEventWireup="true" CodeBehind="About.aspx.cs"
Inherits="practical_2.About" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
    <p style="text-align:center">
        REALME is a brand made for the young generation.
        <br /> We understand young users' expectations for tech,
        <br /> and exceed it by enabling them to more quickly
        <br /> experience advanced technology and leading performance.
    </p>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
</asp:Content>
```

## WELCOME TO REALME

[Home](#)   [About](#)   [Contact](#)   [Login](#)

REALME is a brand made for the young generation.  
We understand young users' expectations for tech,  
and exceed it by enabling them to more quickly  
experience advanced technology and leading performance.

## PRACTICAL-3

### Design a Web Application using advanced controls.

#### PRACTICAL NO. 3

##### Design a web application using advanced controls.

###### 1. Ad Rotator

Create a new project and add webform, XML file and images.

After adding a AdRotator in the web form Add the xml file in the AdRotator.

###### Webform.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="pract3.WebForm1" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
    <title></title>
```

```
</head>
```

```
<body style="font-weight: 700">
```

```
    <form id="form1" runat="server">
```

```
        <div>
```

```
            &nbsp;Ad Rotator example<br />
```

```
            <br />
```

```
            <asp:AdRotator ID="AdRotator1" runat="server" DataSourceID="XmlDataSource1"
OnAdCreated="AdRotator1_AdCreated" />
```

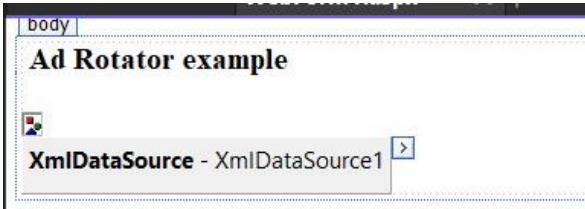
```
            <asp:XmlDataSource ID="XmlDataSource1" runat="server"
DataFile="~/XMLFile1.xml"></asp:XmlDataSource>
```

```
        </div>
```

```
    </form>
```

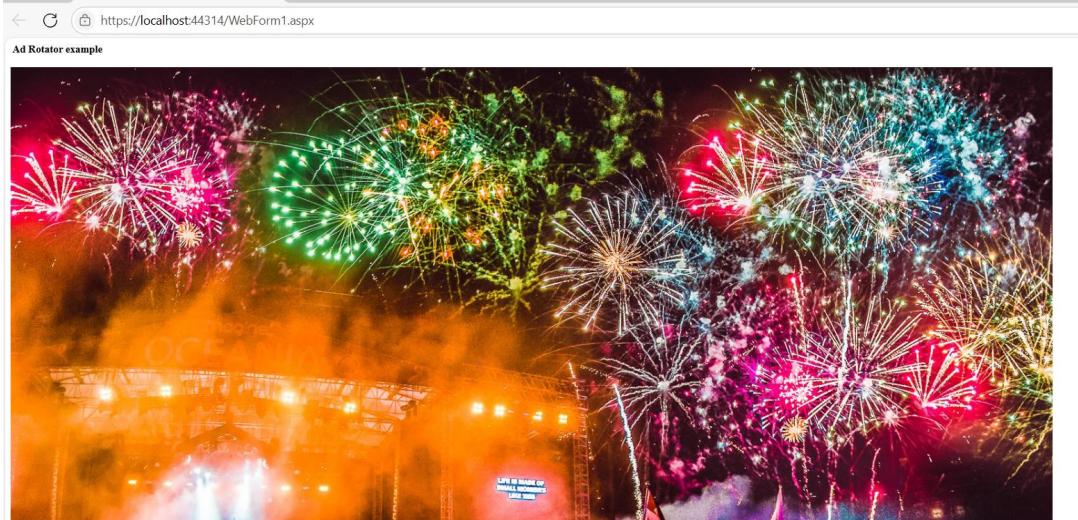
```
</body>
```

```
</html>
```



### Xmlfile.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
    <Ad>
        <ImageUrl>image2.jpeg</ImageUrl>
        <NavigateUrl>google.com</NavigateUrl>
        <AlternateText>First Ad</AlternateText>
        <Impressions>50</Impressions>
    </Ad>
    <Ad>
        <ImageUrl>image3.jpeg</ImageUrl>
        <NavigateUrl>gmail.com</NavigateUrl>
        <AlternateText>Second Ad</AlternateText>
        <Impressions>30</Impressions>
    </Ad>
</Advertisements>
```



← ⌂ https://localhost:44314/WebForm1.aspx

### Ad Rotator example

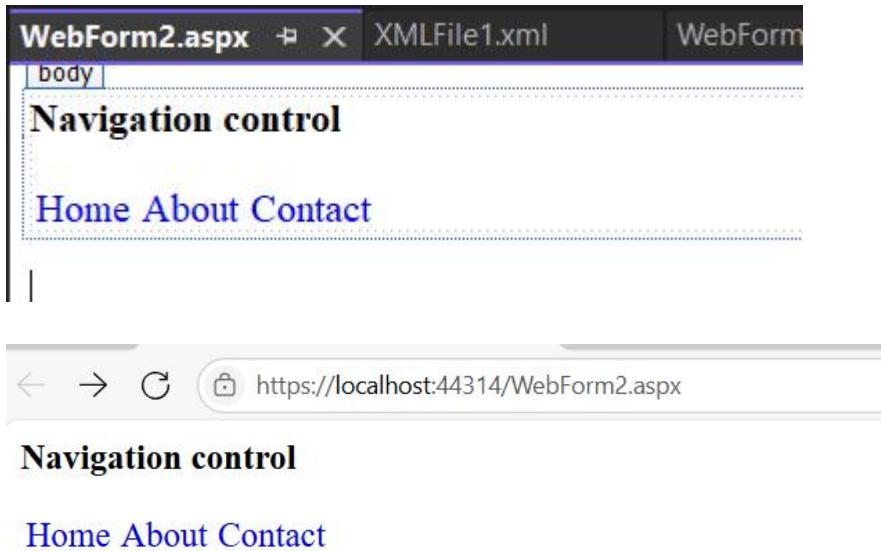


## 2. Navigation Control

### Webform.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs"  
Inherits="pract3.WebForm2" %>
```

```
<!DOCTYPE html>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
    <title></title>  
</head>  
<body>  
    <form id="form1" runat="server">  
        <div>  
            <strong>Navigation control<br />  
            <br />  
            </strong>  
            <asp:Menu ID="Menu1" runat="server" Orientation="Horizontal">  
                <Items>  
                    <asp:MenuItem Text="Home" NavigateUrl="Home.aspx"/>  
                    <asp:MenuItem Text="About" NavigateUrl="About.aspx"/>  
                    <asp:MenuItem Text="Contact" NavigateUrl="Contact.aspx"/>  
                </Items>  
            </asp:Menu>  
        </div>  
    </form>  
</body>  
</html>
```



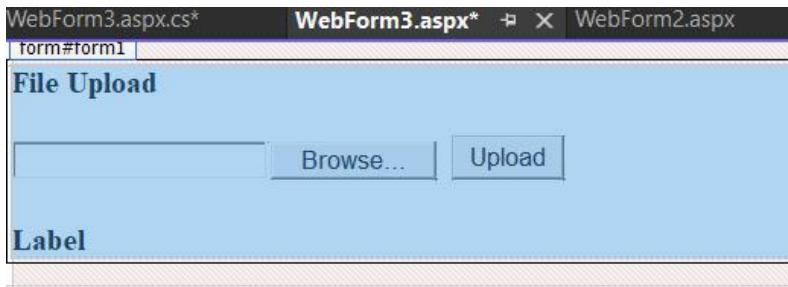
### 3. Upload File

#### Webform.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm3.aspx.cs"
Inherits="pract3.WebForm3" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div style="font-weight: 700">
            File Upload<br />
            <br />
            <asp:FileUpload ID="FileUpload1" runat="server" />
        &nbsp;
            <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Upload" />
            <br />
            <br />
            <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>
</html>
```



### Webform.aspx.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace pract3
{
    public partial class WebForm3 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            if (FileUpload1.HasFile)
            {
                try
                {
                    string filename = Path.GetFileName(FileUpload1.FileName);

                    Label1.Text = "Upload status: File uploaded successfully!";
                }
                catch (Exception ex)
                {
                    Label1.Text = "Upload status: Error - " + ex.Message;
                }
            }
            else
            {
                Label1.Text = "Upload status: No file selected.";
            }
        }
    }
}
```

```
}
```

The screenshot shows a web browser window with the URL <https://localhost:44314/WebForm3.aspx>. The page title is "File Upload". It features a file input field labeled "Choose File" with the placeholder "No file chosen" and a "Upload" button. Below the form, a success message reads "Upload status: File uploaded successfully!".

← ⏪ https://localhost:44314/WebForm3.aspx

**File Upload**

Choose File No file chosen

Upload

**Upload status: File uploaded successfully!**

## PRACTICAL-4

### Webpage Demonstrating Connection-Oriented Architecture (ASP.NET Web Forms with SQL Server Database)

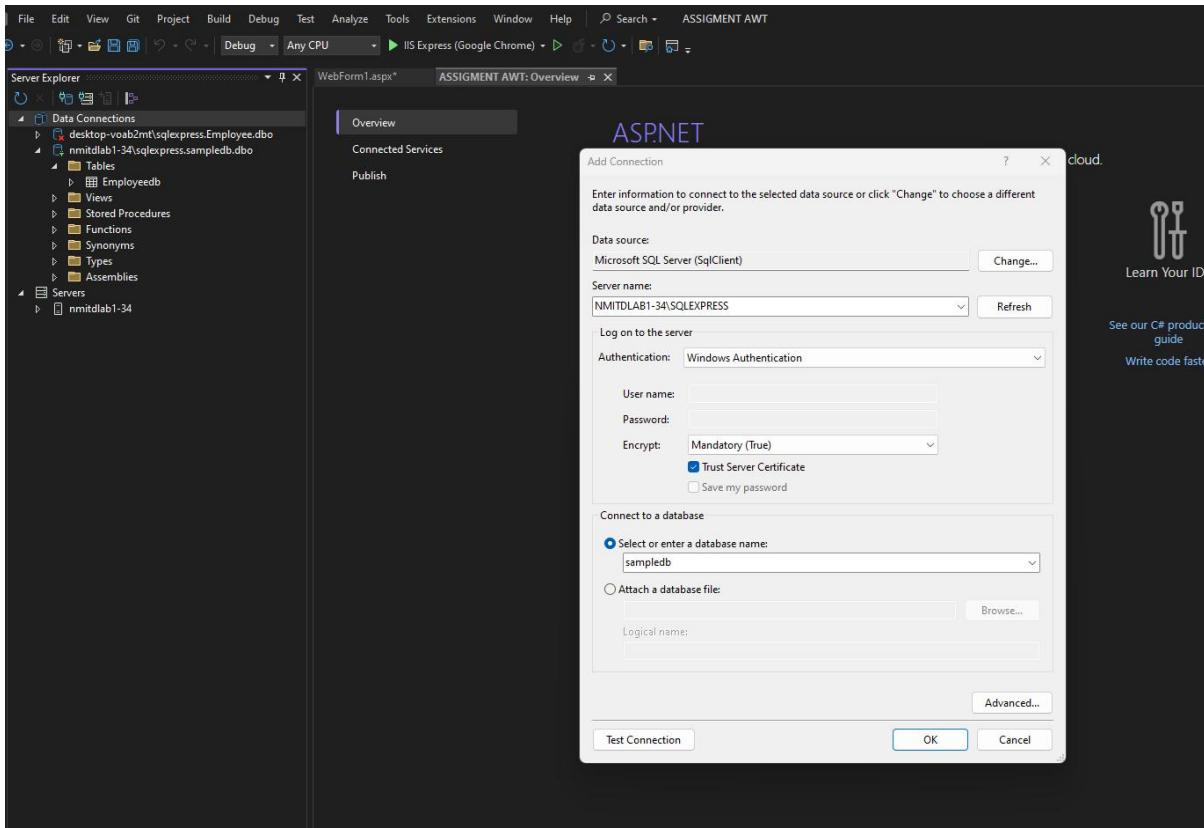
The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer pane displays the database structure of 'NMITDLAB1-34\SQLEXPRESS'. In the center, a query window titled 'SQLQuery3.sql' contains the following T-SQL code:

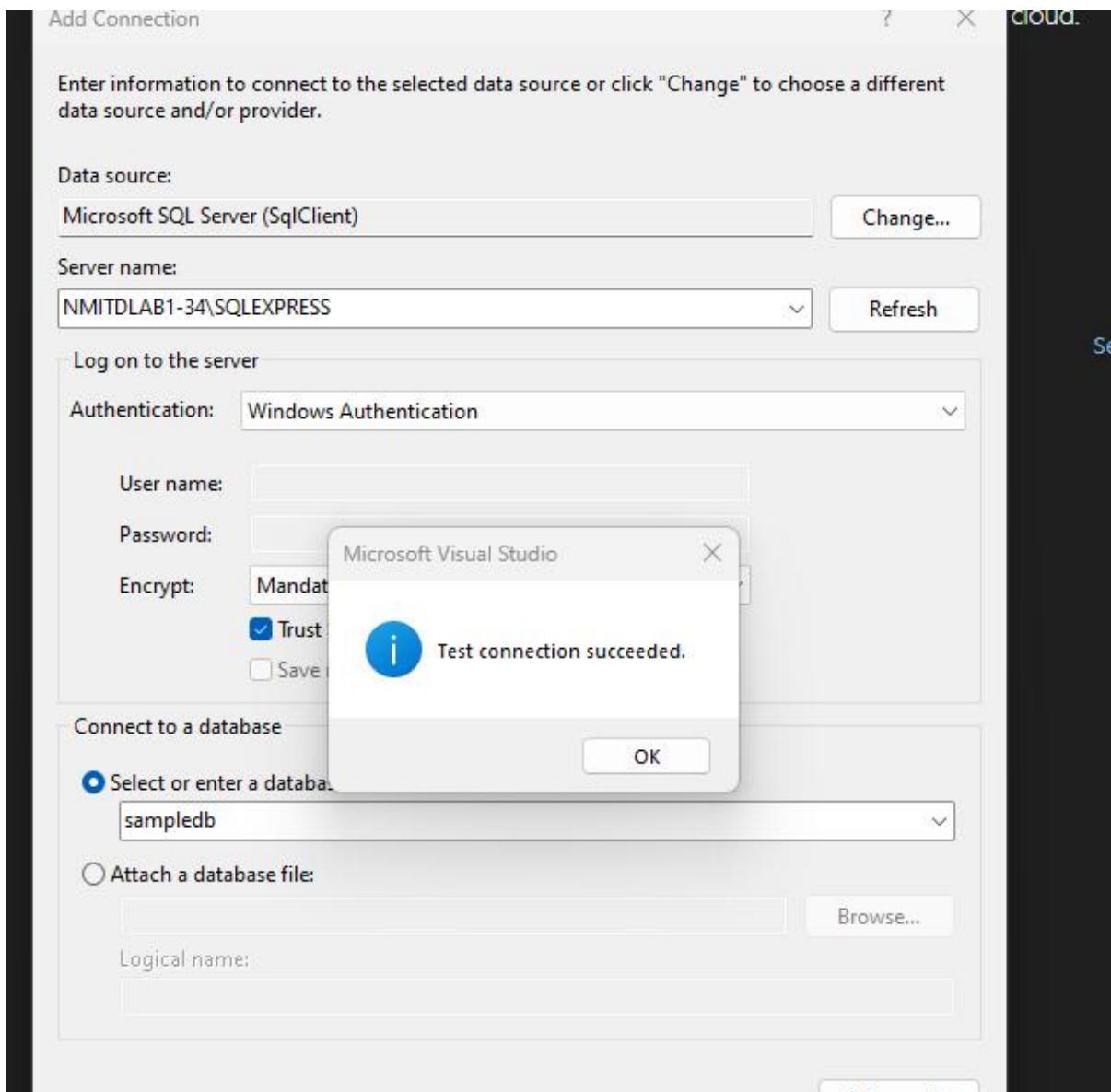
```
SELECT TOP (1000) [ID]
      ,[Name]
      ,[Email]
  FROM [sampledb].[dbo].[Employeedb]
```

The results pane below shows the output of the query:

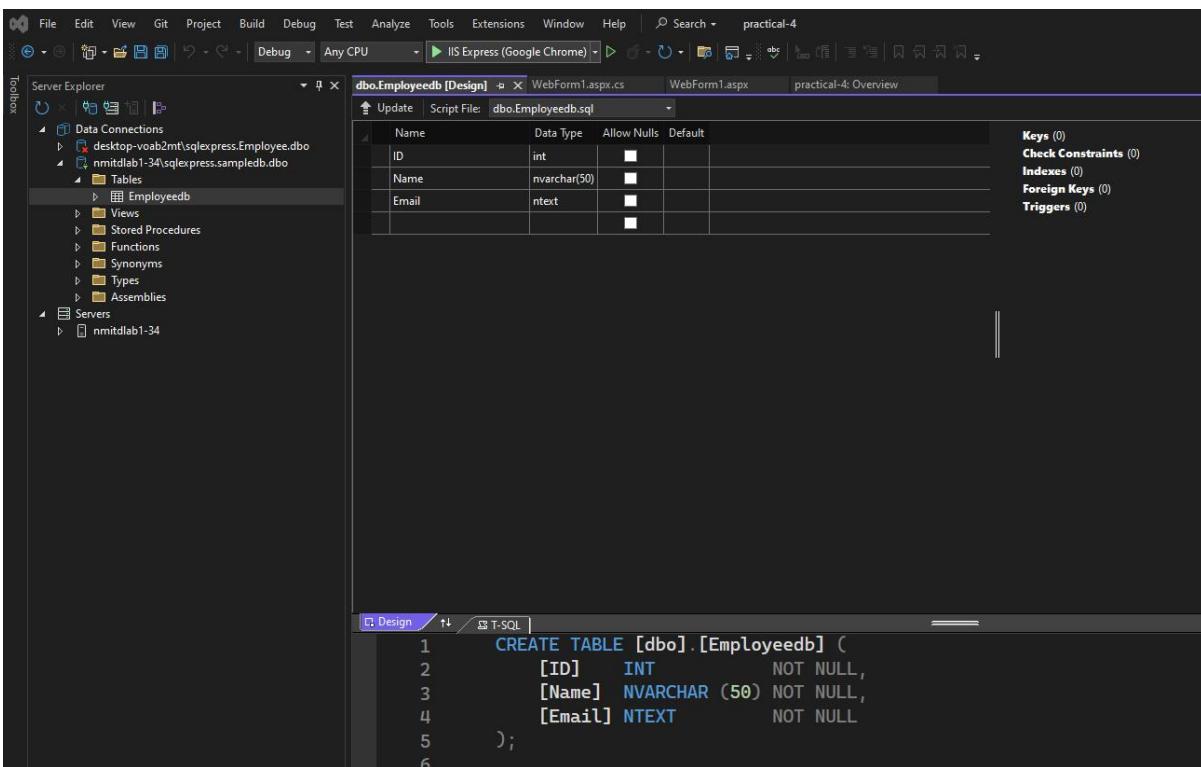
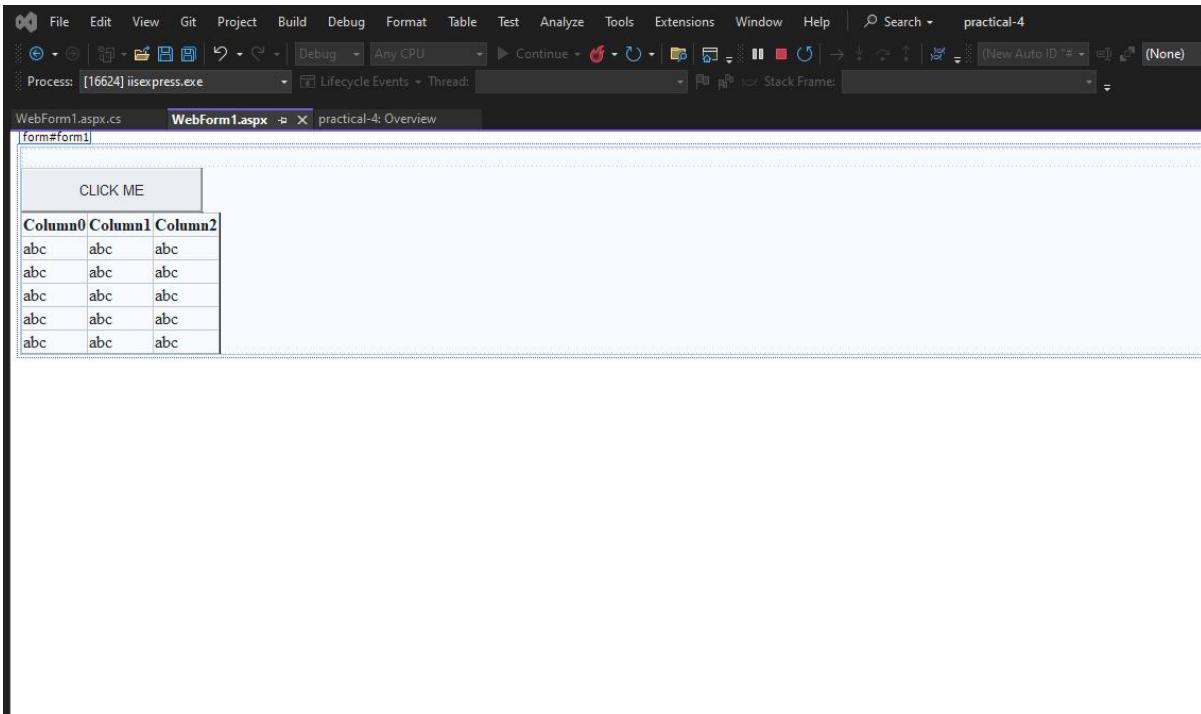
| ID | Name    | Email             |
|----|---------|-------------------|
| 1  | Nutan   | nutan@email.com   |
| 2  | Aleena  | aleena@email.com  |
| 3  | Krishna | krishna@email.com |
| 4  | Soham   | soham@email.com   |

#### STEP 1: CREATE SQL SERVER TABLE





STEP 2:



```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="practical_4.WebForm1" %>
```

```
<!DOCTYPE html>
```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

```

```
</div>
<asp:Button ID="Button1" runat="server" Height="42px" OnClick="Button1_Click" Text="CLICK ME" Width="171px" />
<asp:GridView ID="GridView1" runat="server">
</asp:GridView>
</form>
</body>
</html>
```

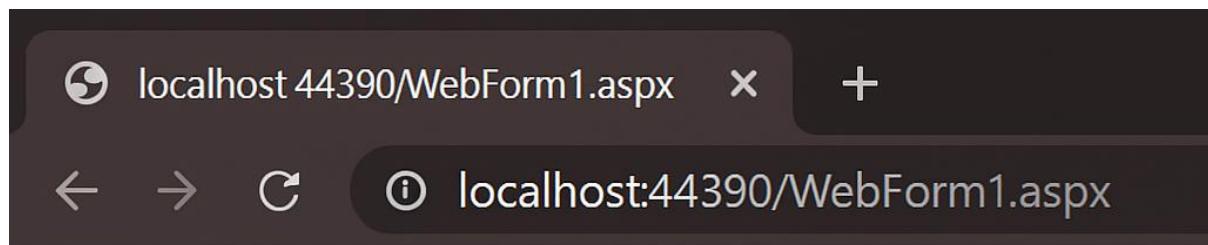
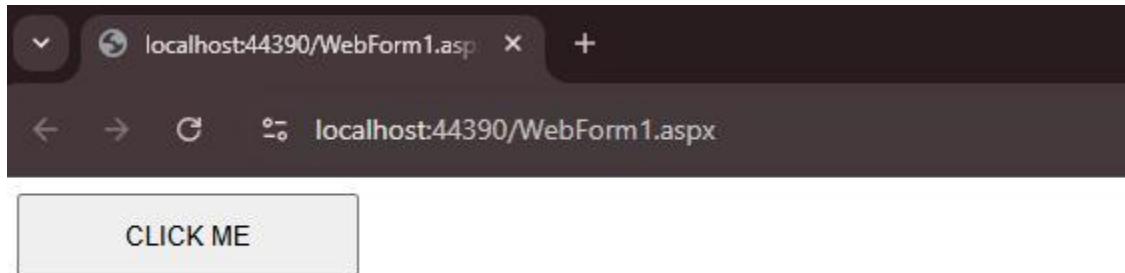
## STEP 3

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace practical_4
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            string connectionString = "Data Source=NMITDLAB1-34\\SQLEXPRESS;Initial Catalog=sampledb;Integrated Security=True";
            using (SqlConnection conn = new SqlConnection(connectionString))
            {
                try
                {
                    conn.Open();
                    string query = "SELECT * FROM EmployeeDb";
                    SqlDataAdapter da = new SqlDataAdapter(query, conn);
                    DataTable dt = new DataTable();
                    da.Fill(dt);
                    GridView1.DataSource = dt;
                    GridView1.DataBind();
                }
                catch (Exception ex)
                {
                    Response.Write("<script>alert('Error:' + ex.Message + '')</script>");
                }
            }
        }
    }
}
```



| ID | Name  | Email           |
|----|-------|-----------------|
| 1  | Siri  | siri@email.com  |
| 2  | John  | john@email.com  |
| 3  | Katie | katie@email.com |
| 4  | Sandy | sandy@email.com |

## PRACTICAL-5

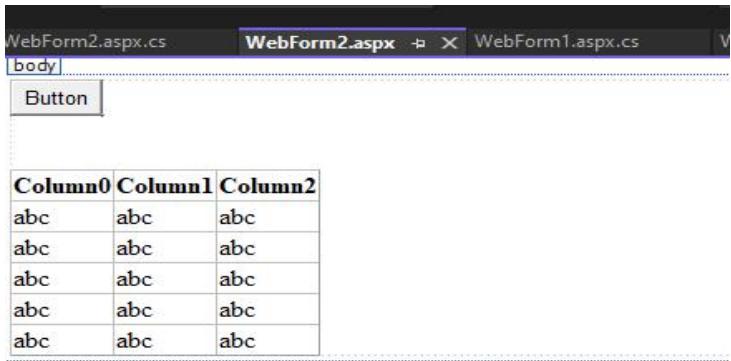
### Webpage Demonstrating Disconnected Architecture (ASP.NET Web Forms with SQL Server Database)

#### Webform.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs"
Inherits="PRAC_7_042.WebForm2" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Button" />
            <br />
            <br />
            <asp:GridView ID="GridView1" runat="server">
            </asp:GridView>
        </div>
    </form>
</body>
</html>
```



### Webform.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

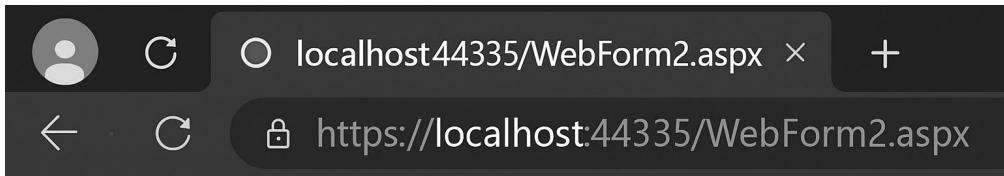
namespace PRAC_7_042
{
    public partial class WebForm2 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            string connStr = "Data Source=NMITDLAB1-33\\SQLEXPRESS;Initial Catalog=Sampledb;Integrated Security=True";
        }
    }
}
```

```
SqlDataAdapter da;
DataSet ds = new DataSet();
try
{
    using (SqlConnection conn = new SqlConnection(connStr))
    {
        string query = "Select * from Employee";
        da = new SqlDataAdapter(query, conn);
        da.Fill(ds, "Employee");
    }
    GridView1.DataSource = ds.Tables["Employee"];
    GridView1.DataBind();
}

catch (Exception ex)
{
    Response.Write("<script>alert('Error: " + ex.Message + "');</script>");
}
}
```



## Disconnected Architecture

Button

| ID | Name  | Email           |
|----|-------|-----------------|
| 1  | Siri  | siri@email.com  |
| 2  | John  | john@email.com  |
| 3  | Katie | katie@email.com |
| 4  | Sandy | sandy@email.com |

## PRACTICAL-6

Create a webpage that demonstrates the use of data bound controls of ASP.NET.

In SSMS, create a table and insert values into it

Open Visual Studio, create a new project and connect it to sever through Server explorer

Add a webform and the add a datalist into that webform .

Now add SQL data source to the datalist and also select the table name and columns from it.

### Webform.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm3.aspx.cs"
Inherits="prac2.WebForm3" %>

<!DOCTYPE html>

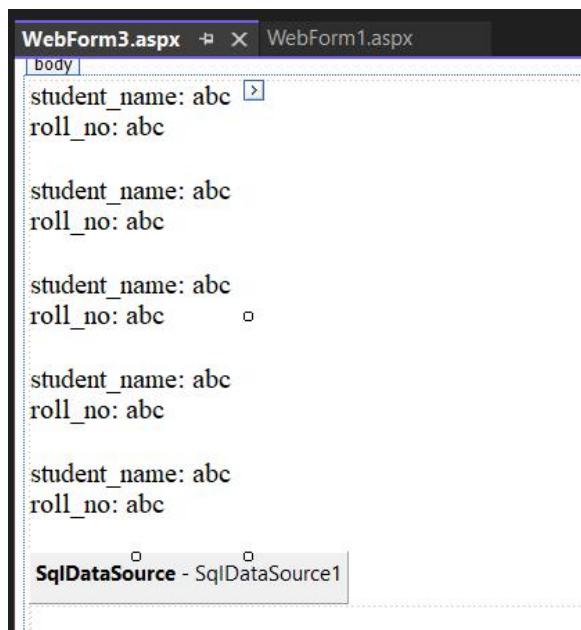
<html xmlns="http://www.w3.org/1999/xhtml">

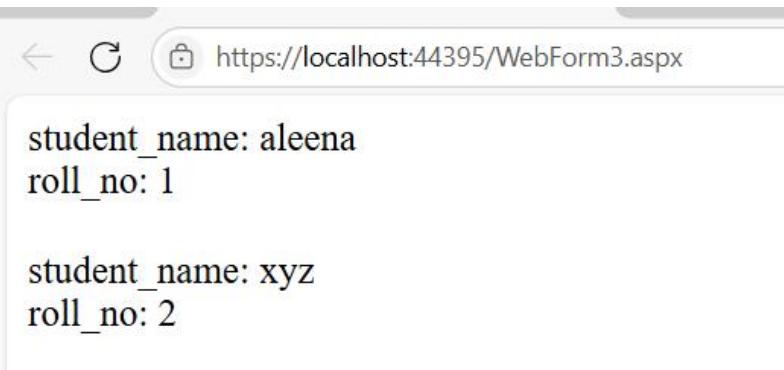
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
```

```
<asp:DataList ID="DataList1" runat="server" DataSourceID="SqlDataSource1" Width="155px">
    <ItemTemplate>
        student_name:
        <asp:Label ID="student_nameLabel" runat="server" Text='<%# Eval("student_name") %>' />
        <br />
        roll_no:
        <asp:Label ID="roll_noLabel" runat="server" Text='<%# Eval("roll_no") %>' />
        <br />
    <br />
    </ItemTemplate>
</asp:DataList>

<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:pract1ConnectionString %>" 
ProviderName="<%$ ConnectionStrings:pract1ConnectionString.ProviderName %>" 
SelectCommand="SELECT [student_name], [roll_no] FROM [slist]"></asp:SqlDataSource>

<div>
</div>
</form>
</body>
</html>
```



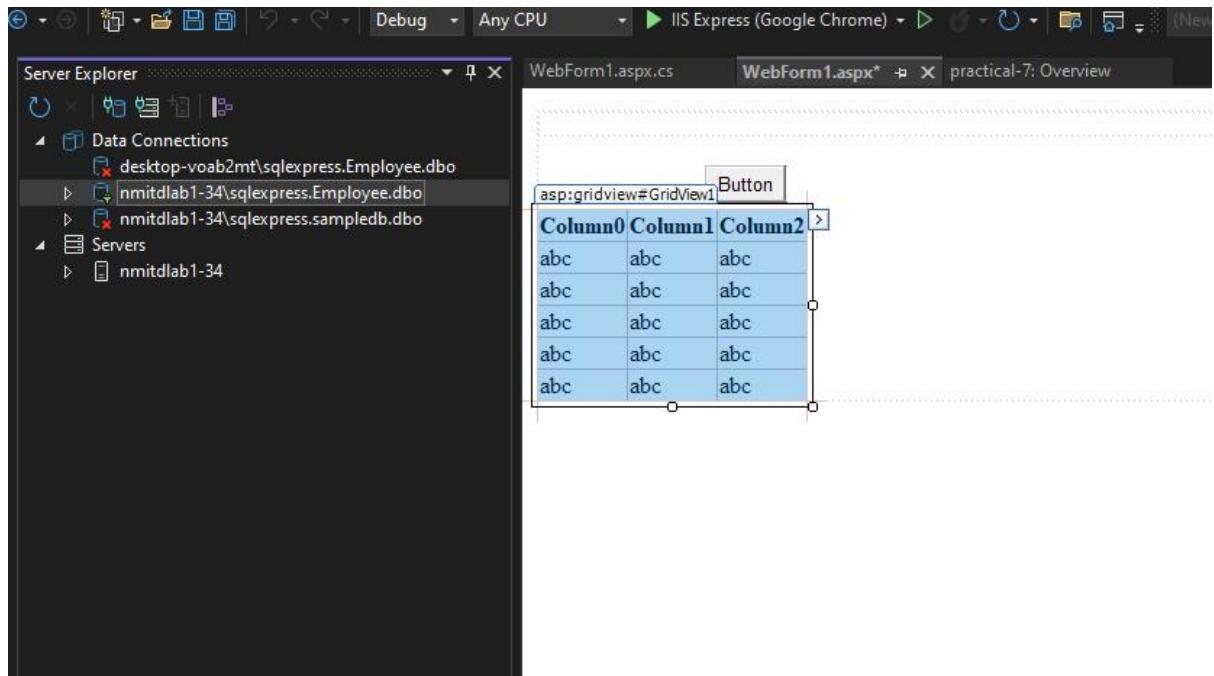


A screenshot of a web browser window. The address bar shows the URL <https://localhost:44395/WebForm3.aspx>. The page content displays two rows of student information:

| student_name | roll_no |
|--------------|---------|
| aleena       | 1       |
| xyz          | 2       |

### **PRACTICAL-7**

**Design a webpage to demonstrate the working of a simple stored procedure.**



```
-- Create Procedure
-- 
-- Use the Specified command (Ctrl+Shift+F2) to
-- values below.
-- 
-- This block of code is part of the definition.
-- =====
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author: [REDACTED]
-- Create date: [REDACTED]
-- Description: [REDACTED]
-- =====
PROCEDURE [dbo].[GetUsers]
    -- Add the parameters for the stored procedure here
    @Param1, [REDACTED]
    @Param2, [REDACTED]
```

## 1. CREATE THE STORED PROCEDURE IN SQL SERVER

Run this SQL script in SQL Server Management Studio (SSMS):

```
CREATE PROCEDURE GetUsers
As
BEGIN
SELECT emp_id, emp_name FROM Employee
END
```

```
CREATE PROCEDURE GetUsers
As
BEGIN
    SELECT emp_id, emp_name FROM Employee
END
```

```
CREATE PROCEDURE GetUsers
As
BEGIN
    SELECT emp_id, emp_name FROM Employee
END
```

200 %

Messages

Commands completed successfully.

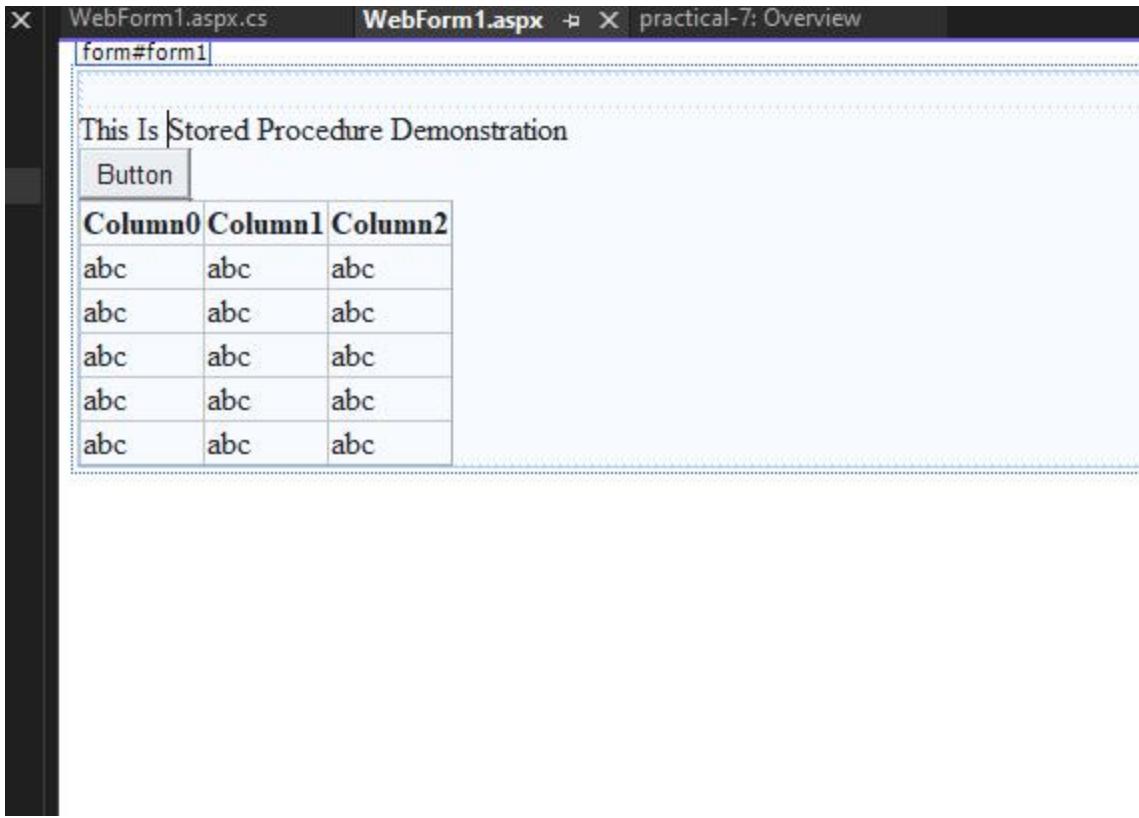
Completion time: 2025-03-10T15:46:30.7522632+05:30

## 2. CREATE AN ASP.NET WEB APPLICATION IN VISUAL STUDIO

- OPEN visual studio
- CREATE a new ASP>NET web forms application
- ADD a Web Form (Default.aspx)

## 3. DESIGN THE WEB FORM (Default.aspx)

Modify the WebForm1.aspx file;



```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="practical_7.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
        </div>

        &nbsp;This Is Stored Procedure Demonstration<br />
        <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Button" />
        <asp:GridView ID="GridView1" runat="server">
        </asp:GridView>
    </form>
</body>
</html>
```

#### 4. CODE BEHIND FILE (Default.aspx.cs)

Modify Default.aspx.cs to execute the stored procedure and display results in GridView:

```
using System;
using System.Collections.Generic;
```

```
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace practical_7
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            string connStr = "Data Source=NMITDLAB1-34\\SQLEXPRESS;Initial Catalog=Employee;Integrated Security=True";
            using (SqlConnection conn
                  = new SqlConnection(connStr))
            {
                using (SqlCommand cmd = new SqlCommand("GetUsers, conn"))
                {
                    cmd.CommandType = CommandType.StoredProcedure;
                    conn.Open();
                    SqlDataAdapter da = new SqlDataAdapter(cmd);
                    DataTable dt = new DataTable();
                    da.Fill(dt);

                    GridView1.DataSource = dt;
                    GridView1.DataBind();
                }
            }
        }
    }
}
```

## PRACTICAL-8

Design a webpage to demonstrate the working of parameterized stored procedure.

## 1. SQL Server

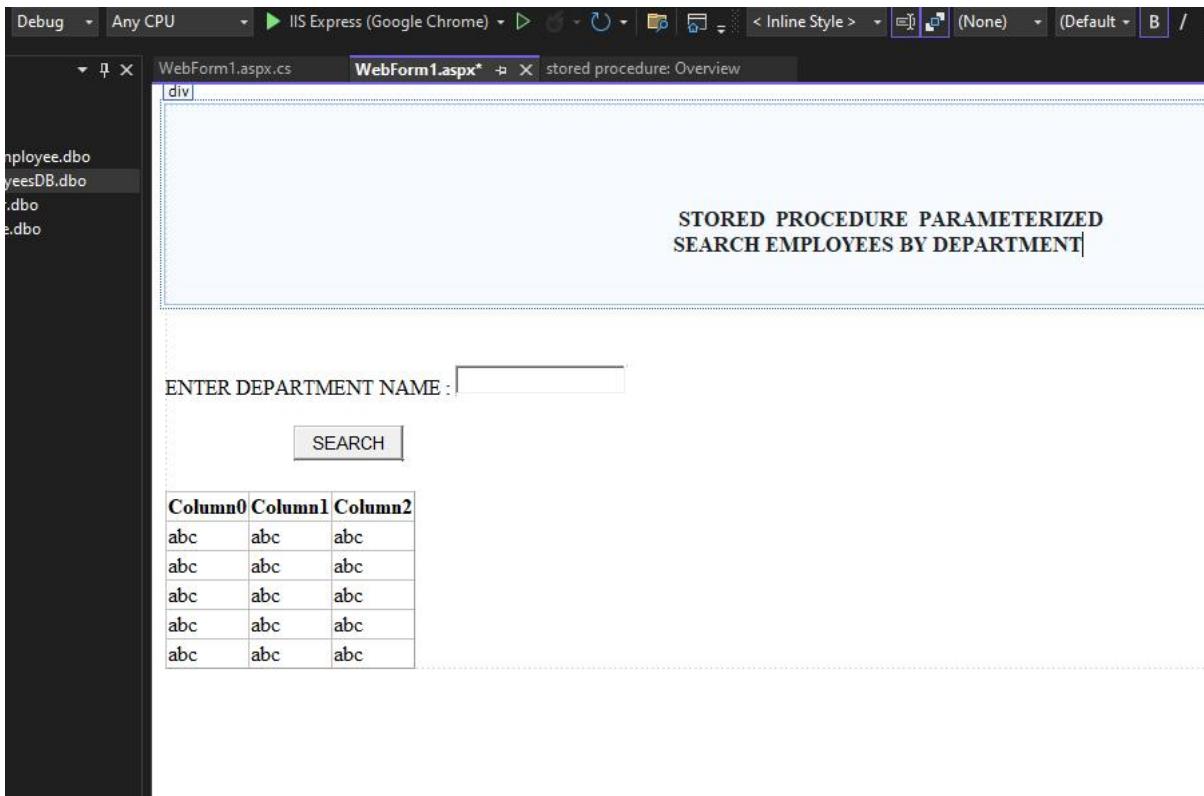
The screenshot shows the SSMS interface with a query window titled 'NMITDLAB1:3A\SQLExpress - dbo.table2'. The table contains the following data:

|   | EmplID | Name    | Department | Salary |
|---|--------|---------|------------|--------|
| 1 | 1      | Siri    | IT         | 30000  |
| 2 | 2      | John    | finance    | 35000  |
| 3 | 3      | Katie   | Programmer | 40000  |
| 4 | 4      | Michael | HR         | 50000  |
| 5 | 5      | James   | Programmer | 45000  |
| 8 | 6      | Robert  | Programmer | 47000  |
| 7 | 7      | Sandy   | finance    | 35000  |

The screenshot shows the SSMS interface with a query editor containing the following T-SQL code:

```
CREATE PROCEDURE GetEmployeeDepartment
    @DepartmentName CHAR(10)
AS
BEGIN
    SELECT * FROM table2 WHERE Department = @DepartmentName;
END;
```

The code is highlighted with syntax coloring. The status bar at the bottom shows 'Commands completed successfully.' and a completion time of 'Completion time: 2025-03-12T14:12:32.9516655+05:30'.



## 2. CREATE ASP.NET WEB FORM

- DESIGN A WEB FORM (WebForm1.aspx)

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="stored_procedure.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div style="font-weight: 700; height: 149px; margin-bottom: 46px">
            <br />
            <br />
            <br />
            <br />
        </div>
        STORED PROCEDURE PARAMETERIZED<br />
        SEARCH EMPLOYEES BY DEPARTMENT<br />
        <br />
        <br />
        <br />
        <br />
        <br />
        <p>        ENTER DEPARTMENT NAME : <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
        </p>
        <p>            <asp:Button ID="Button2" runat="server" OnClick="Button2_Click" Text="SEARCH" />
        </p>
        <p>            <asp:GridView ID="GridView1" runat="server">

```

```
</asp:GridView>
</p>
</form>
</body>
</html>
```

---

### 3. BACKEND CODE (Default.aspx.cs)

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace stored_procedure
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            String connStr = "Data Source=NMITDLAB1-34\\SQLEXPRESS;Initial Catalog=EmployeesDB;Integrated Security=True";
            {
                SqlConnection conn = new SqlConnection(connStr);
                SqlCommand cmd = new SqlCommand("GetDepartment", conn);
                cmd.CommandType = CommandType.StoredProcedure;
                cmd.Parameters.AddWithValue("@DepartmentName", TextBox1.Text);
                SqlDataAdapter da = new SqlDataAdapter(cmd);
                DataTable dt = new DataTable();
                da.Fill(dt);
                GridView1.DataSource = dt;
                GridView1.DataBind();
            }
        }
    }
}
```

Google Chrome isn't your default browser Set as default

**STORED PROCEDURE PARAMETERIZED  
SEARCH EMPLOYEES BY DEPARTMENT**

ENTER DEPARTMENT NAME :

**STORED PROCEDURE PARAMETERIZD  
SEARCH EMPLOYEES BY DEPARTMEN**

ENTER DEPARTMENT NAME:  it

| EmpID | Name | Department | Salary |
|-------|------|------------|--------|
| 1     | Siri | IT         | 30000  |
| 5     | Addy | IT         | 30000  |

**STORED PROCEDURE PARAMETERIZED  
SEARCH EMPLOYEES BY DEPARTMENT**

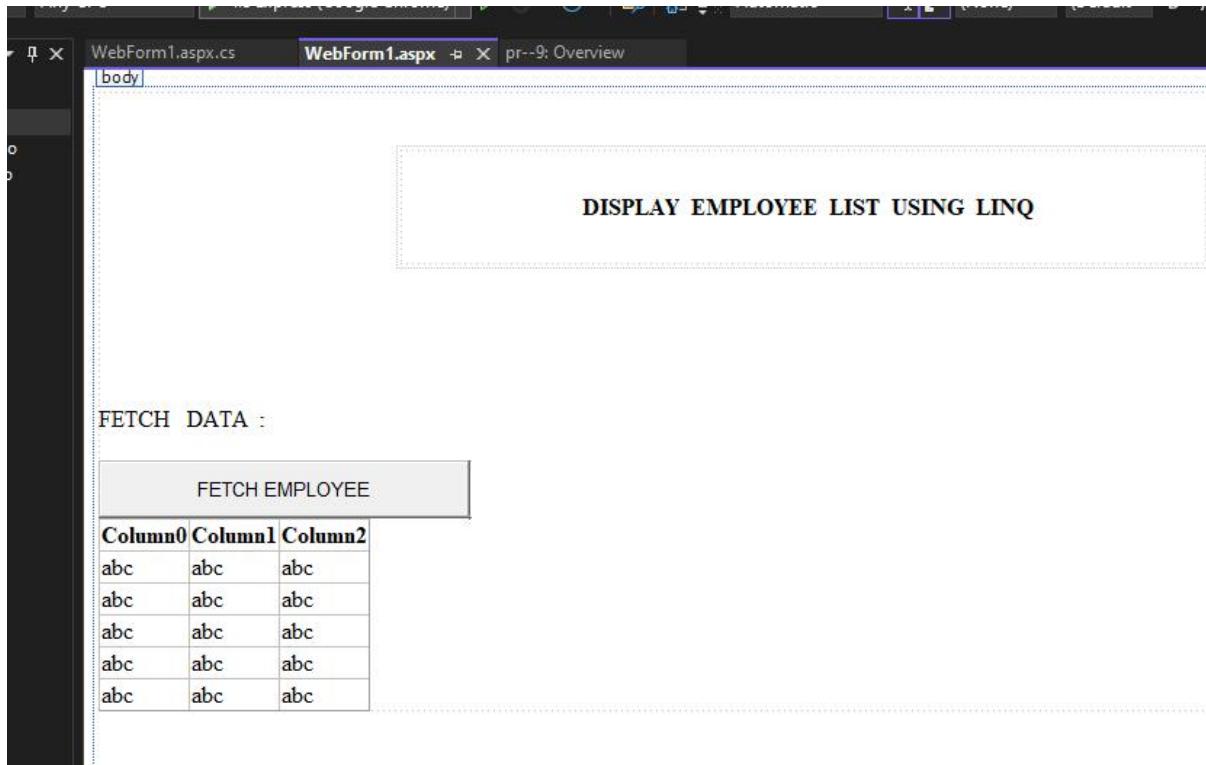
ENTER DEPARTMENT NAME :

| EmpID | Name  | Department | Salary |
|-------|-------|------------|--------|
| 4     | Adifa | HR         | 50000  |

## PRACTICAL-9

Design a webpage to display the use of LINQ.

1. Create ASP.NET Web Form(webForm1.aspx):



```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="pr_9.WebForm1" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
        .auto-style1 {
            width: 50%;
            height: 83px;
            margin-left: 206px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <br />
            <table class="auto-style1">
                <tr>
                    <td><strong>DISPLAY &nbsp; EMPLOYEE LIST USING LINQ</strong></td>
                </tr>
            </table>
            <br />
        </div>
    </form>
</body>
```

```

<br />
<br />
</div>
<p>
    FETCH DATA :</p>
<asp:Button ID="Button1" runat="server" Height="40px" OnClick="Button1_Click" Text="FETCH EMPLOYEE"
Width="256px" />
<asp:GridView ID="GridView1" runat="server">
</asp:GridView>
</form>
<p>
    &nbsp;</p>
</body>
</html>

```

---

## 2. BACKEND CODE (Default.aspx.cs)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

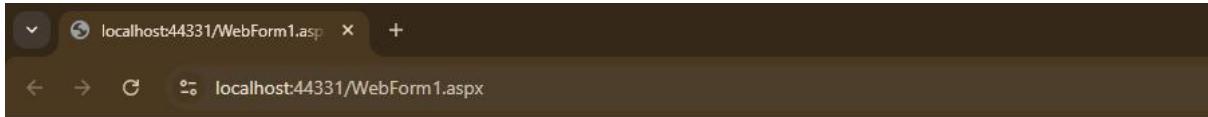
namespace pr__9
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        public class Employee
        {
            public int EmpID { get; set; }
            public string Name { get; set; }
            public string Department { get; set; }
            public int Salary { get; set; }
        }
        private List<Employee> employees = new List<Employee>
        {
            new Employee { EmpID = 1, Name = "Aleena", Department = "IT", Salary = 100000 },
            new Employee { EmpID = 2, Name = "Krishna", Department = "HR", Salary = 90000 },
            new Employee { EmpID = 3, Name = "Adifa", Department = "Designner", Salary = 80000 },
            new Employee { EmpID = 4, Name = "Sham", Department = "IT", Salary = 70000 }};

        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            var result = from emp in employees
                        where emp.Department == "Designner" && emp.Salary > 70000
                        select emp;
            GridView1.DataSource = result.ToList();
            GridView1.DataBind();
        }
    }
}

```



### DISPLAY EMPLOYEE LIST USING LINQ

FETCH DATA :

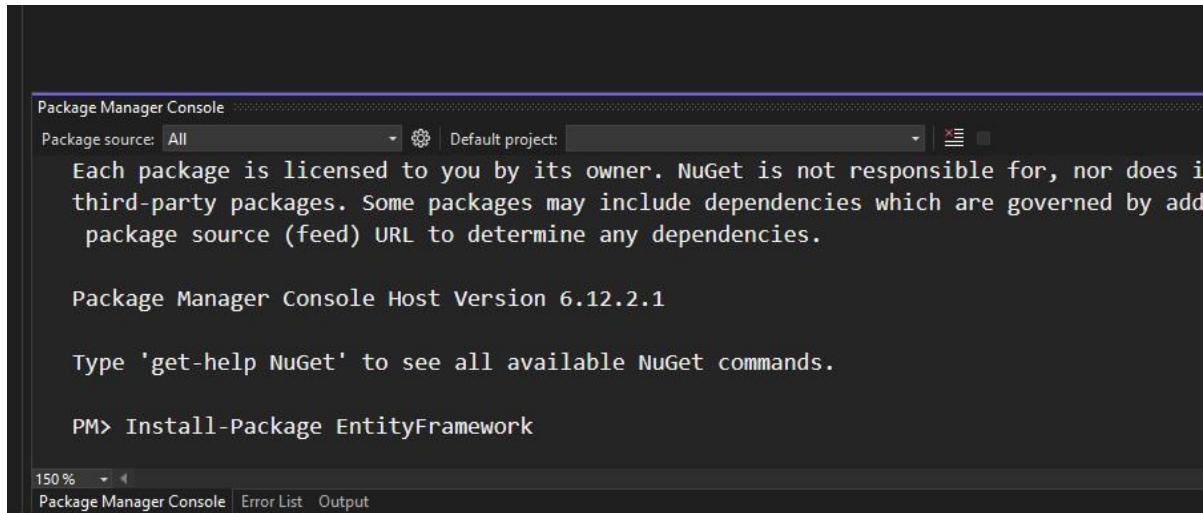
## Display Employee List using LINQ

| EmpID | Name  | Department | Salary |
|-------|-------|------------|--------|
| 3     | ADIFA | IT         | 65000  |

## PRACTICAL-10

**Build websites to demonstrate the working of entity frameworks in dot net.**

# SQL Server: Create a Database & Table



Package Manager Console

Package source: All Default project:

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it govern third-party packages. Some packages may include dependencies which are governed by additional package source (feed) URL to determine any dependencies.

Package Manager Console Host Version 6.12.2.1

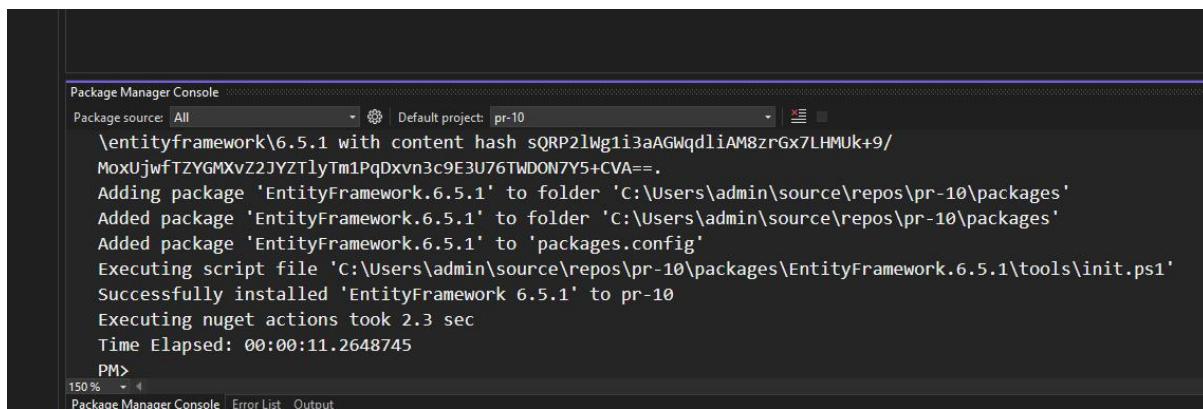
Type 'get-help NuGet' to see all available NuGet commands.

PM> Install-Package EntityFramework

150 %

Package Manager Console Error List Output

```
PM> Install-Package EntityFramework
```



Package Manager Console

Package source: All Default project: pr-10

\EntityFramework\6.5.1 with content hash sQRP2lWg1i3aAGWqdliAM8zrGx7LHMUK+9/MoxUjwfTZGMXvZJYZTlyTm1PqDxvn3c9E3U7GTWDON7Y5+CVA==.

Adding package 'EntityFramework.6.5.1' to folder 'C:\Users\admin\source\repos\pr-10\packages'

Added package 'EntityFramework.6.5.1' to folder 'c:\Users\admin\source\repos\pr-10\packages'

Added package 'EntityFramework.6.5.1' to 'packages.config'

Executing script file 'C:\Users\admin\source\repos\pr-10\packages\EntityFramework.6.5.1\tools\init.ps1'

Successfully installed 'EntityFramework 6.5.1' to pr-10

Executing nuget actions took 2.3 sec

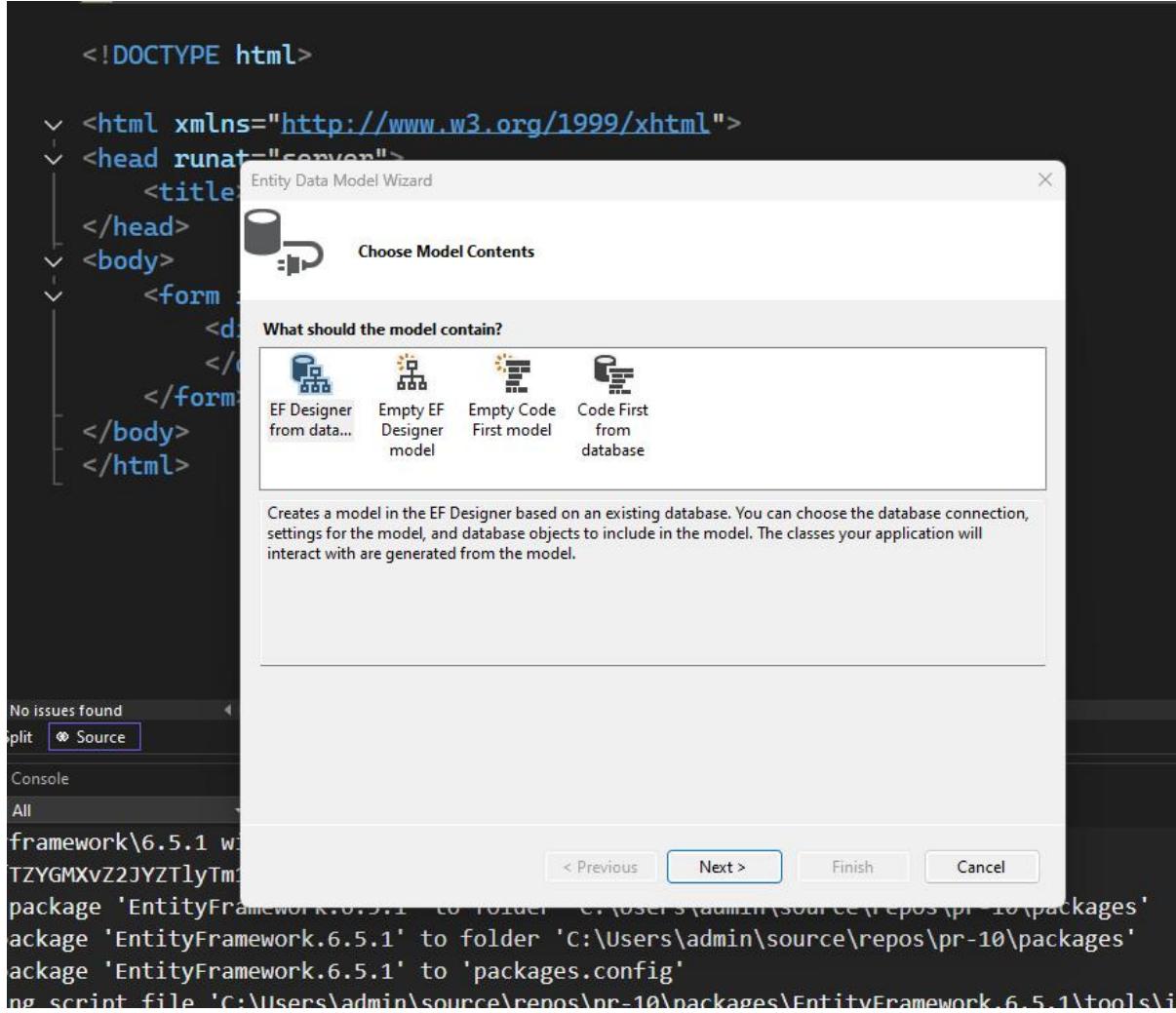
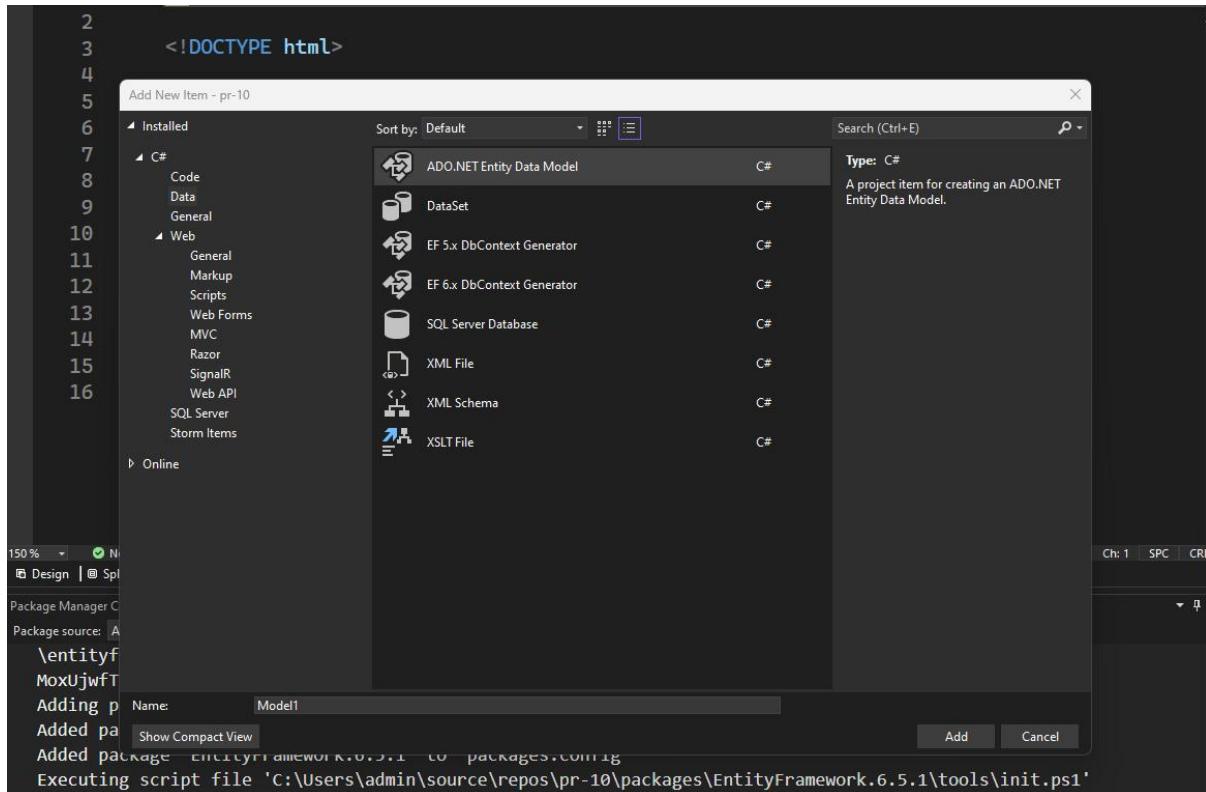
Time Elapsed: 00:00:11.2648745

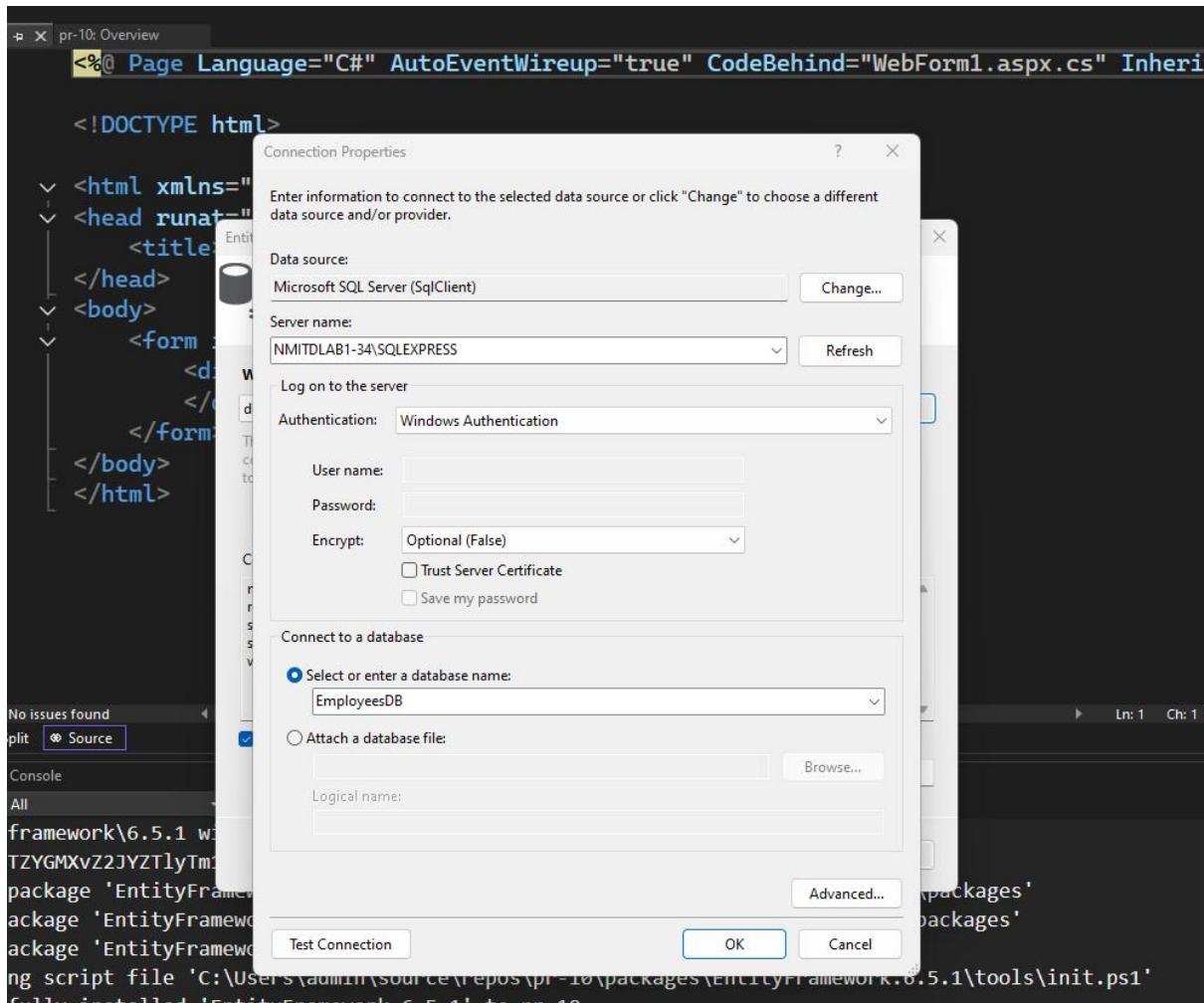
PM>

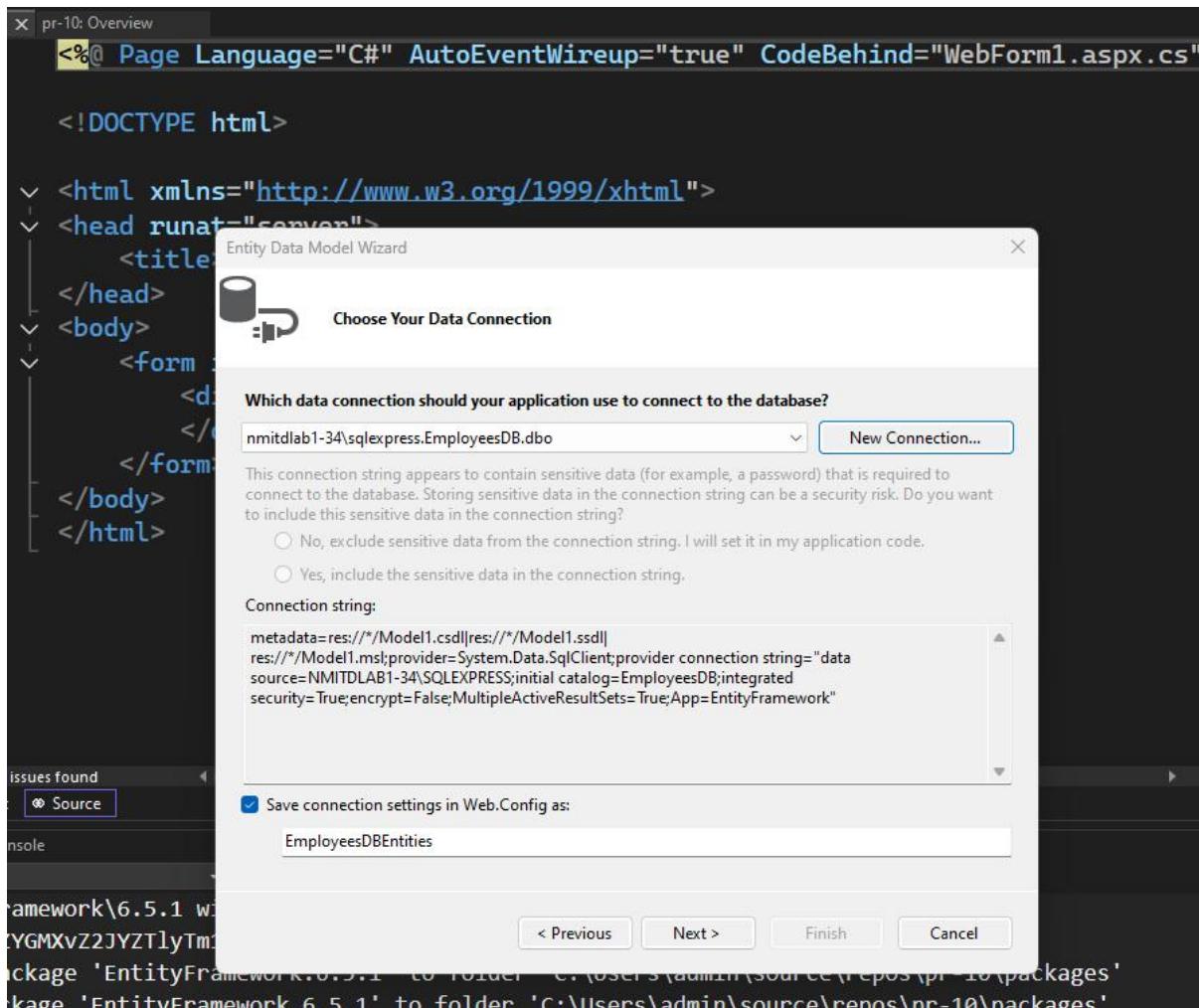
150 %

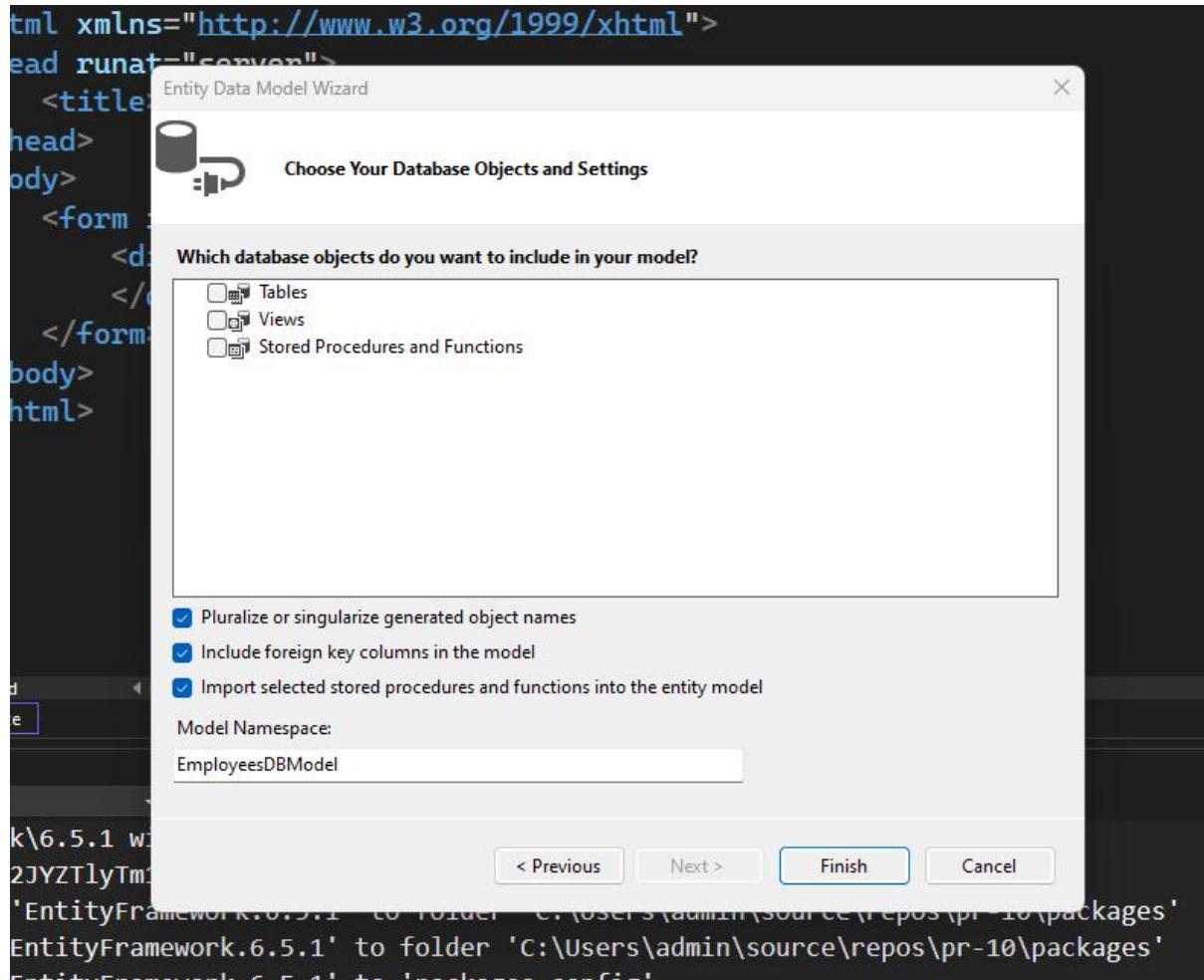
Package Manager Console Error List Output

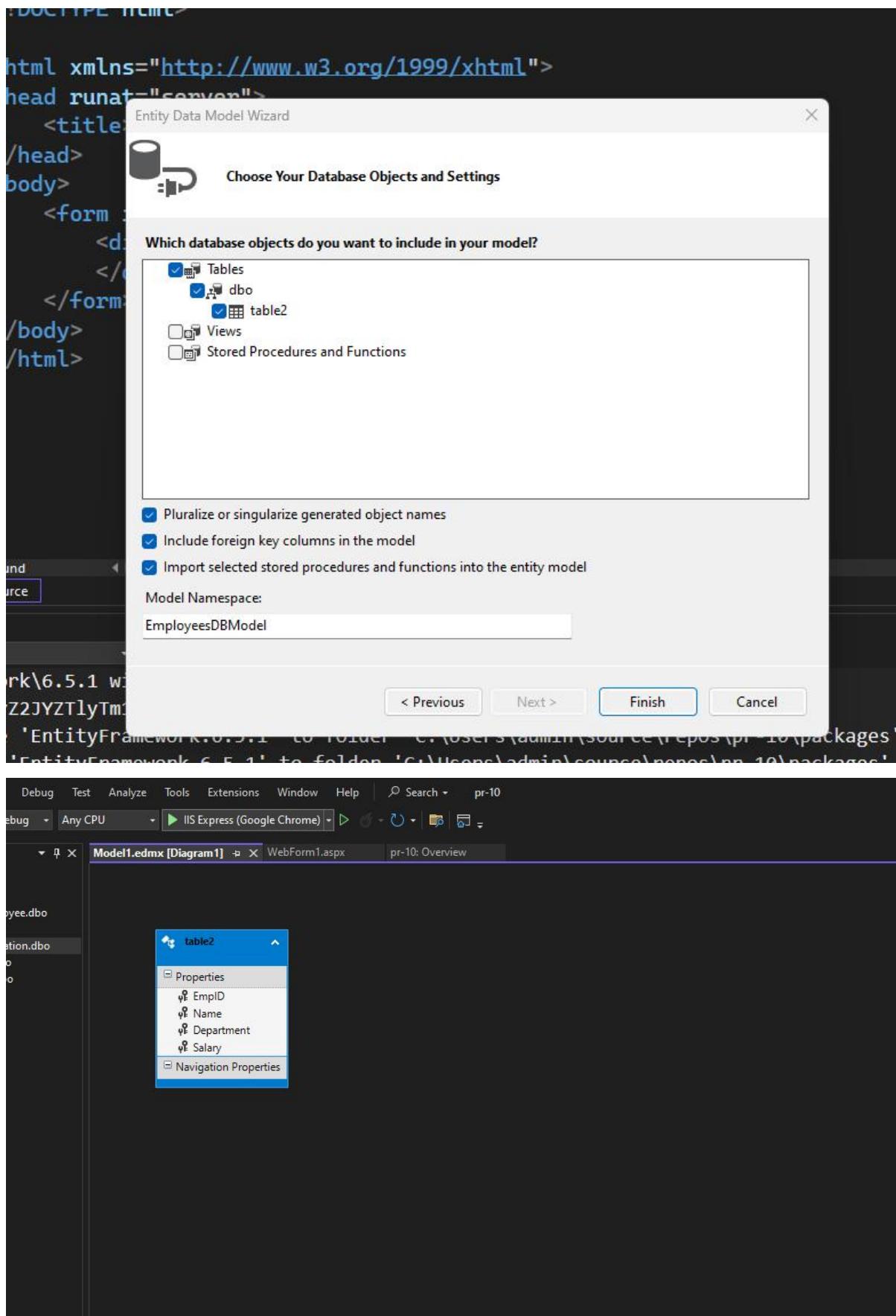
```
PM> Install-Package EntityFramework -Version 6.5.1
```











```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="practical_number_10.WebForm1" %>
```

```
<!DOCTYPE html>
<html lang="en">
<head runat="server">
    <title>Entity Framework CRUD Demo</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Employee Management (Entity Framework)</h2>

            <!-- Add Employee Form -->
            <asp:Label runat="server" Text="Name:></asp:Label>
            <asp:TextBox ID="txtName" runat="server" style="margin-left: 46px"></asp:TextBox>
            <br />

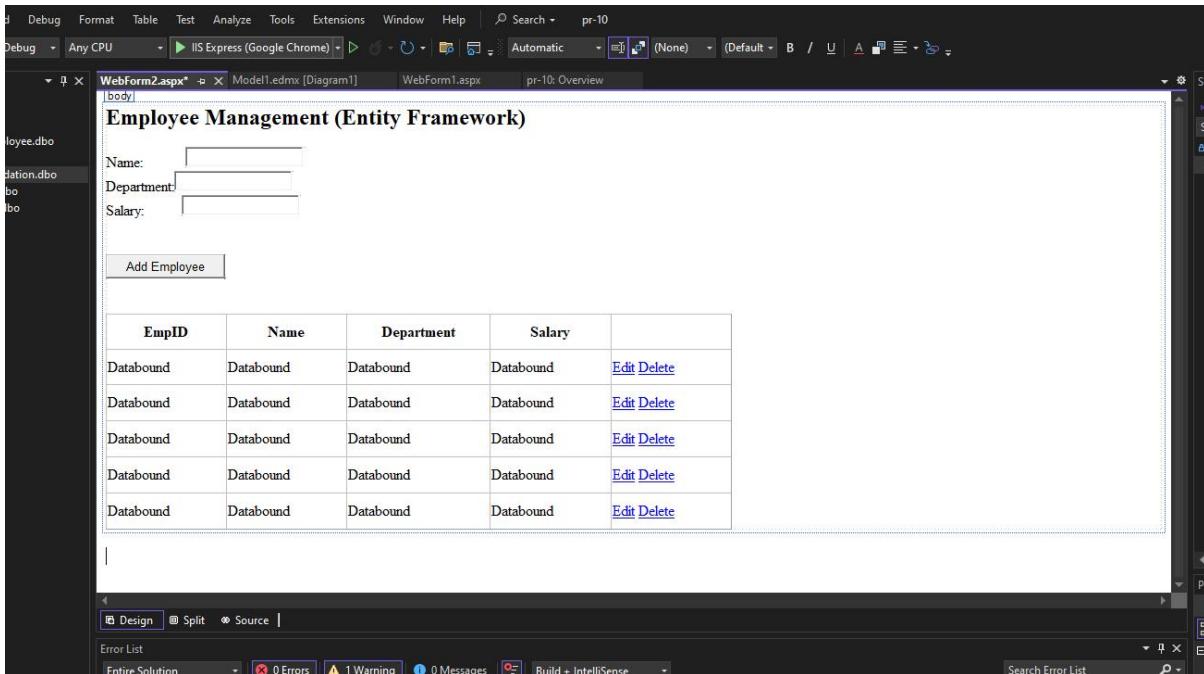
            <asp:Label runat="server" Text="Department:></asp:Label>
            <asp:TextBox ID="txtDepartment" runat="server"></asp:TextBox> <br />

            <asp:Label runat="server" Text="Salary:></asp:Label>
            <asp:TextBox ID="txtSalary" runat="server" style="margin-left: 41px"></asp:TextBox>
            <br />
            <br />
            <br />

            <asp:Button ID="btnAdd" runat="server" Text="Add Employee" OnClick="btnAdd_Click" />
            <br /><br />

            <!-- Display Employees -->
            <asp:GridView ID="gvEmployees" runat="server" AutoGenerateColumns="False" DataKeyNames="EmpID"
                OnRowEditing="gvEmployees_RowEditing" OnRowUpdating="gvEmployees_RowUpdating"
                OnRowCancelingEdit="gvEmployees_RowCancelingEdit" OnRowDeleting="gvEmployees_RowDeleting" Height="233px"
                Width="677px">
                <Columns>
                    <asp:BoundField DataField="EmpID" HeaderText="EmpID" ReadOnly="True" />
                    <asp:BoundField DataField="Name" HeaderText="Name" />
                    <asp:BoundField DataField="Department" HeaderText="Department" />
                    <asp:BoundField DataField="Salary" HeaderText="Salary" />

                    <asp:CommandField ShowEditButton="True" ShowDeleteButton="True" />
                </Columns>
            </asp:GridView>
        </div>
    </form>
</body>
</html>
```



## PRACTICAL-12

### Design Web Applications using Server Side Session Management Techniques

#### WebForm1.aspx

```

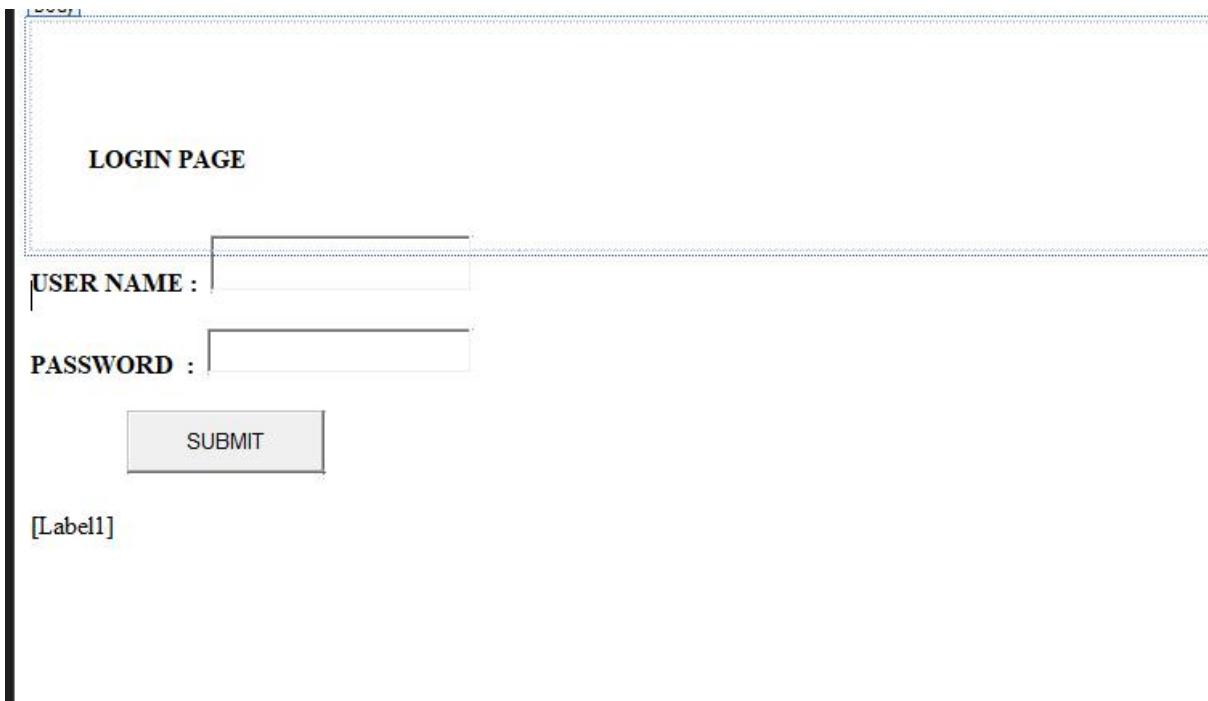
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="practical_12_53_.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>LOGIN PAGE </title>
    <style type="text/css">
        .auto-style1 {
            height: 142px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server" class="auto-style1">
        <div>
            <br />
            <br />
            <br />
            <br />
        </div>
    </form>
</body>
</html>

```

```
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<strong>LOGIN PAGE
<br />
<br />
</strong>
<br />
<strong>USER NAME :&nbsp;
<asp:TextBox ID="TextBox1" runat="server" Height="29px" Width="155px"></asp:TextBox>
<br />
<br />
PASSWORD:&nbsp; :&nbsp;</strong>
<asp:TextBox ID="TextBox2" runat="server" Height="22px" Width="157px"></asp:TextBox>
<br />
<p>
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<asp:Button
ID="Button1" runat="server" Height="39px" OnClick="Button1_Click" Text="SUBMIT" Width="123px" />
</p>
<p>
    <asp:Label ID="Label1" runat="server"></asp:Label>
</p>
</form>
</body>
</html>
```



## Backend

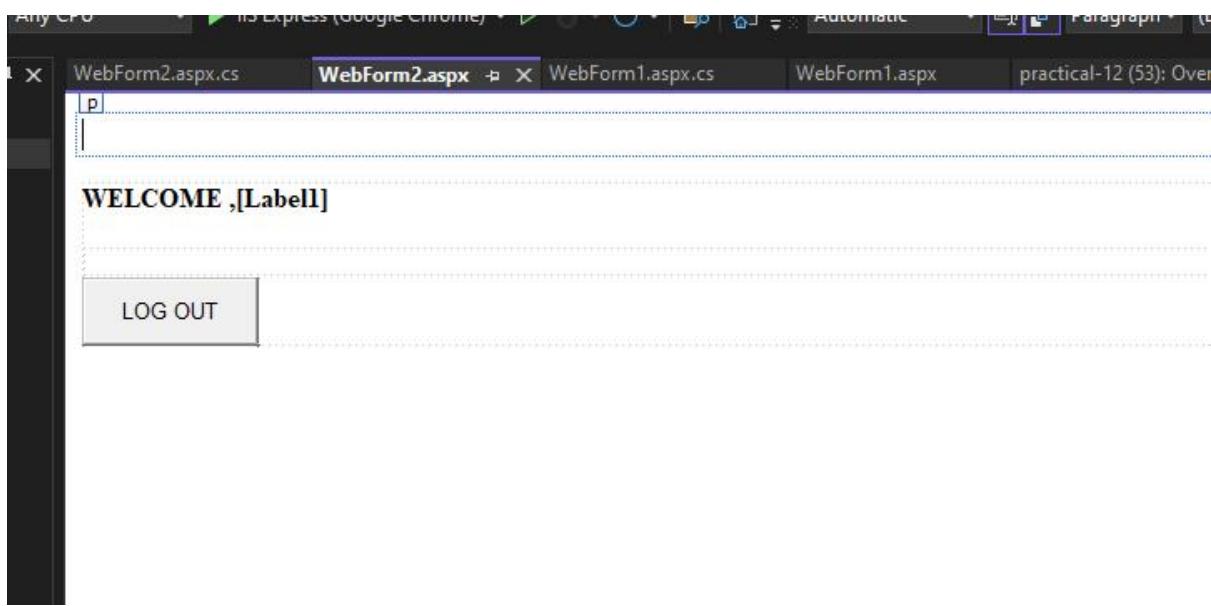
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace practical_12_53_
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
```

```
{  
    if(Session["USERNAME"] != null)  
    {  
        Response.Redirect("WebForm2.aspx");  
    }  
}  
  
protected void Button1_Click(object sender, EventArgs e)  
{  
    string username = TextBox1.Text;  
    string password = TextBox2.Text;  
    if(username == "admin" && password == "1234")  
    {  
        Session["USERNAME"] = username;  
        Response.Redirect("WebForm2.aspx");  
    }  
    else  
    {  
        Label1.Text = "Dear user you have entered the Invalid username or password !";  
    }  
}
```

## WebForm2.aspx

```
Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs" Inherits="practical_12__53_.WebForm2" %>  
  
<!DOCTYPE html>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
  <head runat="server">  
    <title>Dashboard</title>  
  </head>  
  <body>  
    <p>  
      <br />  
    </p>  
    <form id="form1" runat="server">  
      <p>  
        <strong>WELCOME ,<asp:Label ID="Label1" runat="server"></asp:Label>  
      </strong>  
      </p>  
      <div>  
        </div>  
        <asp:Button ID="Button1" runat="server" Height="41px" OnClick="Button1_Click" Text="LOG OUT" Width="106px" />  
    </form>  
  </body>  
</html>
```

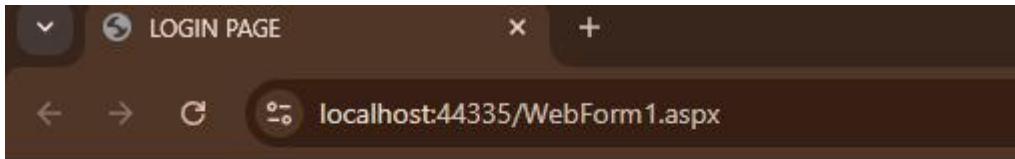


## Backend

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace practical_12_53_
{
    public partial class WebForm2 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Session["USERNAME"] == null)
            {
                Response.Redirect("WebForm1.aspx");
            }
            else
            {
                Label1.Text = Session["USERNAME"].ToString();
            }
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Session.Abandon();
            Response.Redirect("WebForm1.aspx");
        }
    }
}
```

**LOGIN PAGE****USER NAME :** **PASSWORD :** **SUBMIT****LOGIN PAGE****USER NAME :** **PASSWORD :** **SUBMIT**

**WELCOME ,admin**

**LOG OUT**

---

## **Login Page**

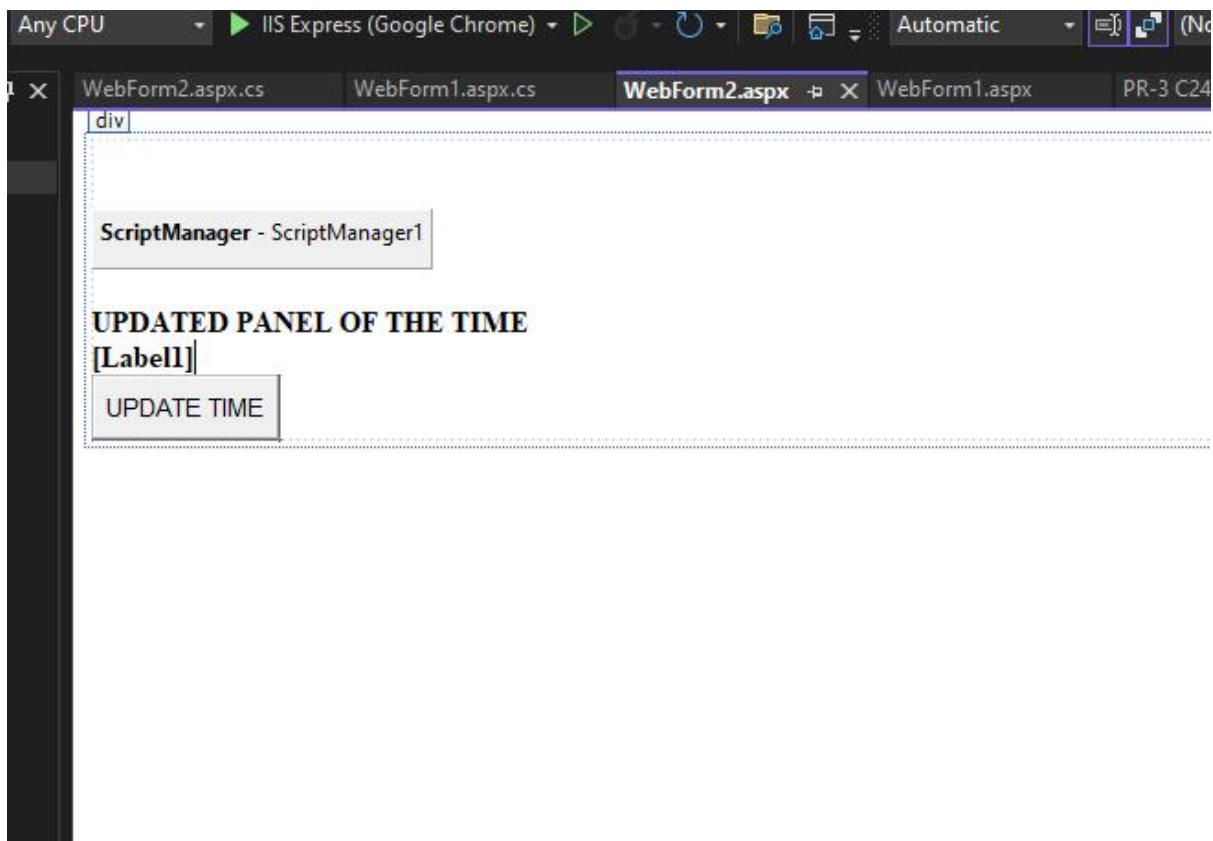
Username:

Password:

**Login**

## PRACTICAL-13

Build a web page using AJAX Controls.



```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs" Inherits="PR_3_C24053.WebForm2" %>

<!DOCTYPE html>

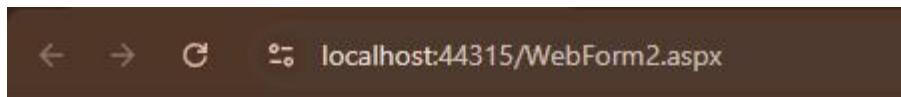
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <br />
            <asp:ScriptManager ID="ScriptManager1" runat="server">
            </asp:ScriptManager>
            <br />
            <strong>UPDATED PANEL OF THE TIME</strong>
            <br />
            <asp:Label ID="Label1" runat="server"></asp:Label>
            <br />
            <asp:Button ID="Button1" runat="server" Height="36px" OnClick="Button1_Click" Text="UPDATE TIME" Width="103px" />
        </div>
    </form>
</body>
</html>
```

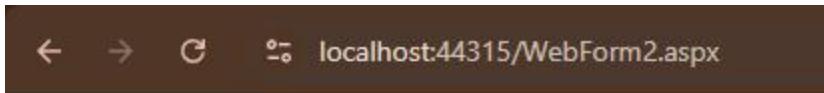
**BACKEND:**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace PR_3_C24053
{
    public partial class WebForm2 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                Label1.Text = "Latest Update time:" + DateTime.Now.ToString("HH:mm:ss");
            }
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Label1.Text = "Latest Update time:" + DateTime.Now.ToString("HH:mm:ss");
        }
    }
}
```

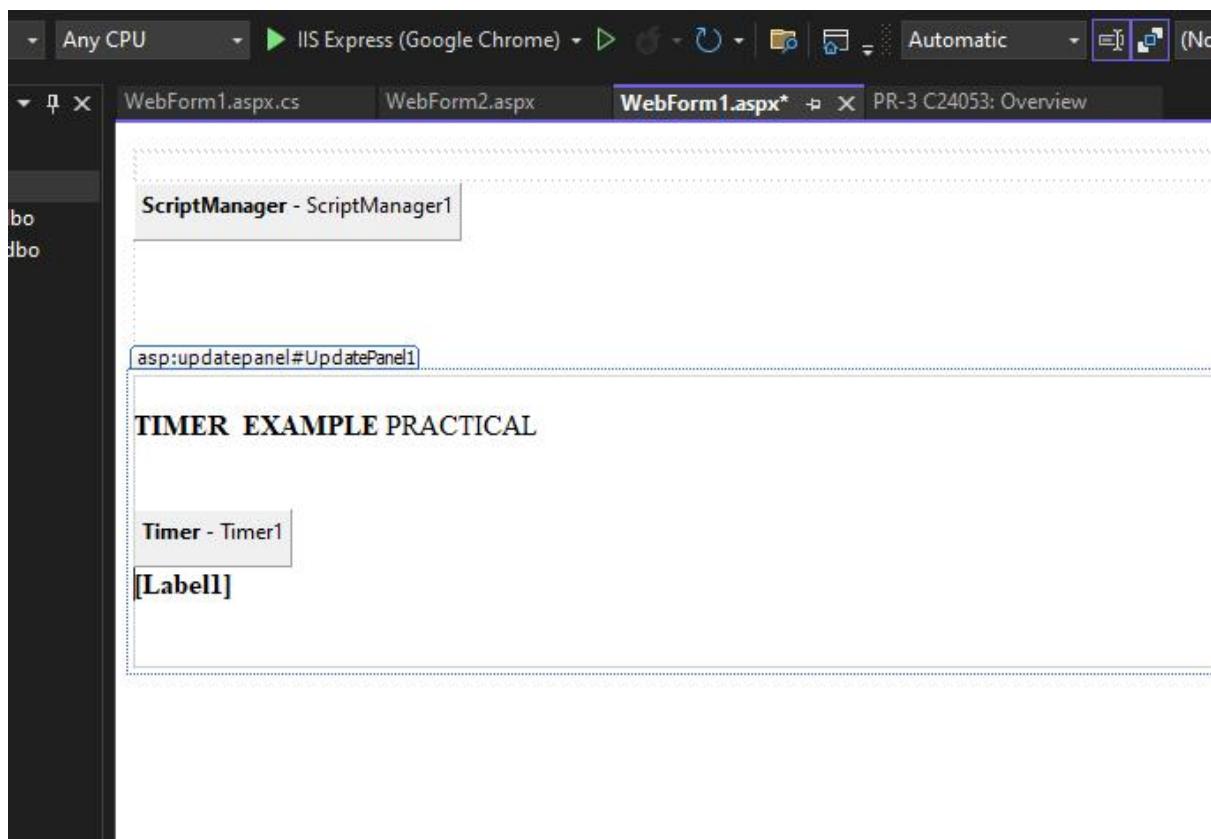
**UPDATED PANEL OF THE TIME****Latest Update time:16:19:04****UPDATE TIME**



## UPDATED PANEL OF THE TIME

Latest Update time:16:19:21

UPDATE TIME



```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="PR_3_C24053.WebForm1" %>
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
        </div>
        <asp:ScriptManager ID="ScriptManager1" runat="server">
        </asp:ScriptManager>
        <br />
```

```
<br />
<br />
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
    <ContentTemplate>
        <br />
        <strong>TIMER&nbsp; EXAMPLE </strong>PRACTICAL<strong><br />
        <br />
        <asp:Timer ID="Timer1" runat="server" Interval="5000" OnTick="Timer1_Tick">
            </asp:Timer>
        <asp:Label ID="Label1" runat="server"></asp:Label>
        &nbsp; </strong>
        <br />
        <br />
    </ContentTemplate>
</asp:UpdatePanel>
</form>
</body>
</html>
```

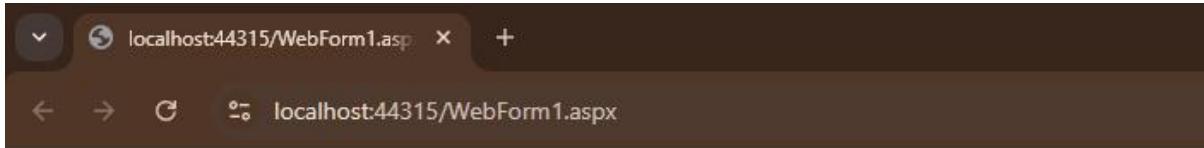
## BACKEND:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace PR_3_C24053
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)

        {
            if (!IsPostBack)
            {
                Label1.Text = "Latest Update time:" + DateTime.Now.ToString("HH:mm:ss");
            }
        }

        protected void Timer1_Tick(object sender, EventArgs e)
        {
            Label1.Text = "Latest Update time:" + DateTime.Now.ToString("HH:mm:ss");
        }
    }
}
```



## TIMER EXAMPLE PRACTICAL

Latest Update time:16:15:02

## PRACTICAL-14

Build a web application to create and use web service in ASP.net

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

namespace web_services
{
    /// <summary>
    /// Summary description for WebService1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.
    // [System.Web.Script.Services.ScriptService]
    public class WebService1 : System.Web.Services.WebService
    {

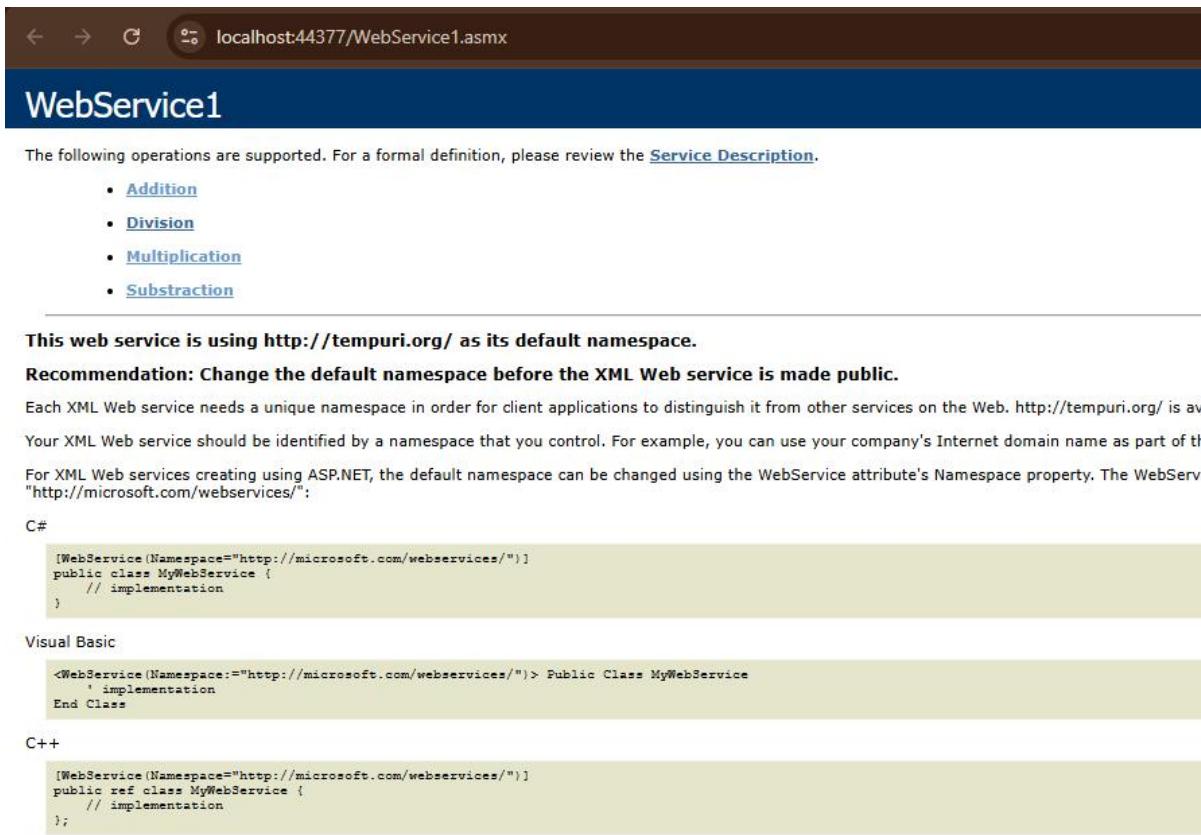
        [WebMethod]
        public double Addition(double a, double b)
        {
            return (a + b);
        }
    }
}
```

```
[WebMethod]
public double Subtraction(double a, double b)
{
    return (a - b);
}

[WebMethod]
public double Multiplication(double a, double b)
{
    return (a * b);
}

[WebMethod]
public double Division(double a, double b)
{
    return (a / b);
}

}
```



The following operations are supported. For a formal definition, please review the [Service Description](#).

- [Addition](#)
- [Division](#)
- [Multiplication](#)
- [Subtraction](#)

---

**This web service is using <http://tempuri.org/> as its default namespace.**

**Recommendation: Change the default namespace before the XML Web service is made public.**

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. <http://tempuri.org/> is available for use by anyone, so it is important to choose a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace.

For XML Web services created using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is defined in the <http://microsoft.com/webservices/>:

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```

Visual Basic

```
<WebService(Namespace:="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementation
End Class
```

C++

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public ref class MyWebService {
    // implementation
};
```

For more details on XML namespaces, see the W3C recommendation on [Namespaces in XML](#).

## WebService1

Click [here](#) for a complete list of operations.

### Addition

#### Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

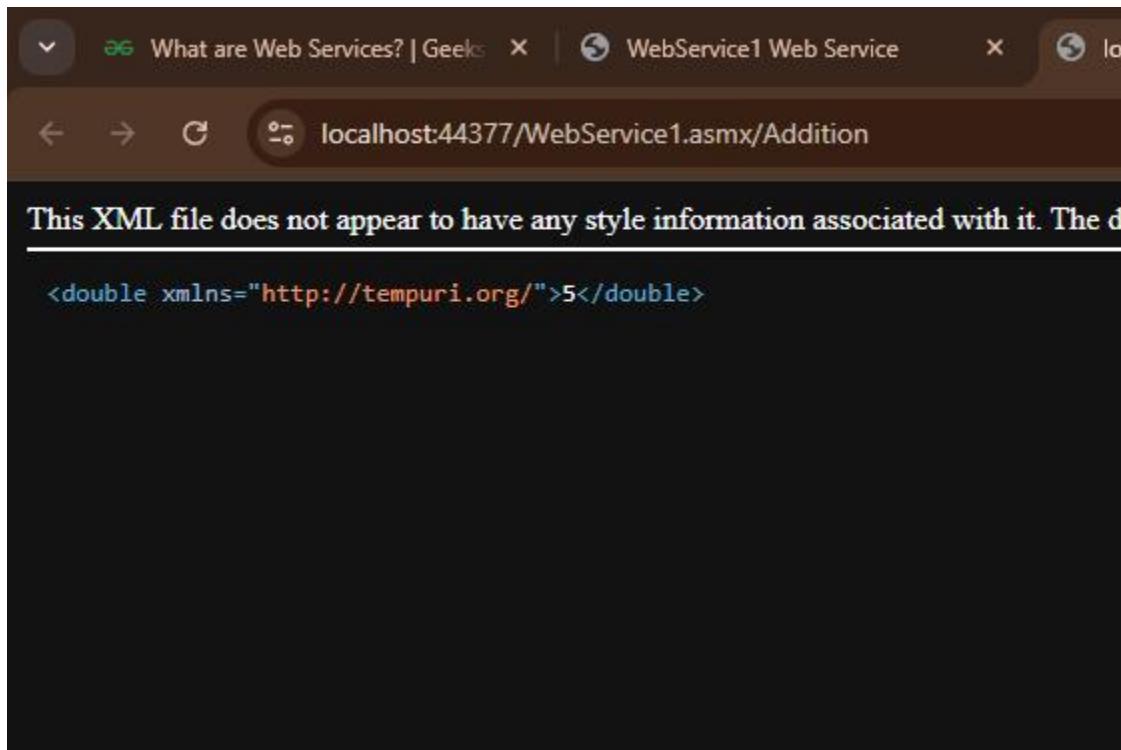
| Parameter | Value |
|-----------|-------|
| a:        | 2     |
| b:        | 3     |

#### SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /WebService1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/Addition"

<?xml version="1.0" encoding="utf-8"?>
```



localhost:44377/WebService1.asmx?op=Division

## WebService1

---

Click [here](#) for a complete list of operations.

### Division

**Test**

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

| Parameter     | Value                |
|---------------|----------------------|
| a:            | <input type="text"/> |
| b:            | <input type="text"/> |
| <b>Invoke</b> |                      |

### SOAP 1.1

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="web_services.WebForm1" %>
```

```
<!DOCTYPE html>
```

```

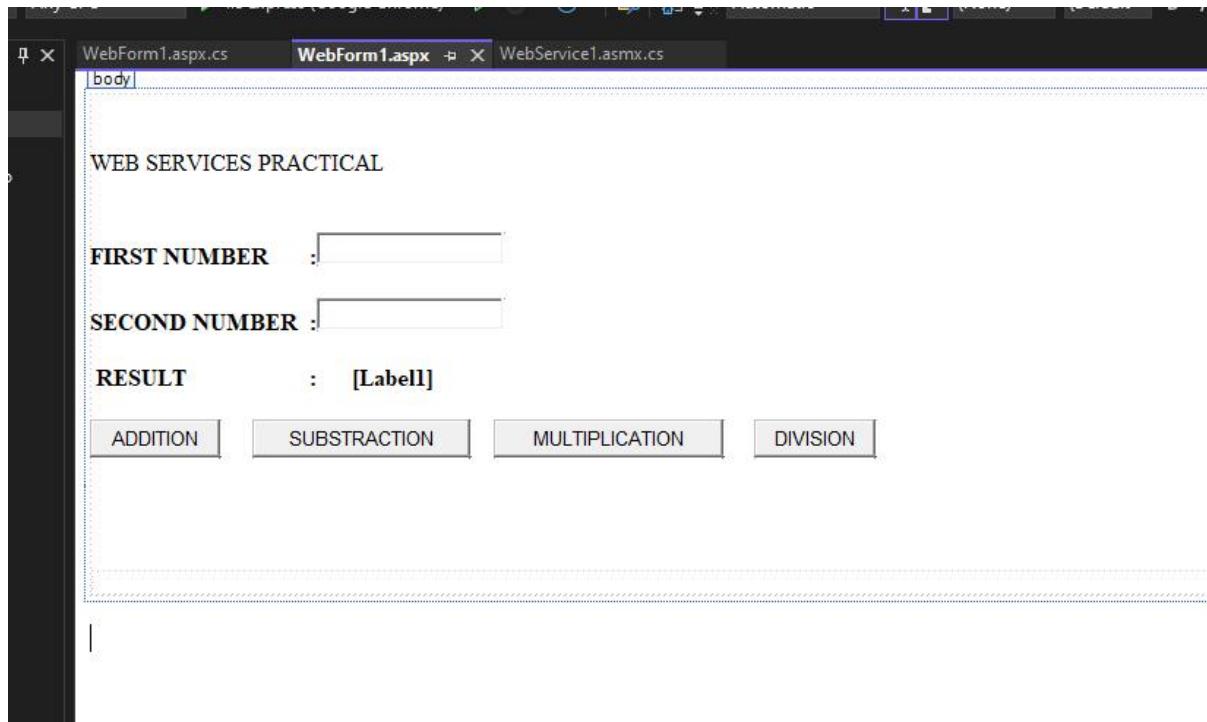
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<style type="text/css">
.auto-style1 {
  margin-left: 22px;
}
.auto-style2 {
  margin-left: 16px;
}
.auto-style3 {
  margin-left: 21px;
}
</style>
</head>
<body>
<form id="form1" runat="server">
<p>
  &ampnbsp</p>
<p>
  WEB SERVICES PRACTICAL
</p>
<p>
  <br />
  <strong>FIRST NUMBER</strong> :<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
</p>
<p>
  <strong>SECOND NUMBER</strong> :<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
</p>
<p>
  <strong>RESULT</strong>
</p>

```

```

<asp:Label ID="Label1" runat="server"></asp:Label>
</strong>
</p>
<asp:Button ID="Button4" runat="server" OnClick="Button4_Click" Text="ADDITION" />
<asp:Button ID="Button3" runat="server" CssClass="auto-style1" OnClick="Button3_Click" Text="SUBTRACTION" />
<asp:Button ID="Button2" runat="server" CssClass="auto-style2" OnClick="Button2_Click" Text="MULTIPLICATION" />
<asp:Button ID="Button1" runat="server" CssClass="auto-style3" OnClick="Button1_Click" Text="DIVISION" />
<br />
<p>
  &nbsp;</p>
<div>
</div>
</form>
</body>
</html>

```



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace web_services
{
  public partial class WebForm1 : System.Web.UI.Page
  {
    WebService1 wb = new WebService1();
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button4_Click(object sender, EventArgs e)
    {

```

```
double no1 = Convert.ToDouble(textBox1.Text);
double no2 = Convert.ToDouble(textBox2.Text);

double result = wb.Addition(no1, no2);
Label1.Text = "Addition" + result.ToString();

}

protected void Button3_Click(object sender, EventArgs e)
{

double no1 = Convert.ToDouble(textBox1.Text);
double no2 = Convert.ToDouble(textBox2.Text);

double result = wb.Subtraction(no1, no2);
Label1.Text = "Substraction" + result.ToString();

}

protected void Button2_Click(object sender, EventArgs e)
{

double no1 = Convert.ToDouble(textBox1.Text);
double no2 = Convert.ToDouble(textBox2.Text);

double result = wb.Multiplication(no1, no2);
Label1.Text = "Multiplication" + result.ToString();

}

protected void Button1_Click(object sender, EventArgs e)
{
    double no1 = Convert.ToDouble(textBox1.Text);
    double no2 = Convert.ToDouble(textBox2.Text);

    double result = wb.Division(no1, no2);
    Label1.Text = "Division" + result.ToString();
}
}
```



## WEB SERVICES PRACTICAL

**FIRST NUMBER** :

**SECOND NUMBER** :

**RESULT** :



## WEB SERVICES PRACTICAL

**FIRST NUMBER** :

**SECOND NUMBER** :

**RESULT** : **Addition8**



## WEB SERVICES PRACTICAL

**FIRST NUMBER :**

**SECOND NUMBER :**

**RESULT :** **Substraction-4**

**ADDITION**

**SUBTRACTION**

**MULTIPLICATION**

**DIVISION**



## WEB SERVICES PRACTICAL

**FIRST NUMBER :**

**SECOND NUMBER :**

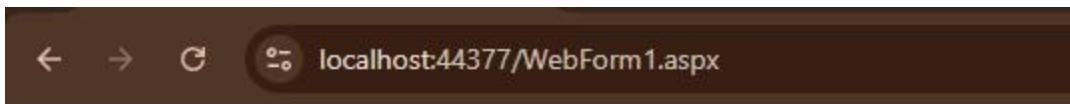
**RESULT :** **Multiplication12**

**ADDITION**

**SUBTRACTION**

**MULTIPLICATION**

**DIVISION**



## WEB SERVICES PRACTICAL

**FIRST NUMBER :**

**SECOND NUMBER :**

**RESULT :** Division **0.3333333333333333**

**ADDITION**

**SUBTRACTION**

**MULTIPLICATION**

**DIVISION**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

namespace simple_interest
{
    /// <summary>
    /// Summary description for WebService1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.
    // [System.Web.Script.Services.ScriptService]
    public class WebService1 : System.Web.Services.WebService
    {

        [WebMethod]

        public double SI(double P, double N, double R)
        {
            return (P*N*R)/100;
        }
    }
}
```

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [SI](#)

**This web service is using <http://tempuri.org/> as its default namespace.**

**Recommendation: Change the default namespace before the XML Web service is made**

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's name as the root of the namespace.

For XML Web services created using ASP.NET, the default namespace can be changed using the WebService attribute in the class declaration, for example:

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```

Click [here](#) for a complete list of operations.

## SI

### Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

| Parameter | Value |
|-----------|-------|
| P:        | 2000  |
| N:        | 1     |
| R:        | 2     |

[Invoke](#)

### SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /WebService1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/SI"

<?xml version="1.0" encoding="utf-8"?>
```



```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="simple_interest.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <p>
        <br />
        <strong>SIMPLE INTEREST :-</strong></p>
    <form id="form1" runat="server">
        <p>
            ENTER THE PRINCIPLE AMOUNT :
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
        </p>
        <p>
            ENTER THE NUMBER OF YEAR &nbsp;&nbsp;&nbsp;&nbsp; :<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
        </p>
        <p>
            ENTER THE RATE OF INTEREST&nbsp;&nbsp;&nbsp; :<asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
        </p>
        <p>
            &nbsp;</p>
        <p>
            <strong>INTEREST AMOUNT :</strong><asp:Label ID="Label1" runat="server"></asp:Label>
        </p>
        <p>
            <strong>&nbsp;</strong><asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="CALCULATE" />
        </p>
        <p>
            &nbsp;</p>
    <div>
    </div>
</form>
</body>
</html>
```

WebForm1.aspx.cs    WebForm1.aspx    WebService1.asmx.cs

body

**SIMPLE INTEREST :-**

ENTER THE PRINCIPLE AMOUNT :

ENTER THE NUMBER OF YEAR :

ENTER THE RATE OF INTEREST :

**INTEREST AMOUNT :[Label1]**

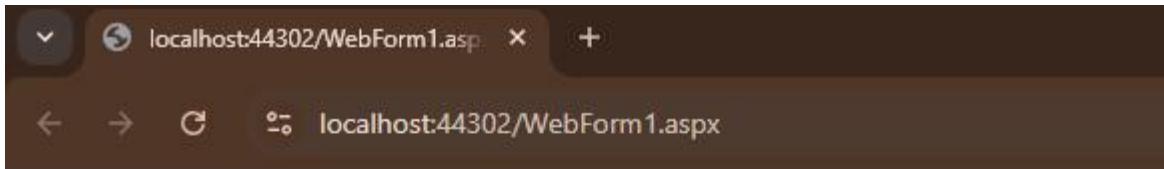
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace simple_interest
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        WebService1 service1 = new WebService1 ();
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            double P = Convert.ToDouble(TextBox1.Text);
            double R = Convert.ToDouble(TextBox2.Text);
            double N = Convert.ToDouble(TextBox3.Text);

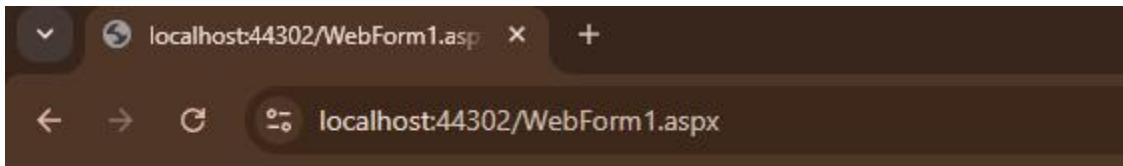
            double result = service1.SI(P, N, R);
            Label1.Text = result.ToString();
        }
    }
}
```

**SIMPLE INTEREST :-**

ENTER THE PRINCIPLE AMOUNT :

ENTER THE NUMBER OF YEAR :

ENTER THE RATE OF INTEREST :

**INTEREST AMOUNT :****SIMPLE INTEREST :-**

ENTER THE PRINCIPLE AMOUNT :

ENTER THE NUMBER OF YEAR :

ENTER THE RATE OF INTEREST :

**INTEREST AMOUNT :40**

## PRACTICAL – 15

### Build a web application to create and WCF service in ASP.net

#### 1) IService.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name "IService" in both code and config file together.
[ServiceContract]
public interface IService
{
    [OperationContract]
    string GetData(int value);
    [OperationContract]
    double add(double a, double b);

    [OperationContract]
    double sub(double a, double b);

    [OperationContract]
    double mul(double a, double b);

    [OperationContract]
    double div(double a, double b);

    [OperationContract]
    CompositeType GetDataUsingDataContract(CompositeType composite);

    // TODO: Add your service operations here
}

// Use a data contract as illustrated in the sample below to add composite types to service operations.
[DataContract]
public class CompositeType
{
    bool boolValue = true;
    string stringValue = "Hello ";

    [DataMember]
    public bool BoolValue
    {
        get { return boolValue; }
        set { boolValue = value; }
    }
}
```

```

        }

[DataMember]
public string StringValue
{
    get { return stringValue; }
    set { stringValue = value; }
}

}

```

## 2) Service.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "Service" in code, svc and config file
together.
public class Service : IService
{
    public string GetData(int value)
    {
        return string.Format("You entered: {0}", value);
    }
    public double add(double a, double b)
    {
        return a + b;
    }

    public double sub(double a, double b)
    {
        return a - b;
    }

    public double mul(double a, double b)
    {
        return a * b;
    }

    public double div(double a, double b)
    {
        return a / b;
    }
    public CompositeType GetDataUsingDataContract(CompositeType composite)
    {
        if(composite == null)
        {
            throw new ArgumentNullException("composite");
        }
        if(composite.BoolValue)
        {
            composite.StringValue += "Suffix";
        }
        return composite;
    }
}

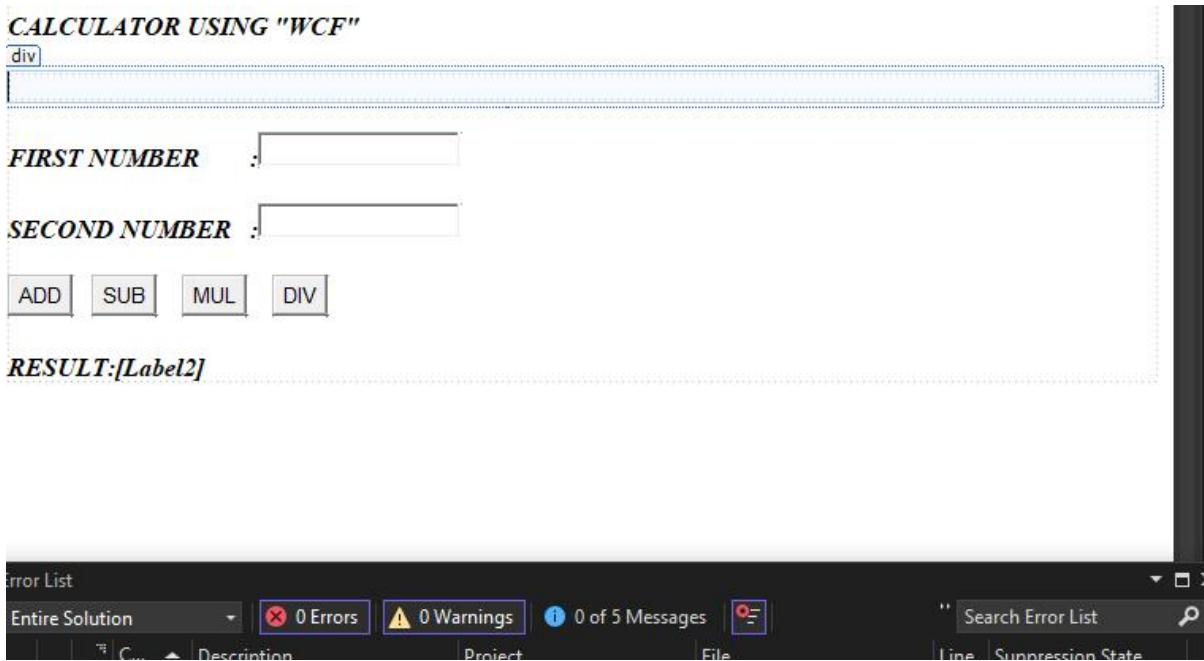
```

## 3) Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
        #form1 {
            font-weight: 700;
            font-style: italic;
        }
        .auto-style1 {
            font-style: italic;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <br />
        CALCULATOR USING "WCF"<br />
        <div>
        </div>
        <p style="font-weight: 700">
            FIRST NUMBER:<br />
            :<asp:TextBox ID="TextBox1" runat="server"
            CssClass="auto-style1"></asp:TextBox>
        </p>
        <p style="font-weight: 700">
            SECOND NUMBER:<br />
            :<asp:TextBox ID="TextBox2" runat="server"
            OnTextChanged="TextBox2_TextChanged"></asp:TextBox>
        </p>
        <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="ADD" />
        &nbsp;&nbsp;
        <asp:Button ID="Button2" runat="server" OnClick="Button2_Click" Text="SUB" />
        &nbsp;&nbsp;
        <asp:Button ID="Button3" runat="server" OnClick="Button3_Click" Text="MUL" />
        &nbsp;&nbsp;
        <asp:Button ID="Button4" runat="server" OnClick="Button4_Click" Text="DIV" />
        <br />
        <br />
        RESULT:<asp:Label ID="Label2" runat="server"></asp:Label>
    </form>
</body>
</html>
```



#### 4) Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Configuration;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    ServiceReference1.ServiceClient service = new ServiceReference1.ServiceClient();
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void Button3_Click(object sender, EventArgs e)
    {
        double a = Convert.ToDouble(TextBox1.Text);
        double b = Convert.ToDouble(TextBox2.Text);

        double result = service.mul(a, b);
        Label2.Text = "Multiplication :" + result.ToString();
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        double a = Convert.ToDouble(TextBox1.Text);
        double b = Convert.ToDouble(TextBox2.Text);

        double result = service.add(a, b);
        Label2.Text = "Addition :" + result.ToString();
    }

    protected void TextBox2_TextChanged(object sender, EventArgs e)
    {
    }
}
```

```
}

protected void Button2_Click(object sender, EventArgs e)
{
    double a = Convert.ToDouble(TextBox1.Text);
    double b = Convert.ToDouble(TextBox2.Text);

    double result = service.sub(a, b);
    Label2.Text = "Substration :" + result.ToString();
}

protected void Button4_Click(object sender, EventArgs e)
{
    double a = Convert.ToDouble(TextBox1.Text);
    double b = Convert.ToDouble(TextBox2.Text);

    double result = service.div(a, b);
    Label2.Text = "Devision :" + result.ToString();
}
```

The screenshot shows a web browser window with the URL `localhost:56980/Default.aspx`. The page title is "CALCULATOR USING "WCF"". The interface includes input fields for "FIRST NUMBER" and "SECOND NUMBER", and buttons for "ADD", "SUB", "MUL", and "DIV". A "RESULT:" label is present below the input fields.

**CALCULATOR USING "WCF"**

**FIRST NUMBER :**

**SECOND NUMBER :**

**ADD** **SUB** **MUL** **DIV**

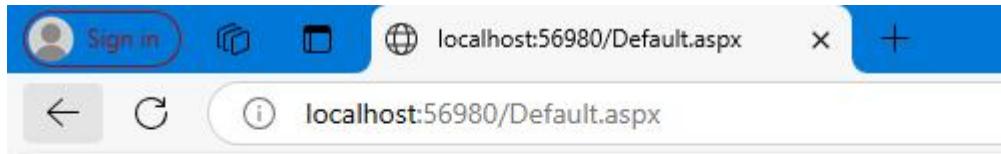
**RESULT:**

A screenshot of a web browser window titled "localhost:56980/Default.aspx". The page displays a calculator interface with the title "CALCULATOR USING "WCF"". It includes input fields for "FIRST NUMBER" (5) and "SECOND NUMBER" (2), and buttons for "ADD", "SUB", "MUL", and "DIV". Below the calculator, the text "RESULT: Addition :7" is shown.

**CALCULATOR USING "WCF"****FIRST NUMBER** :**SECOND NUMBER** :   **RESULT:** *Addition :7*

A screenshot of a web browser window titled "localhost:56980/Default.aspx". The page displays a calculator interface with the title "CALCULATOR USING "WCF"". It includes input fields for "FIRST NUMBER" (5) and "SECOND NUMBER" (2), and buttons for "ADD", "SUB", "MUL", and "DIV". Below the calculator, the text "RESULT: Addition :7" is shown.

**CALCULATOR USING "WCF"****FIRST NUMBER** :**SECOND NUMBER** :   **RESULT:** *Addition :7*

***CALCULATOR USING "WCF"******FIRST NUMBER :*** ***SECOND NUMBER :*** ***[ADD] [SUB] [MUL] [DIV]******RESULT:Multiplication :10***

## PRACTICAL-16

### Design web application using MVC framework

#### Practical 16 : MVC Application using Entity Framework

##### Database Setup

###### Step 1: Create a Database in SQL Server

Open SQL Server Management Studio (SSMS) or Visual Studio SQL Server Object Explorer.

**Create a Database** named StudentDB.

**Create a Table** using the following SQL: Create Database StudentDB

```
use StudentDB
```

```
CREATE TABLE Students (
```

```
    Id INT PRIMARY KEY IDENTITY,  
    Name NVARCHAR(100) NOT  
    NULL, Email NVARCHAR(100)  
    NOT NULL, Age INT NOT NULL
```

```
);
```

SQLQuery1.sql - DE..#P-RLog5JS(np(60))

```
create database StudentsDB;

use StudentDB

CREATE TABLE Students
Id      PRIMARY KEY IDENTITY
Name   NVARCHAR(100) NOT NULL
Email  NVARCHAR(100) NOT NULL
Age    INT
```

100 % <

Results Messages

|   | Id | Name  | Email                  | Age |
|---|----|-------|------------------------|-----|
| 1 | 1  | Adifa | adifa.ansari@gmail.com | 22  |
| 2 | 2  | Sejal | sejal.shah@gmail.com   | 21  |
| 3 | 3  | Sid   | sid.patel@gmail.com    | 18  |

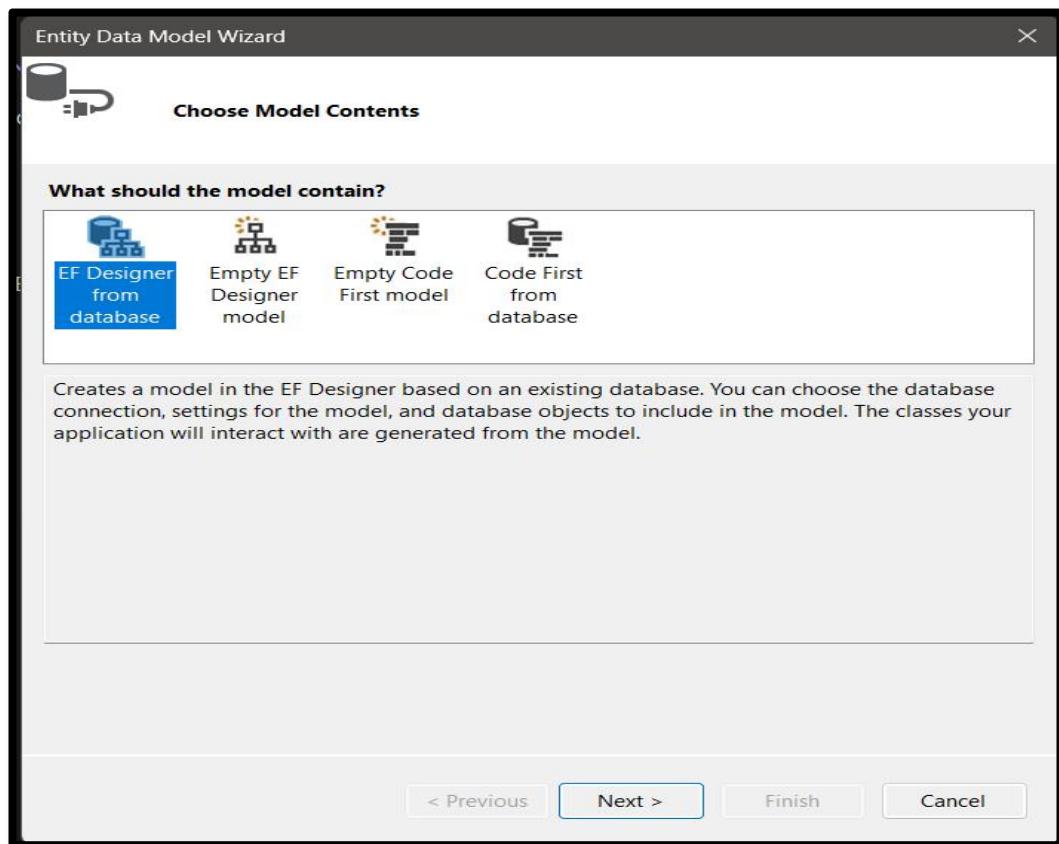
## Step 2: Create a New ASP.NET MVC Project

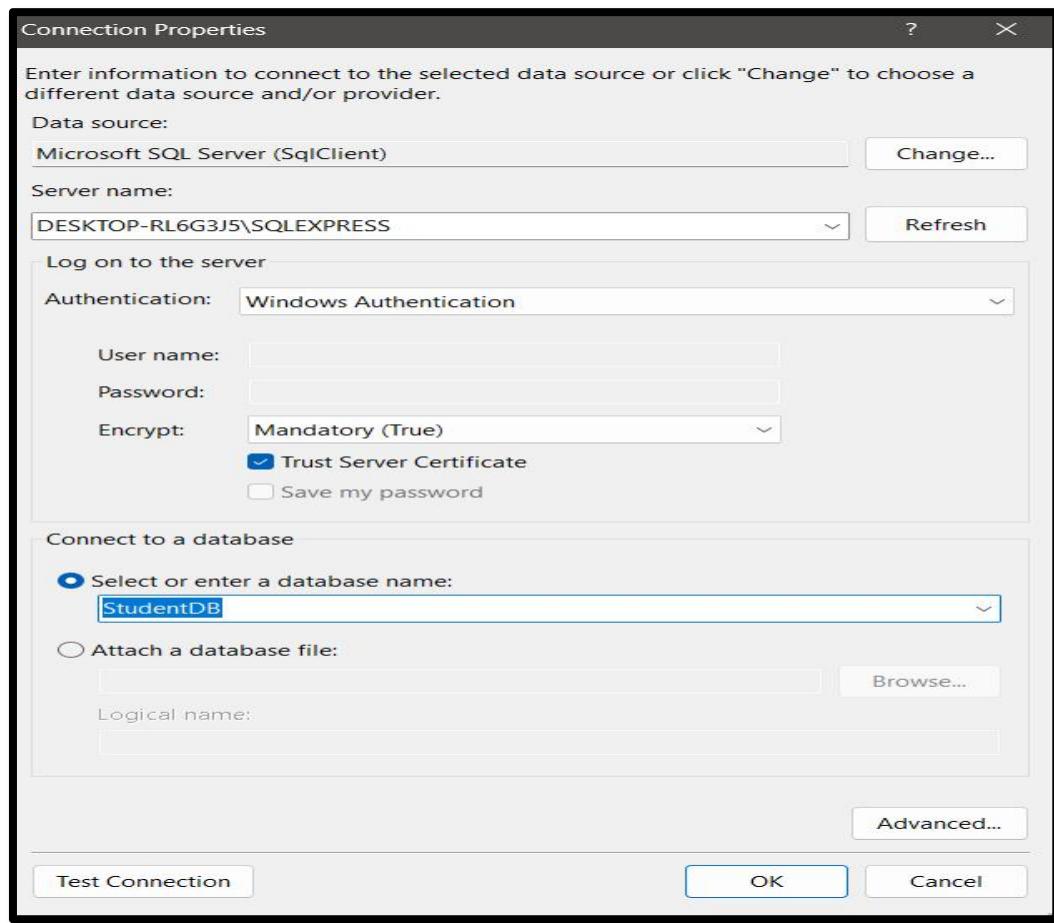
1. Open **Visual Studio**
2. Select **Create a new project**
3. Choose: **ASP.NET Web Application (.NET Framework)**
4. Name: `StudentMVCAApp`
5. Choose **MVC** as the template
6. Click **Create**

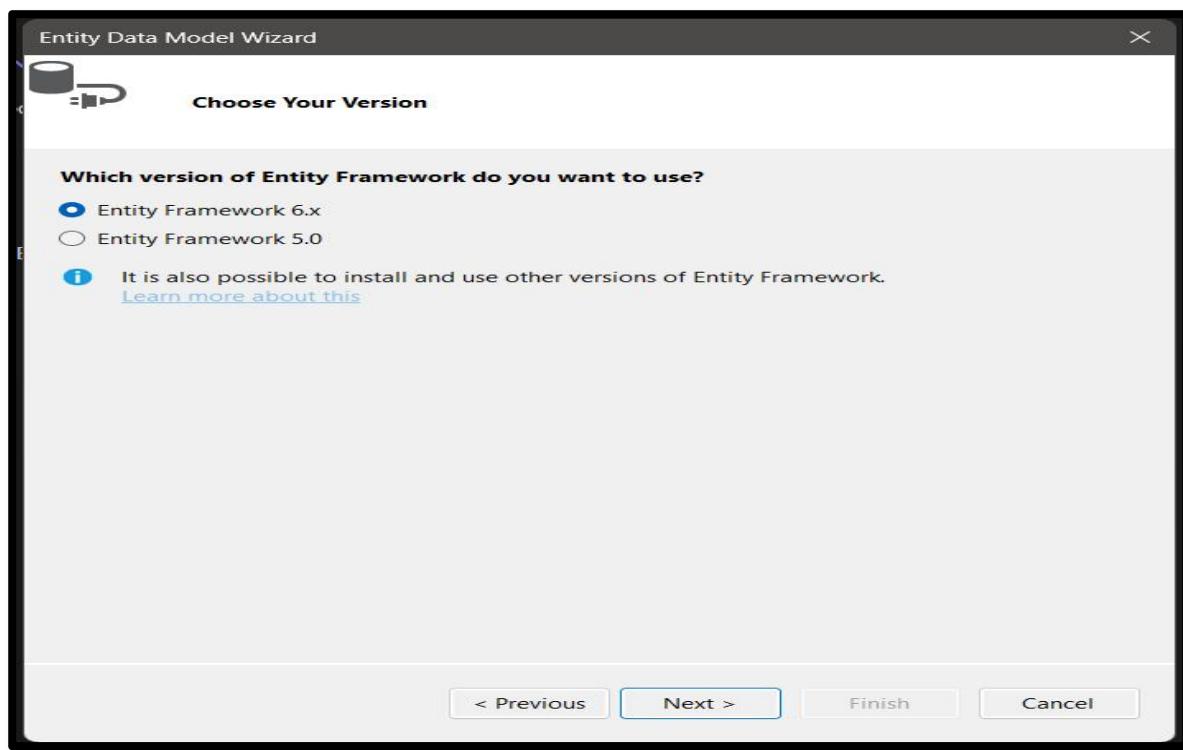
## Entity Framework Model Setup

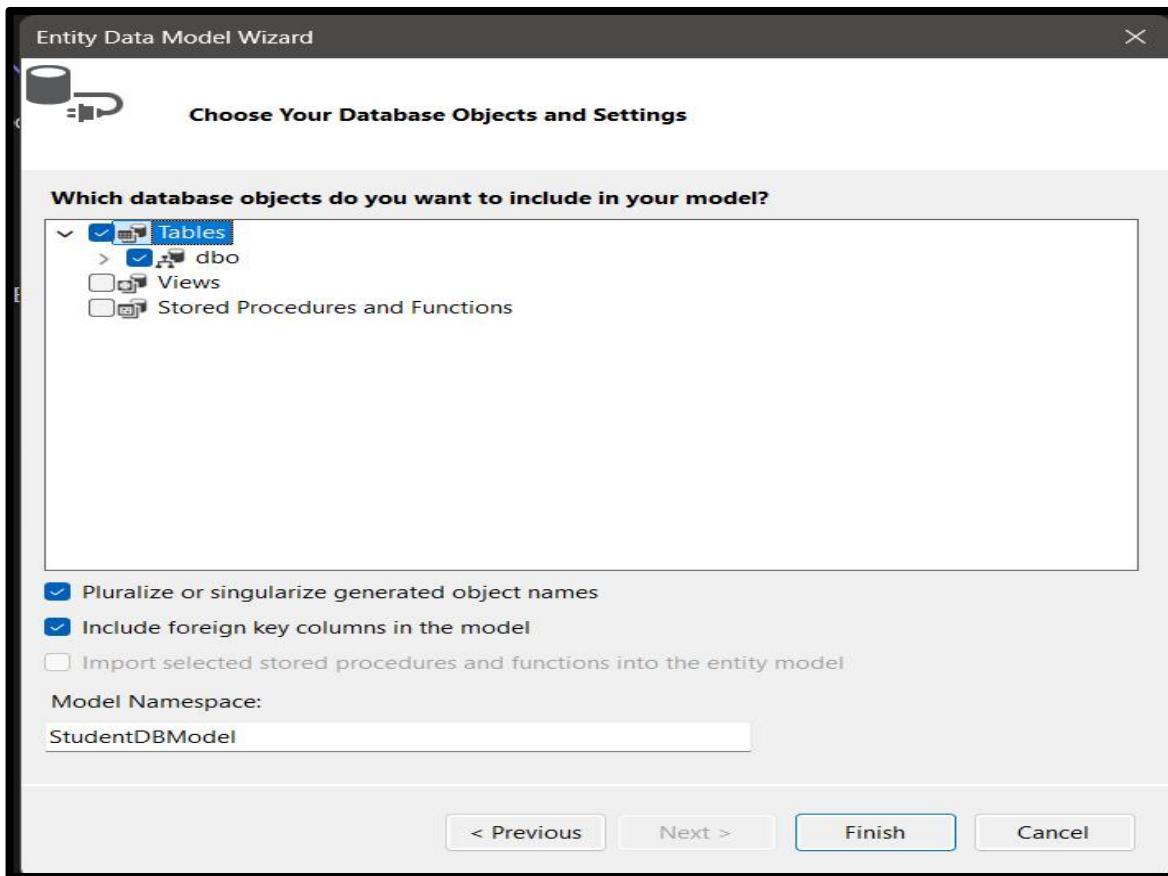
### Step 3: Add Entity Framework Model

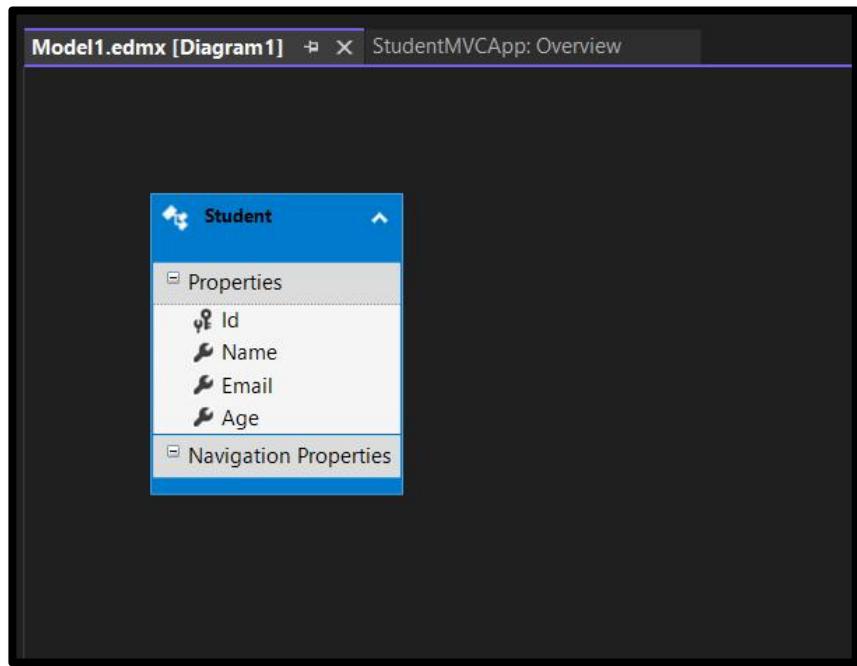
1. Right-click the **Models** folder → Add → New Item
2. Choose **ADO.NET Entity Data Model**
3. Name it: `StudentModel1.edmx`
4. Choose: "EF Designer from database"
5. Select your SQL Server database (`StudentDB`)
6. Select the `Students` table
7. Finish to generate model classes





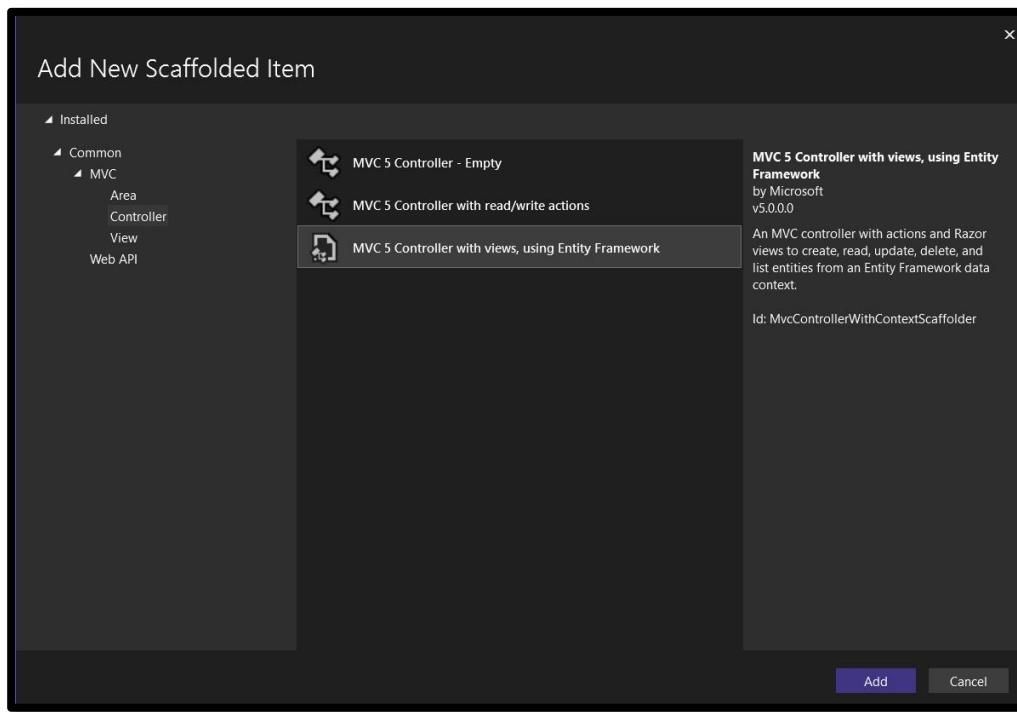


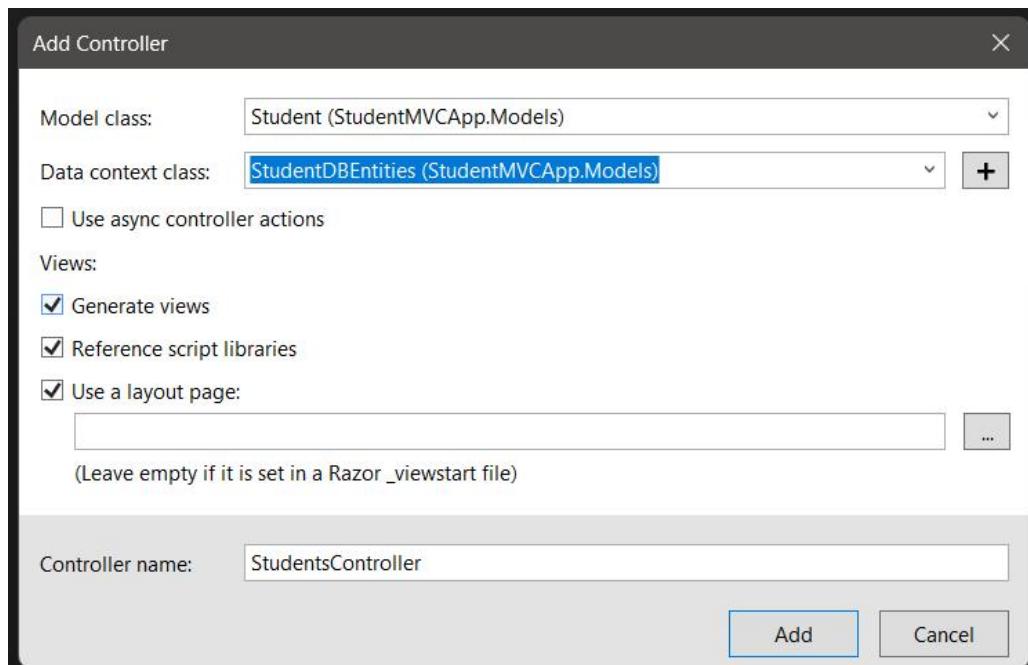




#### Step 4: Create Controller

1. Right-click **Controllers** → Add → Controller
2. Choose: **MVC 5 Controller with views, using Entity Framework**
3. Model class: Student
4. Data context: StudentDBEntities (if using .edmx) or StudentDbContext
5. Click **Add**





## Step 5: Create Views

You can right-click on each controller action and choose **Add View**, or use the auto-generated ones.

Example: Views/Students/Index.cshtml

```
@model IEnumerable<StudentMVCApp.Models.Student>

{@
    ViewBag.Title = "Student List";
}

<h2>Student List</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>

<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Name)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Email)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Age)
        </th>
        <th>Actions</th>
    </tr>
```

```
@foreach (var item in Model) {  
    <tr>  
        <td>@item.Name</td>  
        <td>@item.Email</td>  
        <td>@item.Age</td>  
        <td>  
            @H  
            @Html.ActionLink("Edit", "Edit", new { id = item.Id }) |  
            @Html.ActionLink("Details", "Details", new { id = item.Id }) |  
            tml.ActionLink("Delete", "Delete", new { id = item.Id })  
        </td>  
    </tr>  
}  
  
</table>
```

### Step 6: Set Default Route

In App\_Start/RouteConfig.cs, change default route to:

csharp

CopyEdit

```
defaults: new { controller = "Students", action = "Index", id = UrlParameter.Optional }
```

The screenshot shows a web browser window with the URL `localhost:44300/Students/Index` in the address bar. The page title is "Application name". The main content is titled "Student List" and displays a table of student data. The table has columns for Name, Email, Age, and Actions. The data is as follows:

| Name  | Email                  | Age | Actions   |
|-------|------------------------|-----|---|
| Adifa | adifa.ansari@gmail.com | 22  | <a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a> |
| Sejal | sejal.shah@gmail.com   | 21  | <a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a> |
| Sid   | sid.patel@gmail.com    | 18  | <a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a> |

At the bottom of the page, there is a copyright notice: "© 2025 - My ASP.NET Application".

The screenshot shows a web browser window titled "Create - My ASP.NET Application". The address bar indicates the URL is "localhost:44300/Students/Create". The page content is as follows:

Application name

## Create

**Student**

---

Name

Email

Age

[Back to List](#)

---

© 2025 - My ASP.NET Application