Course Code: CT2207	Course Name: Lab: Web Technologies

Name of Students: Badal Wanjari	Semester/ Section: 3rd/ B
Roll No: 140	Enroll No: 20011045

Practical No: 8

Aim: Write Theory regarding Dropdown List Control in ASP.Net Framework. Create a Web Form which will accept two numbers as input and perform an operation depending on value selected from dropdown list control.

THEORY:

We all must have seen various web pages, which are having drop-down which consist of various options .it is usually a list with options. Mostly in the registration forms or while entering our details while signing on any page, we fill our details, there we can see the drop-down list with countries, states, area, etc. Dropdown List control is usually used by the developer to give a chance to select one option out of multiple options given in the drop-down list or listed items. In this topic, we are going to learn about the drop-down list in asp.net. It is used to store multiple items. Dropdown list control is also named as Combo box control.

Syntax

> *<asp:DropDownList ID="Trainings" runat="server">*
>
> *<asp:ListItem Enabled="true" Text= "Select Subject" Value= "-1"></asp:ListItem>*
>
> *<asp:ListItem Text= "Science" Value="1"><asp:ListItem>*
>
> *< asp:ListItem Text= "Politics" Value="2"><asp:ListItem>*
>
> *</asp:DropDownList>*

Properties of Drop-Down List in ASP.NET

Let us see some of the important properties of drop-down list in ASP .NET

- **DropDownList1.Items.Count:** it is used to provide the total number of options or items in the drop-down list.

- **DropDownList1.Items.Add("ItemName"):** suppose if we want to add some new item, then this property is useful for adding the item in the drop-down list.

- **DropDownList1.Items.Remove("ItemName"):** It will help to remove the item from the drop-down list.

- **DropDownList1.Items.Insert(int index, "ItemName"):** if we want the item to be added at a specific position, this property helps to add a new item at a specific position in the drop-down list control.

- **DropDownList1.Items.RemoveAt(int index):** it will remove the specific item from specific position (index) from the drop down list control.

- **DropDownList1.Items.Clear():** if we don't want all the items and now, we want to add another or maybe we want to change the items from the drop-down list.so it's better to clear all the items first. This property is used to clear all the provided items from the drop-down list.

- **DropDownList1.SelectionItem.Text:** this is one of the important properties because it will return the text value which is in the selected items in the drop-down list.

- **DropDownList1.SelectedIndex:** index will always start from zero. When we select any item from the drop-down list, it is associated with the index. This property will return the position of the selected item that is its index value.

- **DropDownList1.DataSource:** it is mostly the DataTable or DataSet.

- **DropDownList1.DataValueField:** It will bind the value to the drop-down list which will be visible in the dropdown list.

- **Item**: it will provide the collection of the items from the drop-down list.

- **AutoPostBack:** Its value is 'true' or 'false'. True represents that form is posted back automatically to the server when the user changes the dropdown list selection.

- **DataTextField**: values will be visible by the end-users.it is used to set the text in the Dropdown list control.

- **DataValueFeild**: This is used to set the name of the column as a value in the drop-down list. This value is not visible by the end-user.

Program:

ASP.NET is a framework not a specific programming or markup language like JavaScript, HTML, CSS. Therefore, Web Apps/ Web Form created using ASP.Net has many files. Among them, two main files are:

- **Default.aspx**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="WebFormsControlls.WebControls" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style>
      .main{
          width: 100%;
          height: 90vh;
          display: flex;
```

```
                justify-content: center;
                align-items: center;
                flex-direction: column;
            }
            tr{
                padding: 10px 5px;
                margin: 5px 10px;
                display: flex;
                width: 100%;
                align-items: center;
                justify-content: center;
                min-height: 10px;
                min-width: 30%;
                font-size: 1.325rem;
            }
            button{
                padding : 5px 10px;
            }
            td:first-child{
                width: 200px;
                margin-right: 20px
            }
            td:nth-child(2){
                width: 100px;
            }
            tr:nth-child(4), input, h1{
                text-align: center;
            }
            tr:nth-child(5) td:nth-child(2){
                font-size: 2rem;
                text-align: left;
            }
            #SubmitButton, #DropDownList2{
                padding: 4px 5px;
                width: 200px;
                text-align: center;
            }
        </style>
    </head>
    <body>
        <div class="main">
        <div>
            <h1>Practical-8</h1>
            <h3>Create a Web Form which will accept two numbers as input and
perform an operation depending on value selected from
dropdown list control.</h3>
        </div>
        <form id="form1" runat="server">
        <table>
            <tr>
                <td><asp:Label ID="label1" runat="server">Enter First Number :
</asp:Label></td>
                <td><asp:TextBox ID="num1" runat="server" type="number" ToolTip="Enter first
number for calculation"></asp:TextBox></td>
            </tr>
            <tr>
                <td> <asp:Label ID="label2" runat="server">Enter Second Number :
</asp:Label></td>
                <td> <asp:TextBox ID="num2" runat="server" type="number" ToolTip="Enter second
number for calculation"></asp:TextBox></td>
            </tr>
            <tr>
                <td>Operator : </td>
                <td>
                    <asp:DropDownList ID="DropDownList2" runat="server">
```

```
                    <asp:ListItem Value="0" Text="NONE">Select Operator First</asp:ListItem>
                    <asp:ListItem Value="1" Text="Addtion">Addtion</asp:ListItem>
                    <asp:ListItem Value="2" Text="Subtraction">Subtraction</asp:ListItem>
                    <asp:ListItem Value="3"
    Text="Multiplication">Multiplication</asp:ListItem>
                    <asp:ListItem Value="4" Text="Division">Division</asp:ListItem>
                    <asp:ListItem Value="5" Text="Modulus">Modulus</asp:ListItem>
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td colspan="2"><asp:Button ID="SubmitButton" runat="server" Text="Submit"
    OnClick="SubmitButton_Click"/></td>
        </tr>
        <tr>
            <td><asp:Label ID="AnsText" runat="server">Answer : </asp:Label></td>
            <td><asp:Label ID="userInput" runat="server">Calculating....</asp:Label></td>
        </tr>
    </table>
    </form>
    </div>
</body>
</html>
```

- **Default.aspx.cs**

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace WebFormsControlls
{
    public partial class WebControls : System.Web.UI.Page
    {
        protected void SubmitButton_Click(object sender, EventArgs e)
        {
            if(num1.Text == "" || num2.Text == "")
            {
                AnsText.Text = "Error :";
                userInput.Text = "Enter Values First";
            }
            else
            {
                int n1 = Convert.ToInt32(num1.Text);
                int n2 = Convert.ToInt32(num2.Text);
                int opt = Convert.ToInt32(DropDownList2.SelectedValue.ToString());
                if(opt == 0)
                {
                    AnsText.Text = "Error :";
                    userInput.Text = "Select Operator From Dropdown List";
                }
                else if(opt == 1)
                {
                    AnsText.Text = "Answer :";
                    userInput.Text = Convert.ToString(n1 + n2);
                }
                else if (opt == 2)
                {
                    AnsText.Text = "Answer :";
```

```
                    userInput.Text = Convert.ToString(n1 - n2);
                }
                else if (opt == 3)
                {
                    AnsText.Text = "Answer :";
                    userInput.Text = Convert.ToString(n1 * n2);
                }
                else if (opt == 4)
                {
                    AnsText.Text = "Answer :";
                    userInput.Text = Convert.ToString(n1 / n2);
                }
                else if (opt == 5)
                {
                    AnsText.Text = "Answer :";
                    userInput.Text = Convert.ToString(n1 % n2);
                }
            }
        }
    }
}
```
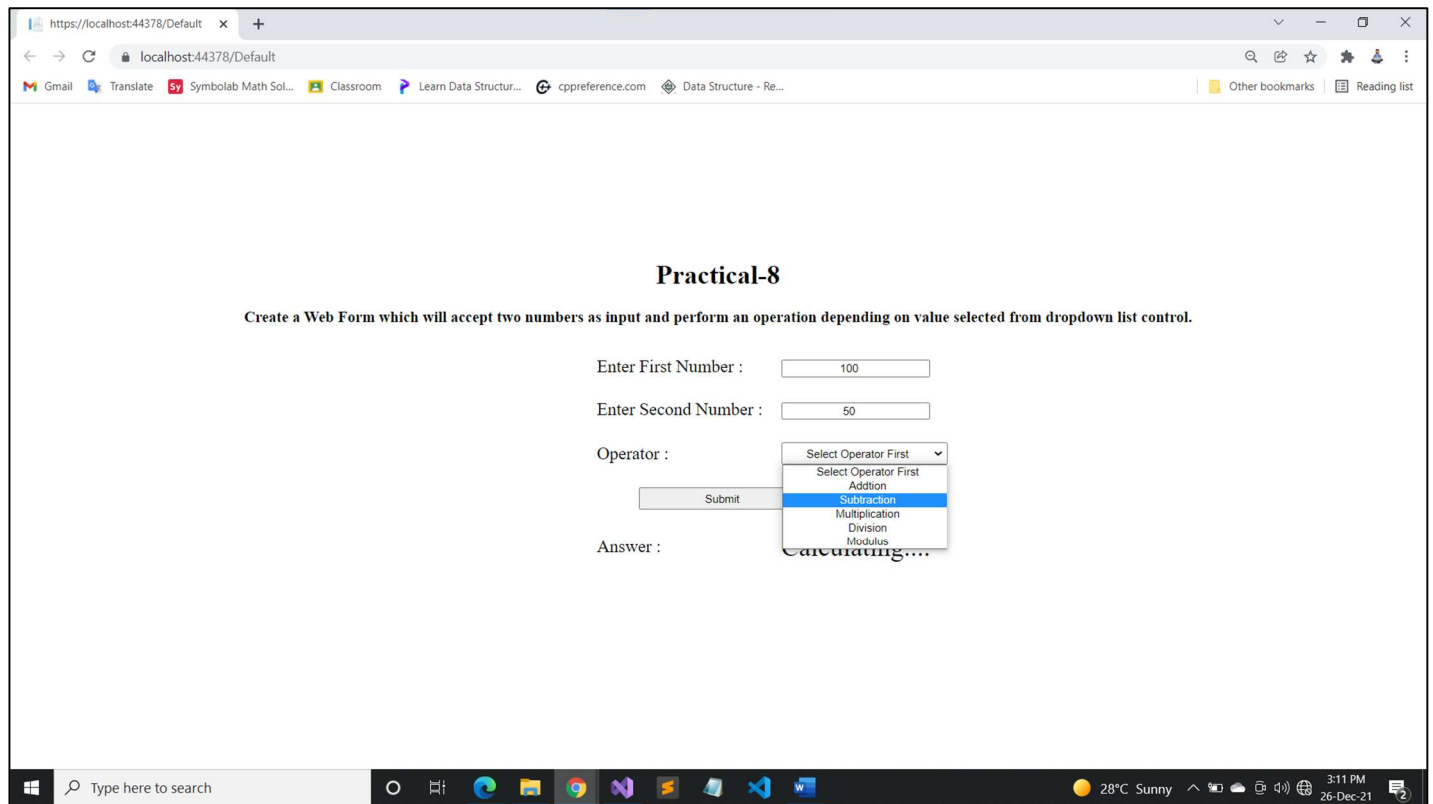
OUTPUT SCREEN SHOTS:

Practical-8

Create a Web Form which will accept two numbers as input and perform an operation depending on value selected from dropdown list control.

Enter First Number :   100

Enter Second Number :   50

Operator :   Subtraction

Submit

Answer :   **50**

CONCLUSION: In this way, I have created a web-form using ASP.Net Web Form which will accept two numbers as input and perform an operation depending on value selected from dropdown list control. Then output answer in web form.