



**AMIT ACADEMY**  
for Computer Education



**Subject Title: WEB DEVELOPMENT USING ASP.NET**

**Subject Code: BCA 501**

## **Unit 2: Web Forms Standard Controls**

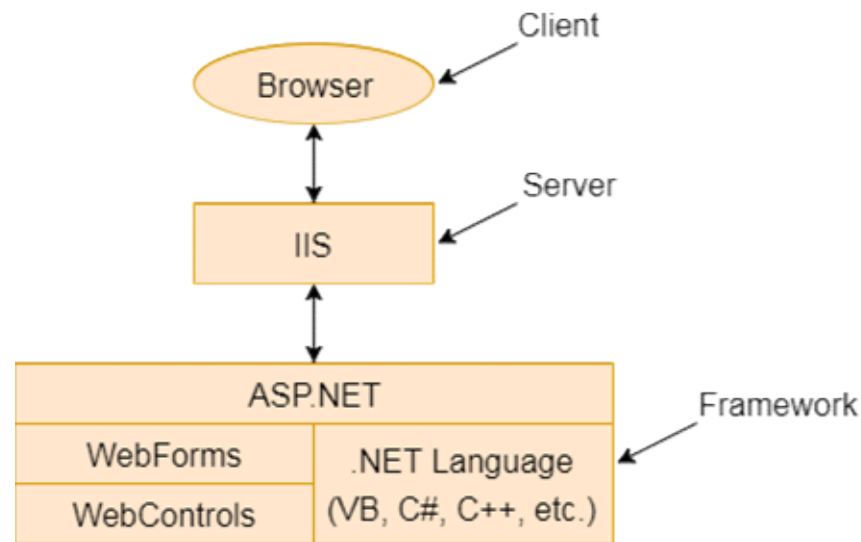
**Prepared by Ankit Rami(AR) Any Query**

**Contact – +91 8460467193**

**Email – [ankitramiblog@gmail.com](mailto:ankitramiblog@gmail.com)**

# ASP.NET Web Forms?

- Web Forms are web pages built on the ASP.NET Technology. It executes on the server and generates output to the browser.
- It is compatible to any browser to any language supported by .NET common language runtime.
- It is flexible and allows us to create and add custom controls.
- We can use Visual Studio to create ASP.NET Web Forms. It is an IDE (Integrated Development Environment) that allows us to drag and drop server controls to the web forms.
- It also allows us to set properties, events and methods for the controls. To write business logic, we can choose any .NET language like: Visual Basic or Visual C#.

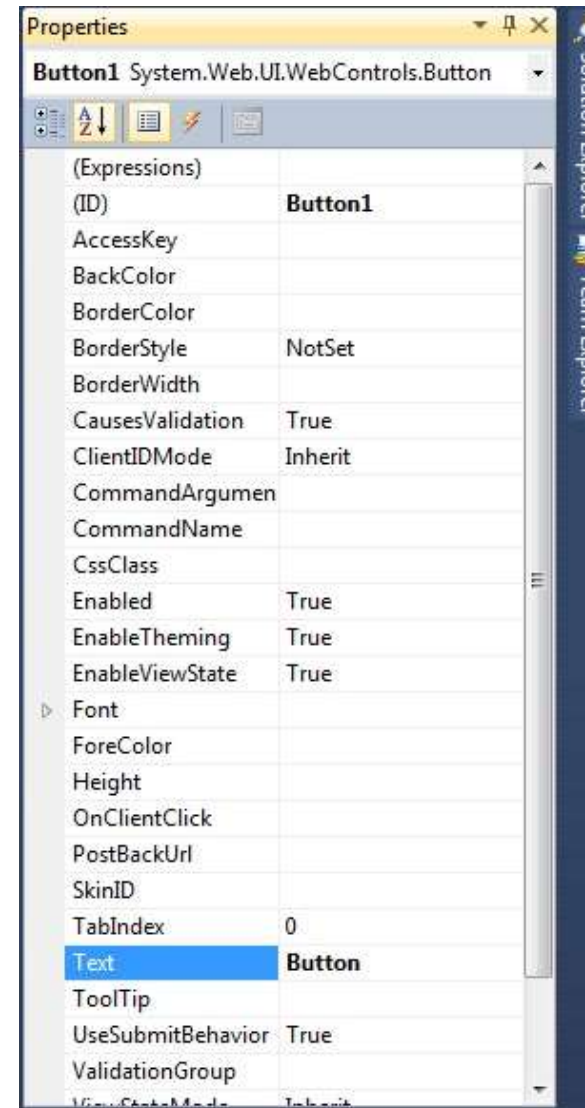
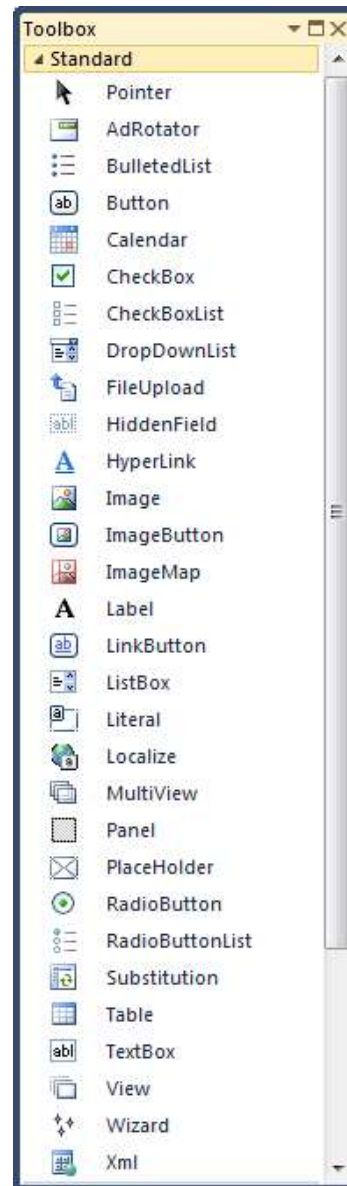


# HTML Control VS ASP.NET Standard Controls

HTML Controls	ASP.Net Controls
HTML control runs at <b>client side</b> .	ASP.Net controls run at <b>server side</b> .
You can run HTML controls at server side by adding attribute <b>runat="server"</b> .	You can not run ASP.Net Controls on client side as these controls have this attribute <b>runat="server"</b> by default.
HTML controls are client side controls, so it does not provide <b>STATE management</b> .	ASP.Net Controls are Server side controls, provides STATE management.
HTML control does not require rendering.	ASP.Net controls require <b>rendering</b> .
As HTML controls runs on client side, execution is <b>fast</b> .	As ASP.Net controls run on server side, execution is <b>slow</b> .
HTML controls do not have any separate <b>class</b> for its controls.	ASP.Net controls have separate class for its each control.
HTML controls does not support <b>Object Oriented paradigm</b> .	With ASP.Net controls, you have full support of Object oriented paradigm.
HTML controls can not be accessed form code behind files.	ASP.Net controls can be directly worked and accessed from <b>code behind files</b> .
HTML control have <b>limited</b> set of properties and/or methods.	ASP.Net controls have <b>rich</b> set of properties and/or methods.
<pre> Enter Name : <input id="txtName" type="text"/>           </pre>	<pre> Enter Name : &lt;asp:TextBox Id="txtName" runat="server"&gt; &lt;/asp:TextBox&gt;           </pre>

# 📢 ASP.NET Standard Controls - I:

- ❑ Label
- ❑ Button
- ❑ TextBox
- ❑ Placeholder
- ❑ FileUpload
- ❑ HiddenField

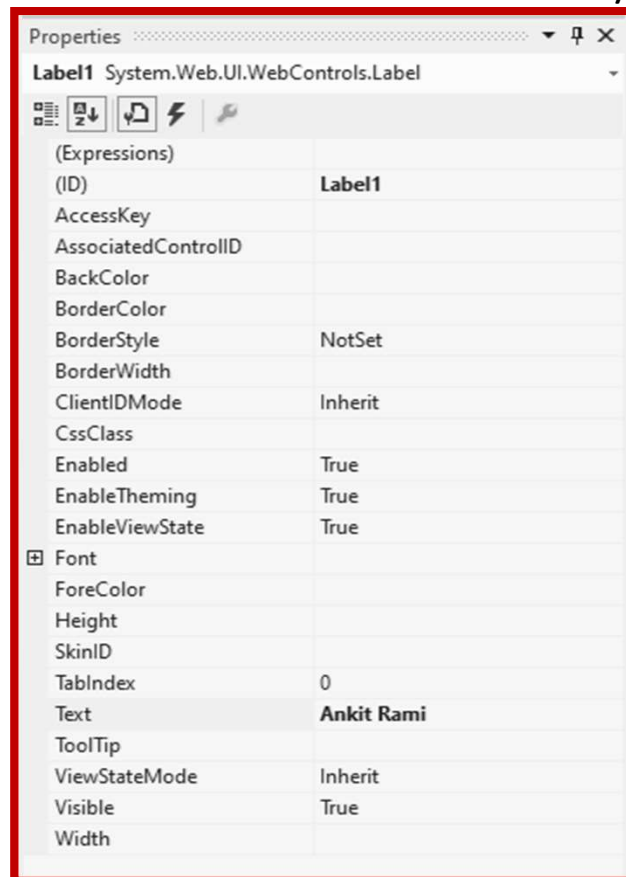




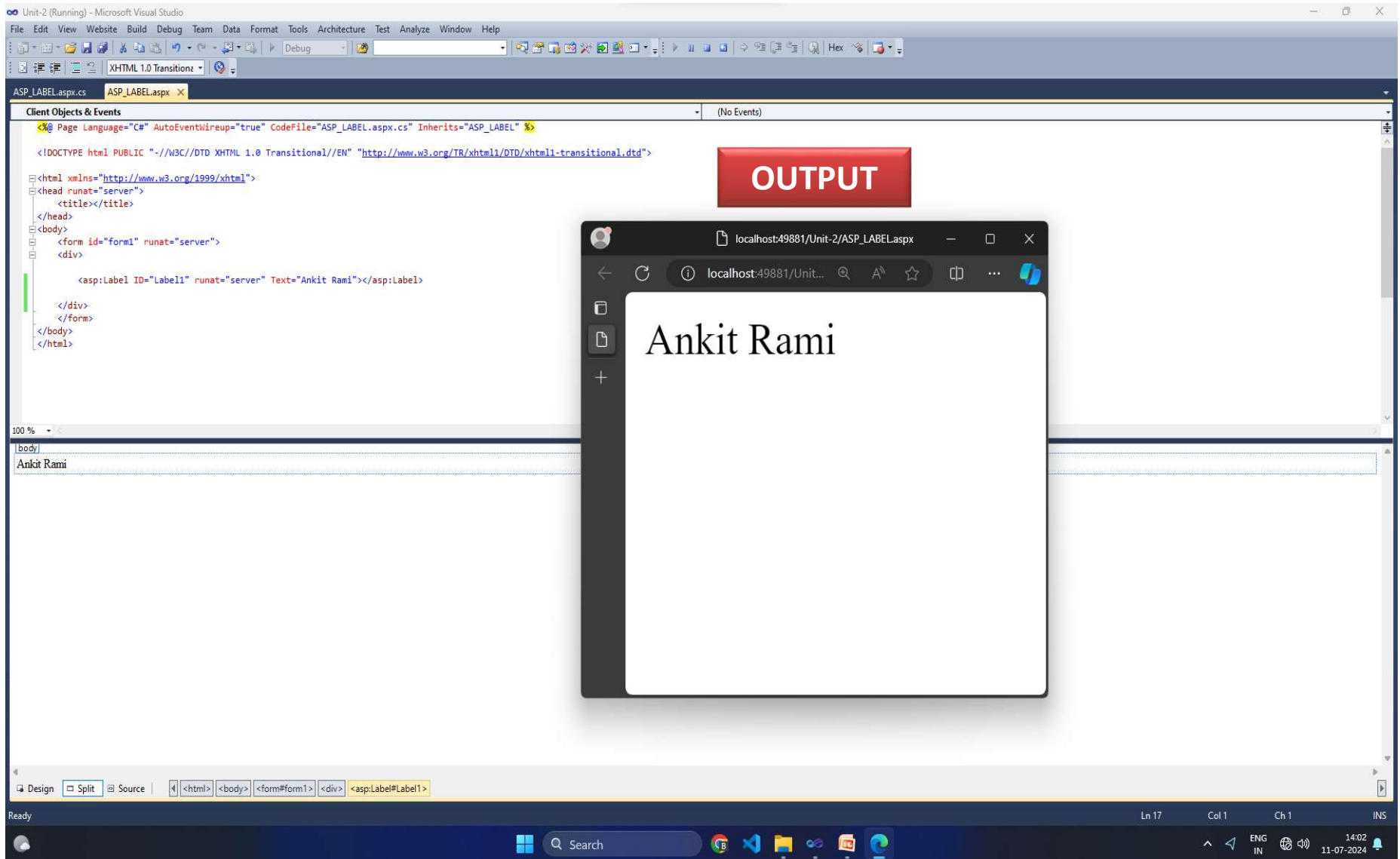
## ASP.NET Standard Controls - I: Label

- ✓ ASP:Label is a Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ This control is used to display textual information on the web forms. It is mainly used to create caption for the other controls like: textbox.

Ex- `<asp:LabelID="Label1" runat="server" Text="Label" ></asp:Label>`



# ASP.NET Standard Controls - I: Label



The screenshot displays the Microsoft Visual Studio IDE with a project named 'Unit-2 (Running)'. The 'Client Objects & Events' window shows the source code for 'ASP\_LABEL.aspx.cs'. The code defines a page with a form containing a label. The label's text is 'Ankit Rami'. A red box labeled 'OUTPUT' is positioned above a browser window that shows the rendered output of the page, which is the text 'Ankit Rami'.

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

<asp:Label ID="Label1" runat="server" Text="Ankit Rami"></asp:Label>

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

localhost:49881/Unit-2/ASP\_LABEL.aspx

localhost:49881/Unit...

Ankit Rami

body  
Ankit Rami

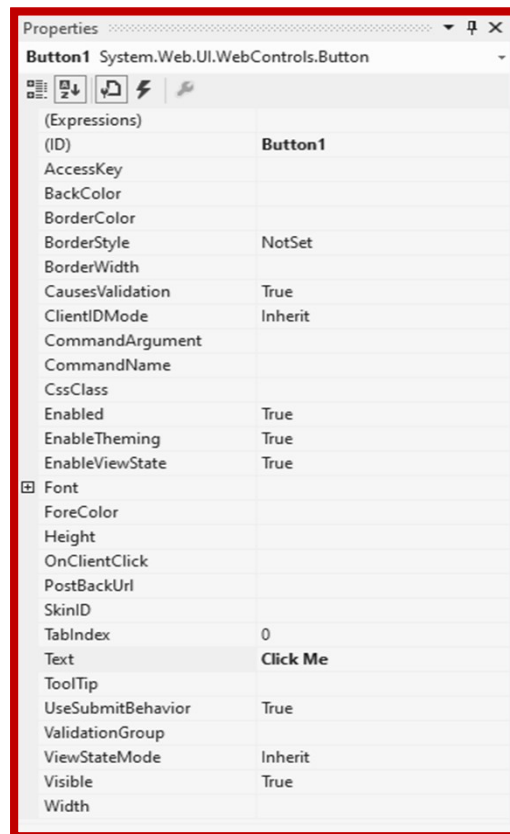
Design Split Source <html> <body> <form#form1> <div> <asp:Label#Label1>

Ready Ln 17 Col 1 Ch 1 INS

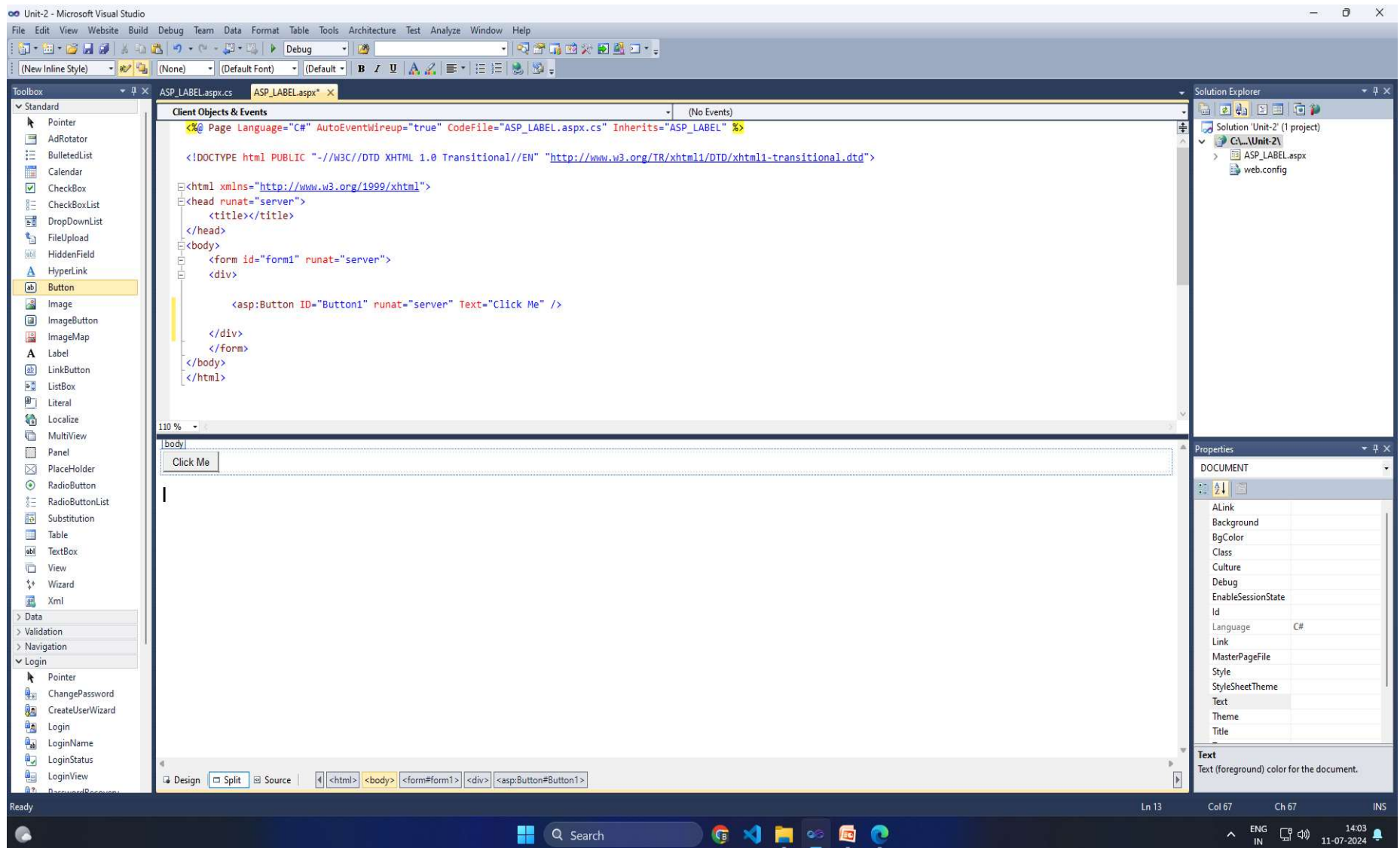
Search 14:02 11-07-2024

## ASP.NET Standard Controls - I: Button

- ✓ ASP:Button is Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ This control is used to perform events. It is also used to submit client request to the server. To create **Button** either we can write code or use the drag and drop facility of visual studio IDE.
- ✓ Ex- `<asp:ButtonID="Button1" runat="server" Text="Submit" ToolTip="Submit"/>`



# 📢 ASP.NET Standard Controls - I: Button



The screenshot displays the Microsoft Visual Studio IDE with the following components:

- Toolbox:** The 'Standard' category is expanded, and the 'Button' control is selected.
- Client Objects & Events:** Shows '(No Events)'.
- Code Editor:** Contains the following ASP.NET code:
 

```

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

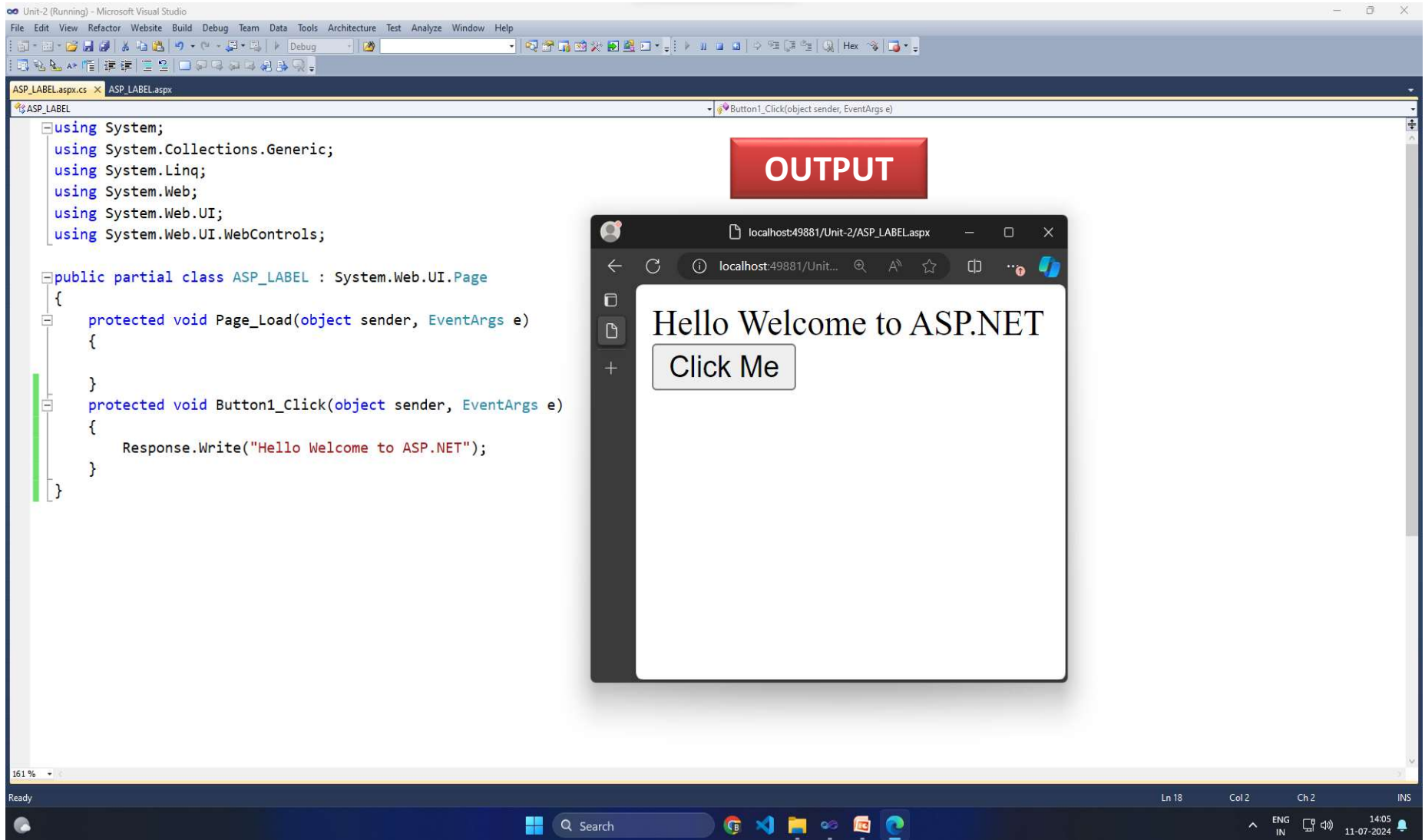
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>

<asp:Button ID="Button1" runat="server" Text="Click Me" />

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

```
- Design View:** Shows a visual representation of the button with the text 'Click Me'.
- Solution Explorer:** Shows the project structure for 'Unit-2' with files 'ASP\_LABEL.aspx' and 'web.config'.
- Properties Window:** Shows the 'DOCUMENT' properties, with 'Text' set to 'Text (foreground) color for the document.'
- Bottom Status Bar:** Shows 'Ln 13 Col 67 Ch 67 INS'.

# ASP.NET Standard Controls - I: Button



The screenshot displays the Microsoft Visual Studio IDE. The left pane shows the source code for `ASP_LABEL.aspx.cs`. The code includes the following:

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

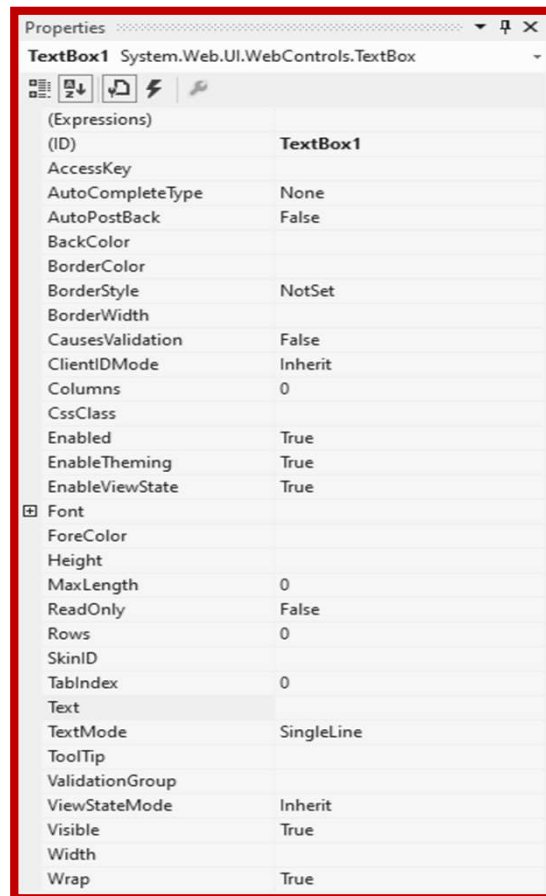
public partial class ASP_LABEL : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Response.Write("Hello Welcome to ASP.NET");
    }
}
```

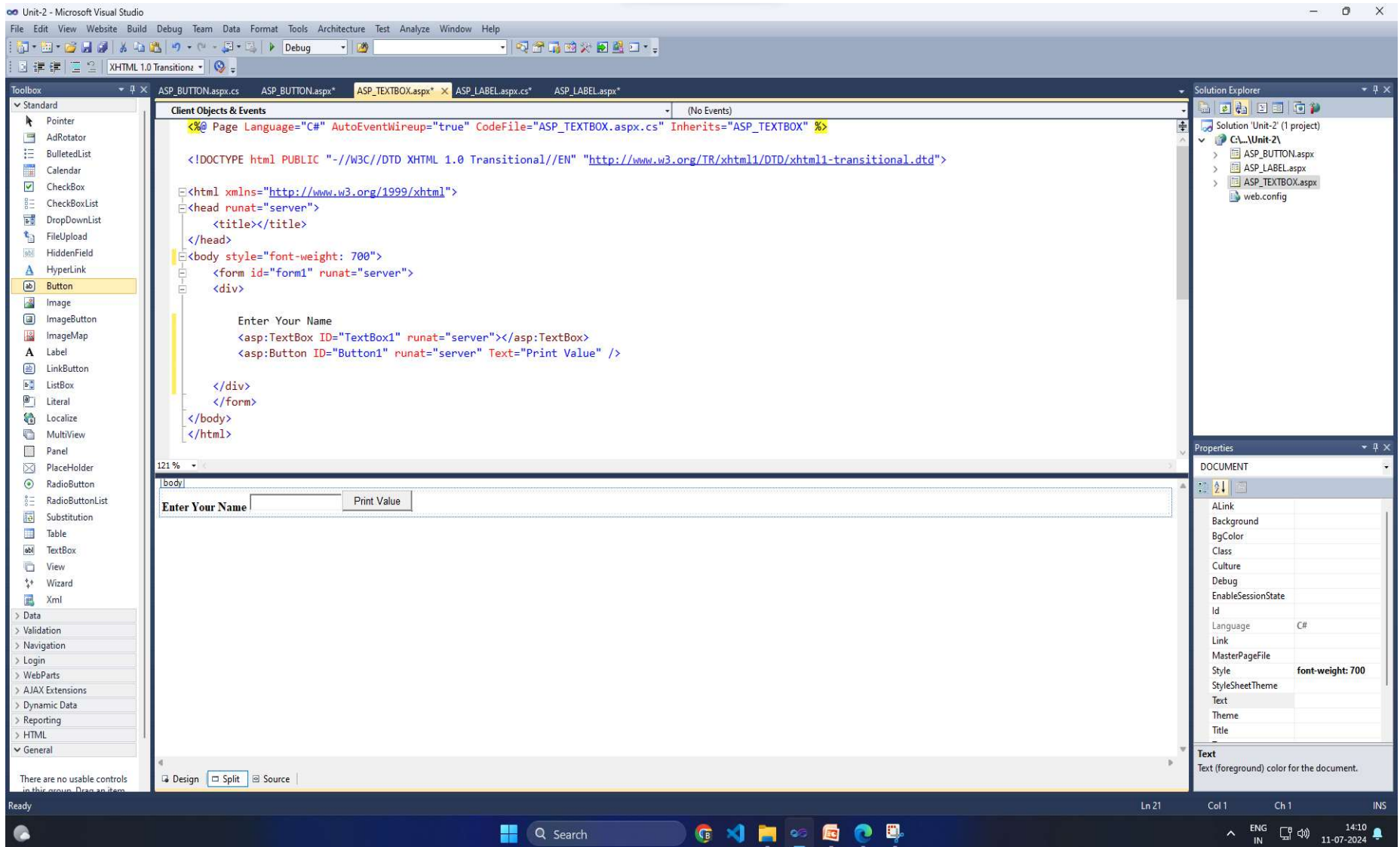
The right pane shows the browser output. A red box labeled "OUTPUT" is positioned above the browser window. The browser window displays the text "Hello Welcome to ASP.NET" and a button labeled "Click Me".

## ASP.NET Standard Controls - I: TextBox

- ✓ ASP:TextBox is Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ This is an input control which is used to take user input. To create **TextBox** either we can write code or use the drag and drop facility of visual studio IDE.
- ✓ Ex- `<asp:TextBox ID="TextBox1" runat="server" ></asp:TextBox>`



# 📢 ASP.NET Standard Controls - I: TextBox



The screenshot displays the Microsoft Visual Studio IDE with the following components:

- Toolbox:** The 'Standard' group is expanded, showing various ASP.NET controls. The 'TextBox' control is highlighted.
- Client Objects & Events:** Shows '(No Events)'.
- Code View:** Displays the following ASP.NET code:
 

```

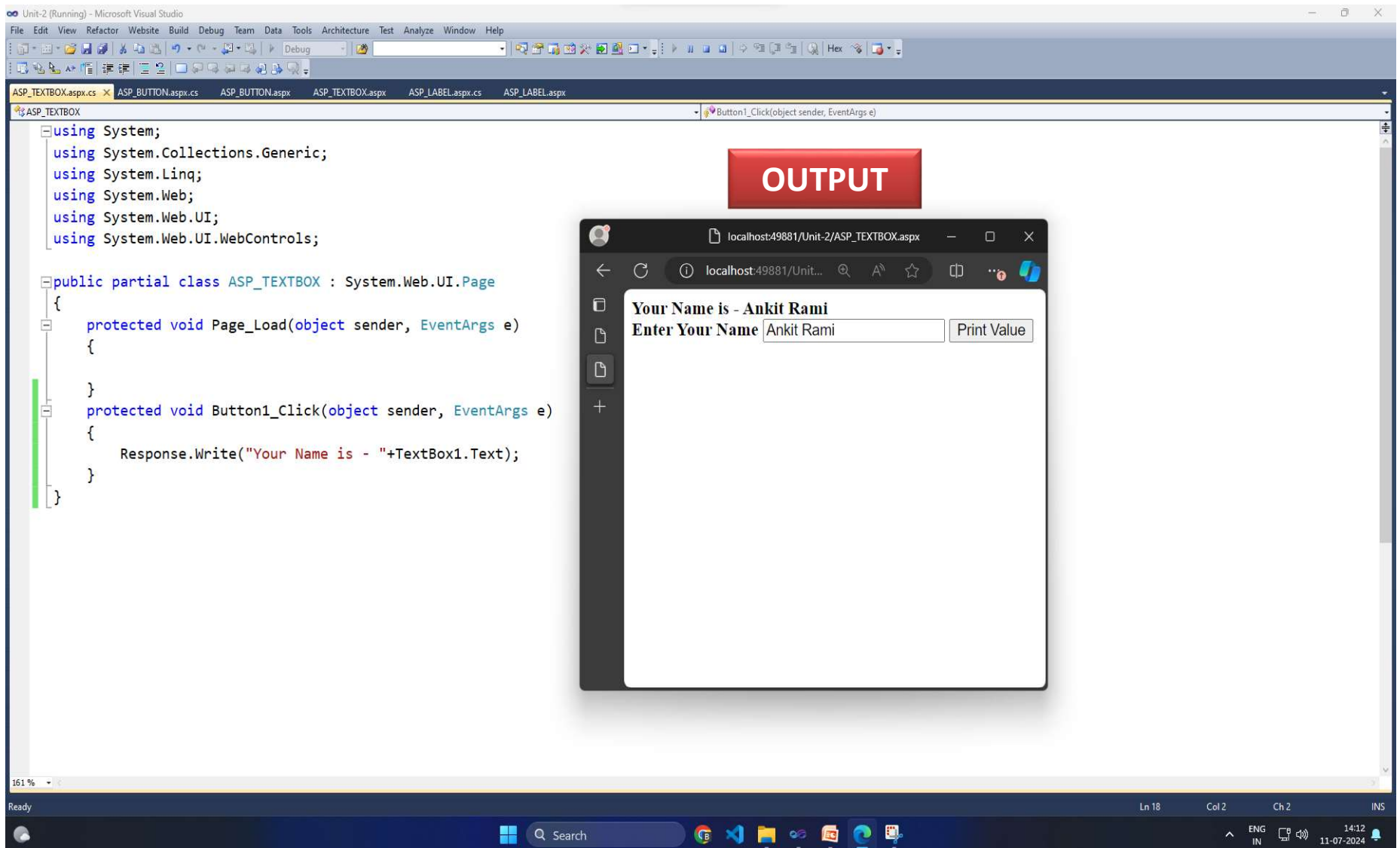
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ASP_TEXTBOX.aspx.cs" Inherits="ASP_TEXTBOX" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<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>
Enter Your Name
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<asp:Button ID="Button1" runat="server" Text="Print Value" />
</div>
</form>
</body>
</html>
      
```
- Design View:** Shows a visual representation of the code above, featuring a text input field with the placeholder text 'Enter Your Name' and a 'Print Value' button.
- Solution Explorer:** Shows the project structure for 'Unit-2', including files like ASP\_BUTTON.aspx, ASP\_LABEL.aspx, and ASP\_TEXTBOX.aspx.
- Properties Window:** Shows the 'DOCUMENT' properties, with 'Style' set to 'font-weight: 700'.



# ASP.NET Standard Controls - I: TextBox



The screenshot displays the Microsoft Visual Studio IDE with a C# code file named `ASP_TEXTBOX.aspx.cs` open. The code defines a partial class `ASP_TEXTBOX` that inherits from `System.Web.UI.Page`. It includes the following code:

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

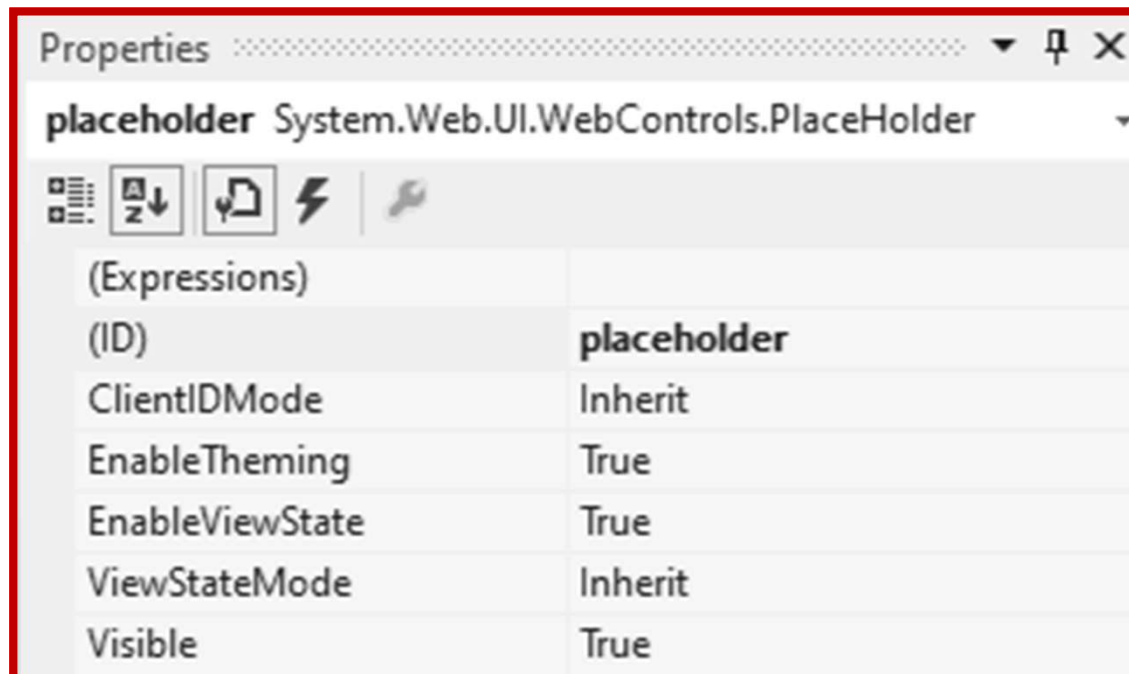
public partial class ASP_TEXTBOX : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Response.Write("Your Name is - "+TextBox1.Text);
    }
}
```

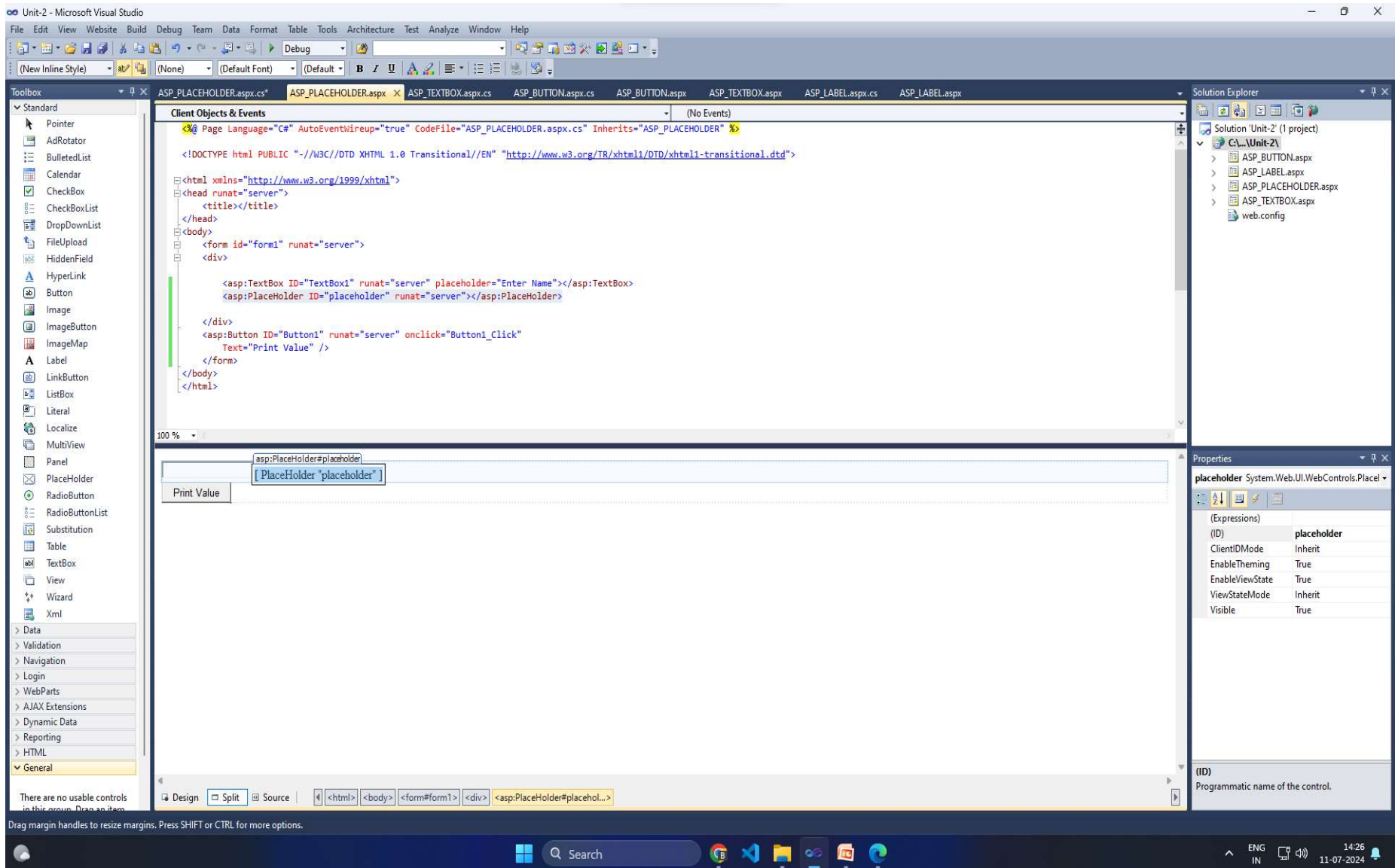
Overlaid on the right side of the IDE is a browser window showing the rendered output. A red box with the word "OUTPUT" is positioned above the browser window. The browser window displays the text "Your Name is - Ankit Rami" and "Enter Your Name" followed by a text input field containing "Ankit Rami" and a "Print Value" button.

## ASP.NET Standard Controls - I: Placeholder

- ✓ ASP:Placeholder is Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ The Placeholder Web server and server control does not have any visible output and is used as a place holder when we add controls at run time.
- ✓ Ex- `<asp:Placeholder ID="placeholder" runat="server"></asp:Placeholder>`



# 📣 ASP.NET Standard Controls - I: Placeholder



The screenshot displays the Microsoft Visual Studio environment for a project named 'Unit-2'. The main window shows the source code for 'ASP\_PLACEHOLDER.aspx.cs', which includes the following code:

```

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

<asp:TextBox ID="TextBox1" runat="server" placeholder="Enter Name"></asp:TextBox>
<asp:Placeholder ID="placeholder" runat="server"></asp:Placeholder>

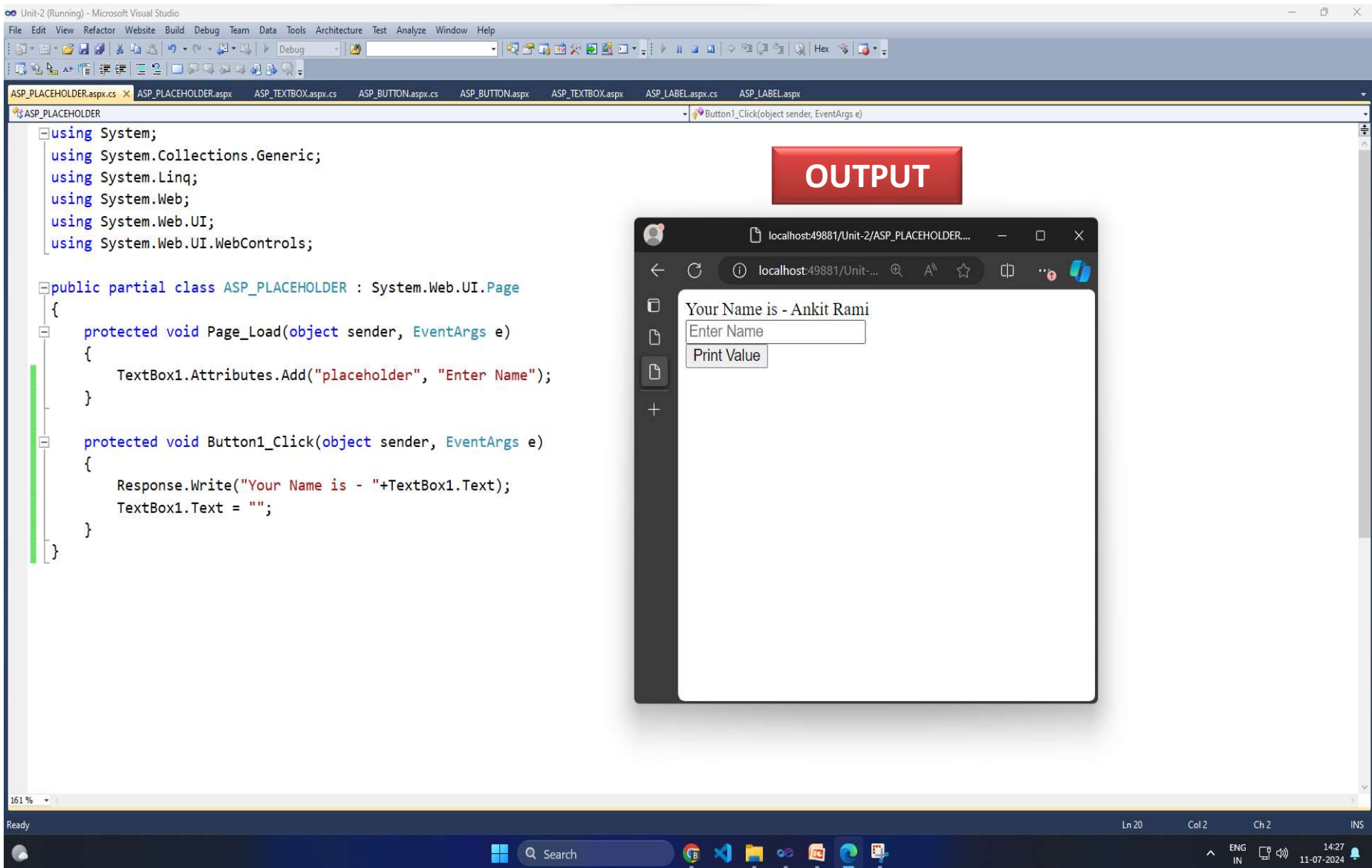
</div>
<asp:Button ID="Button1" runat="server" onclick="Button1_Click"
Text="Print Value" />
</form>
</body>
</html>
  
```

The Design view shows a visual representation of the page with a 'Print Value' button and a placeholder control labeled 'asp:Placeholder#placeholder' containing the text '[Placeholder "placeholder"]'. The Properties window for the placeholder control shows the following properties:

(ID)	placeholder
ClientIDMode	Inherit
EnableTheming	True
EnableViewState	True
ViewStateMode	Inherit
Visible	True

The Solution Explorer on the right shows the project structure for 'Unit-2', including files like ASP\_BUTTON.aspx, ASP\_LABEL.aspx, ASP\_PLACEHOLDER.aspx, ASP\_TEXTBOX.aspx, and web.config.

# ASP.NET Standard Controls - I: Placeholder



The screenshot displays the Visual Studio IDE with a C# code file named `ASP_PLACEHOLDER.cs`. The code implements a simple ASP.NET page that demonstrates the Placeholder control. A red box labeled "OUTPUT" points to a browser window showing the rendered page.

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

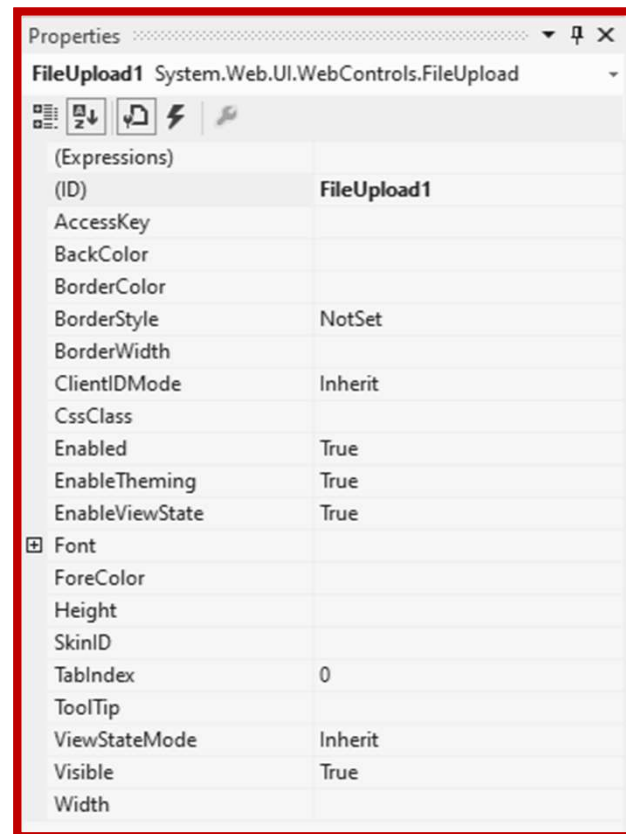
public partial class ASP_PLACEHOLDER : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        TextBox1.Attributes.Add("placeholder", "Enter Name");
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Response.Write("Your Name is - "+TextBox1.Text);
        TextBox1.Text = "";
    }
}
```

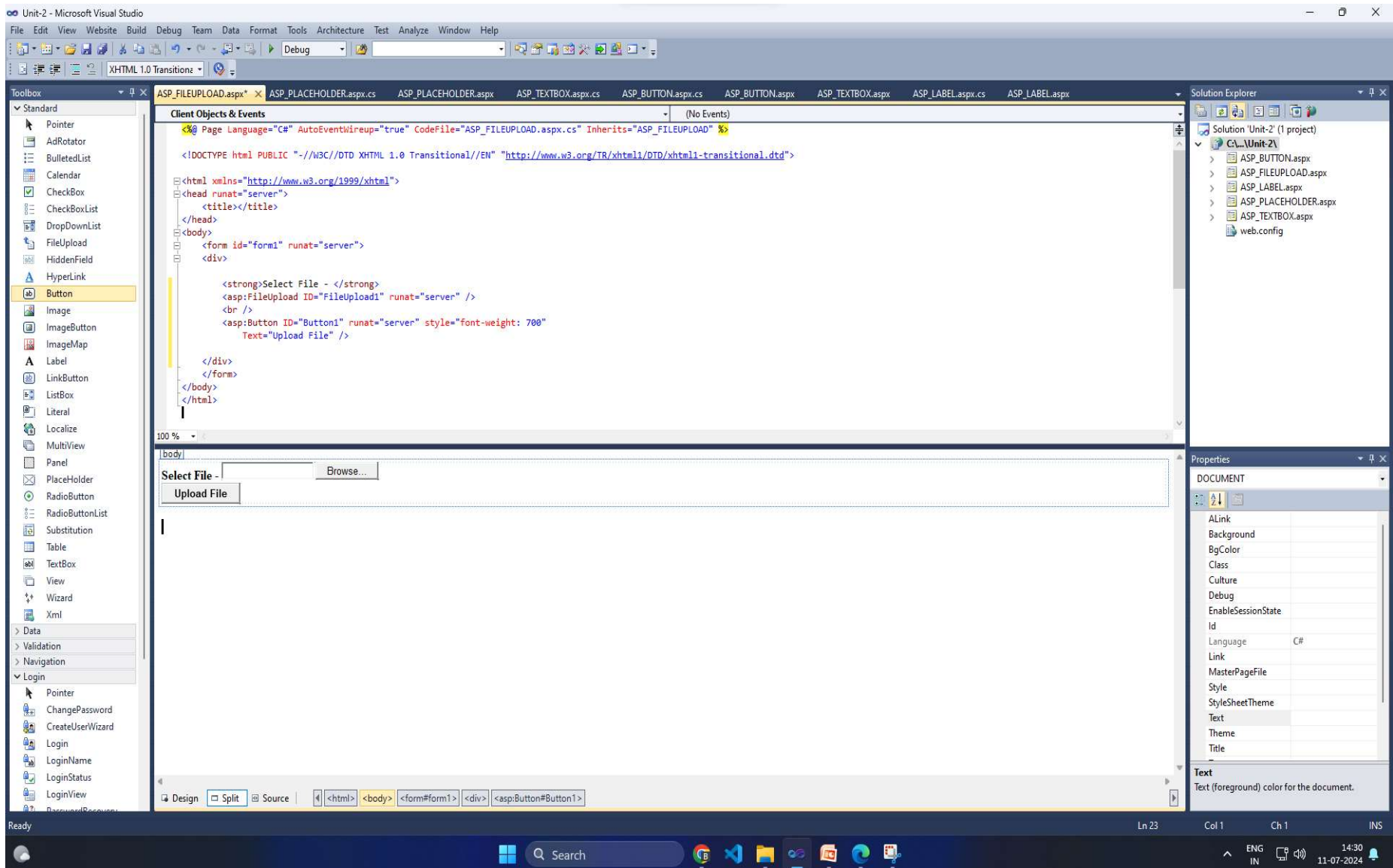
The browser window shows the output of the application. The page displays the text "Your Name is - Ankit Rami" above a text input field with the placeholder text "Enter Name" and a "Print Value" button.

## ASP.NET Standard Controls - I: FileUpload

- ✓ ASP:FileUpload is Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ It is an input controller which is used to upload file to the server. It creates a browse button on the form that pop up a window to select the file from the local machine.
- ✓ To implement FileUpload we can drag it from the toolbox in visual studio.
- ✓ Ex- `<asp:FileUpload ID="FileUpload1" runat="server"/>`



# 📢 ASP.NET Standard Controls - I: FileUpload



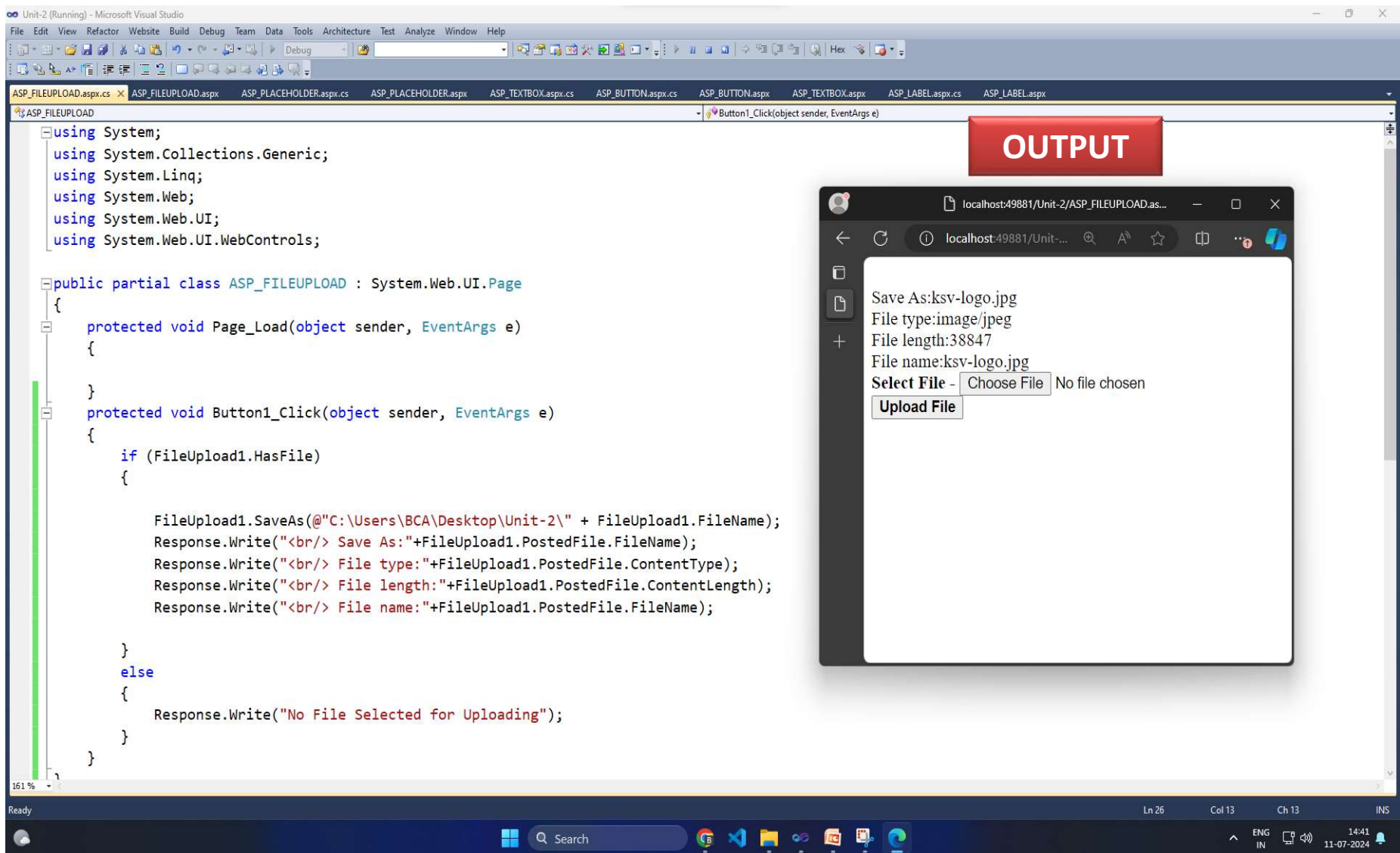
The screenshot displays the Microsoft Visual Studio IDE with the following components:

- Toolbox:** Shows various ASP.NET controls, with the **FileUpload** control selected.
- Client Objects & Events:** Shows the page's code-behind file: `ASP_FILEUPLOAD.aspx.cs`.
- Code Editor:** Contains the following HTML code:
 

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<strong>Select File - </strong>
<asp:FileUpload ID="FileUpload1" runat="server" />
<br />
<asp:Button ID="Button1" runat="server" style="font-weight: 700"
Text="Upload File" />
</div>
</form>
</body>
</html>
```
- Design View:** Shows a visual representation of the code, featuring a text box labeled "Select File -" with a "Browse..." button, and a large "Upload File" button below it.
- Properties Window:** Shows the properties for the selected document, including Language (C#), MasterPageFile, and Style.
- Taskbar:** Shows the Windows taskbar with the search bar and system tray.



# 📢 ASP.NET Standard Controls - I: FileUpload



The screenshot displays the Visual Studio IDE with a C# code file named `ASP_FILEUPLOAD.aspx.cs`. The code implements a file upload control. A red box labeled **OUTPUT** highlights the browser window showing the result of a successful file upload.

```

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

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

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        if (FileUpload1.HasFile)
        {

            FileUpload1.SaveAs(@"C:\Users\BCA\Desktop\Unit-2\" + FileUpload1.FileName);
            Response.Write("<br/> Save As:"+FileUpload1.PostedFile.FileName);
            Response.Write("<br/> File type:"+FileUpload1.PostedFile.ContentType);
            Response.Write("<br/> File length:"+FileUpload1.PostedFile.ContentLength);
            Response.Write("<br/> File name:"+FileUpload1.PostedFile.FileName);

        }
        else
        {
            Response.Write("No File Selected for Uploading");
        }
    }
}
  
```

**OUTPUT**

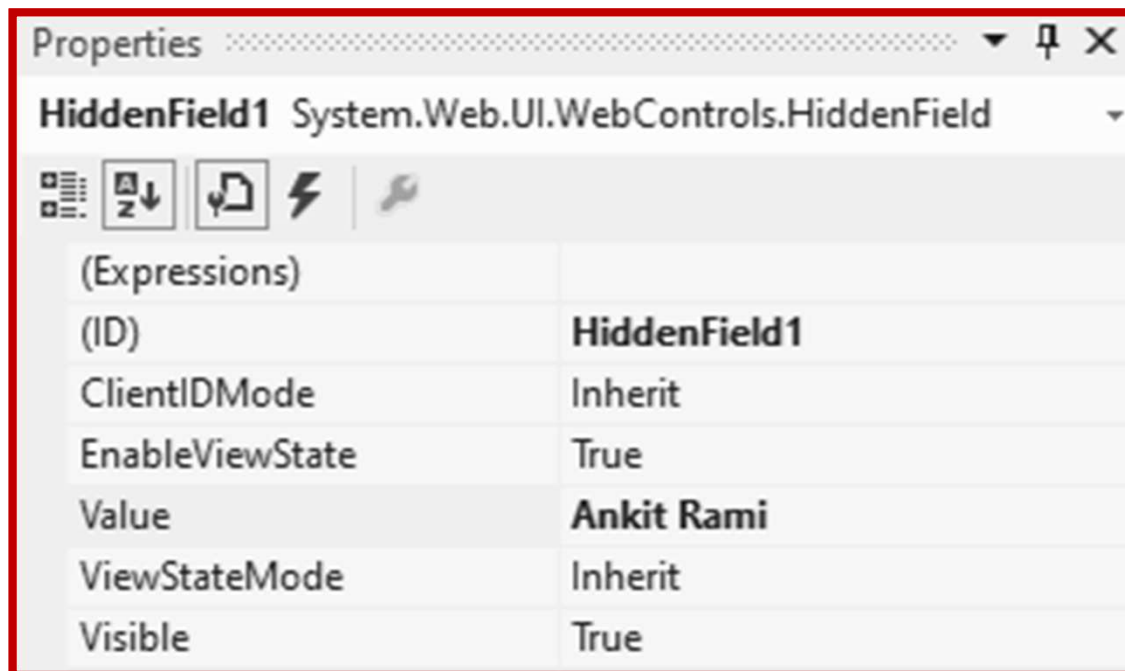
```

Save As:ksv-logo.jpg
File type:image/jpeg
File length:38847
File name:ksv-logo.jpg
Select File - Choose File No file chosen
Upload File
  
```

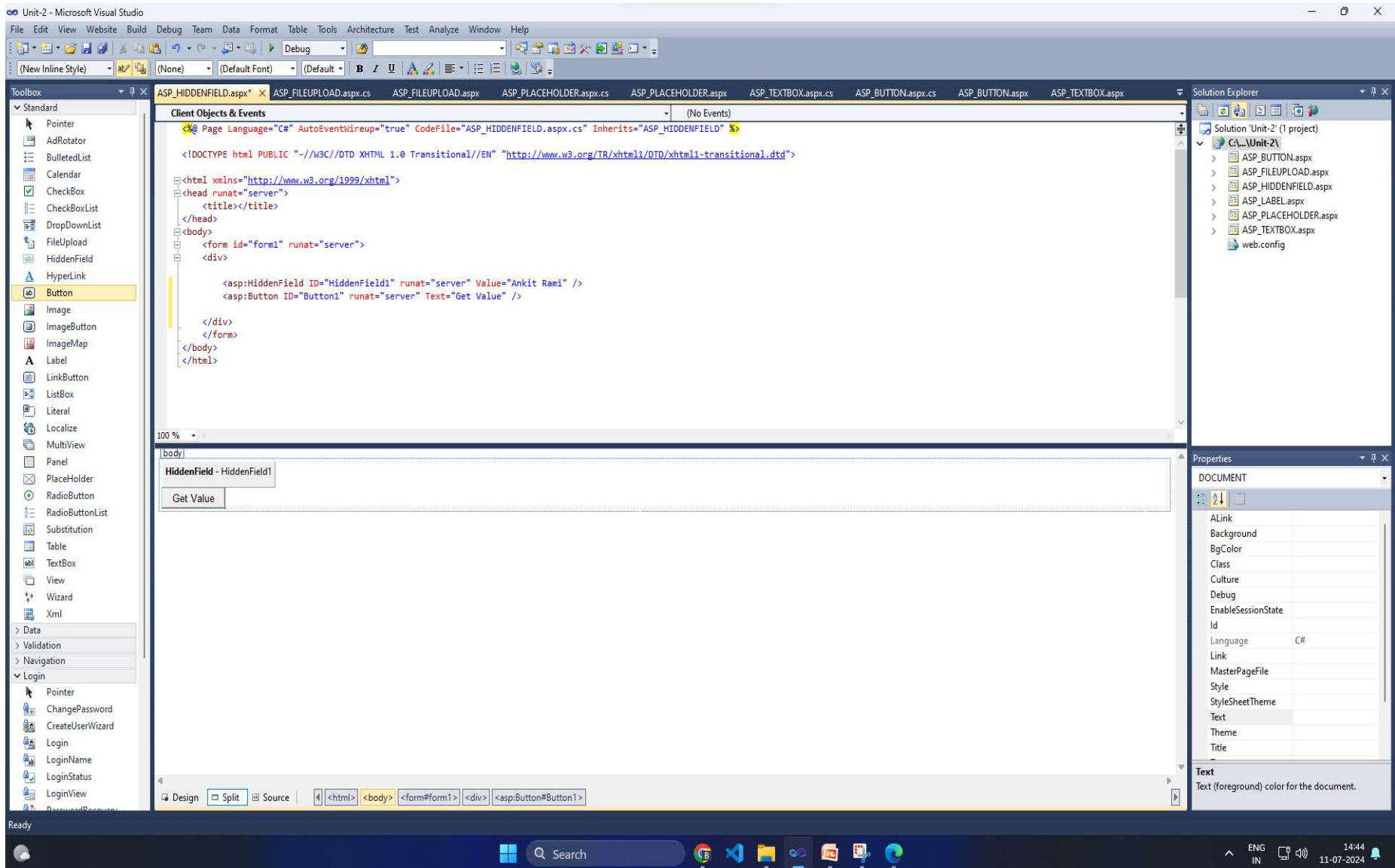


## ASP.NET Standard Controls - I: HiddenField

- ✓ ASP:HiddenField is Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ HiddenField, as name implies, is hidden. This is non visual control in ASP.NET where you can save the value. This is one of the types of client-side state management tools.
- ✓ Ex- `<asp:HiddenFieldID="HiddenField1" runat="server" />`



# 📢 ASP.NET Standard Controls - I: HiddenField



The screenshot displays the Microsoft Visual Studio IDE with the following components:

- Toolbox:** The 'Standard' category is expanded, showing various ASP.NET controls. The 'HiddenField' control is highlighted.
- Client Objects & Events:** Shows the page's metadata, including the language (C#) and the code file path.
- Source View:** Contains the following code:
 

```

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

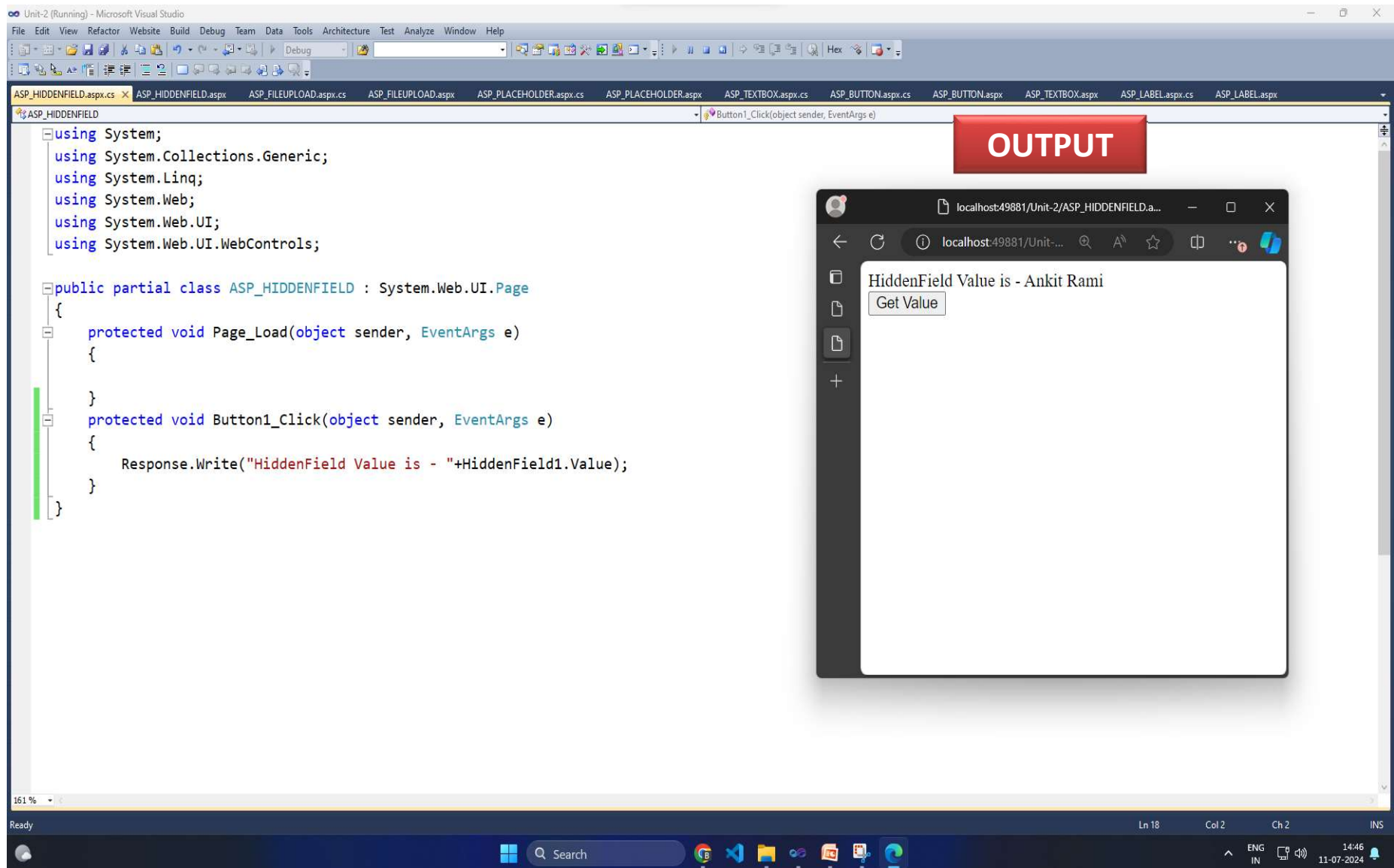
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

<asp:HiddenField ID="HiddenField1" runat="server" Value="Ankit Rami" />
<asp:Button ID="Button1" runat="server" Text="Get Value" />

</div>
</form>
</body>
</html>
      
```
- Design View:** Shows a visual representation of the page with a 'HiddenField - HiddenField1' label and a 'Get Value' button.
- Solution Explorer:** Shows the project structure for 'CA\_Unit-2', including files like ASP\_BUTTON.aspx, ASP\_FILEUPLOAD.aspx, ASP\_HIDDENFIELD.aspx, ASP\_LABEL.aspx, ASP\_PLACEHOLDER.aspx, and ASP\_TEXTBOX.aspx.
- Properties Window:** Shows the 'DOCUMENT' properties, with 'Language' set to 'C#'.

# ASP.NET Standard Controls - I: HiddenField



The screenshot displays the Visual Studio IDE with a C# code file named `ASP_HIDDENFIELD.aspx.cs`. The code defines a partial class `ASP_HIDDENFIELD` that inherits from `System.Web.UI.Page`. It includes the following code:

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

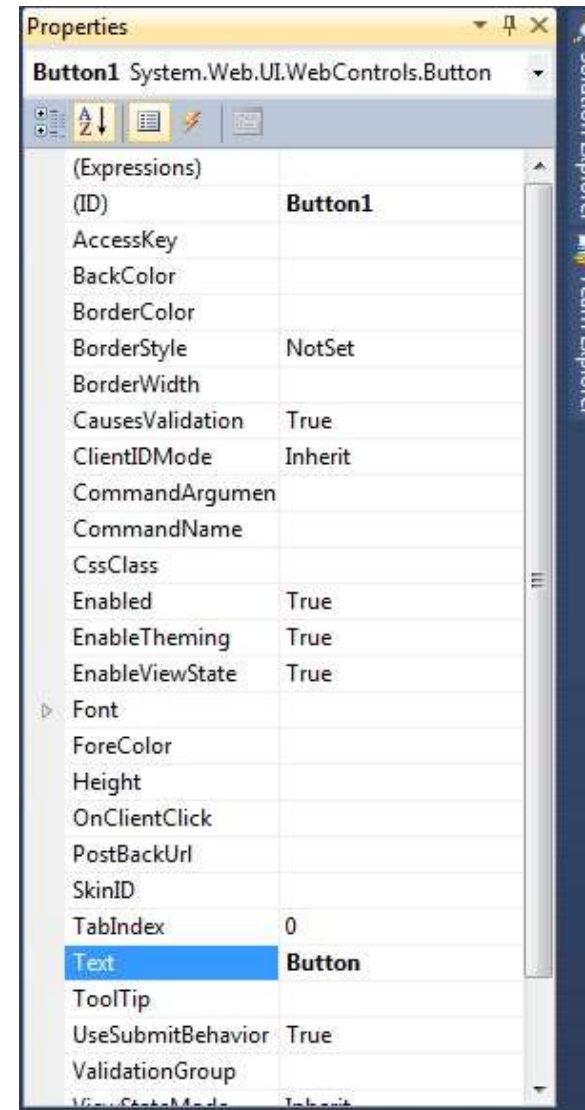
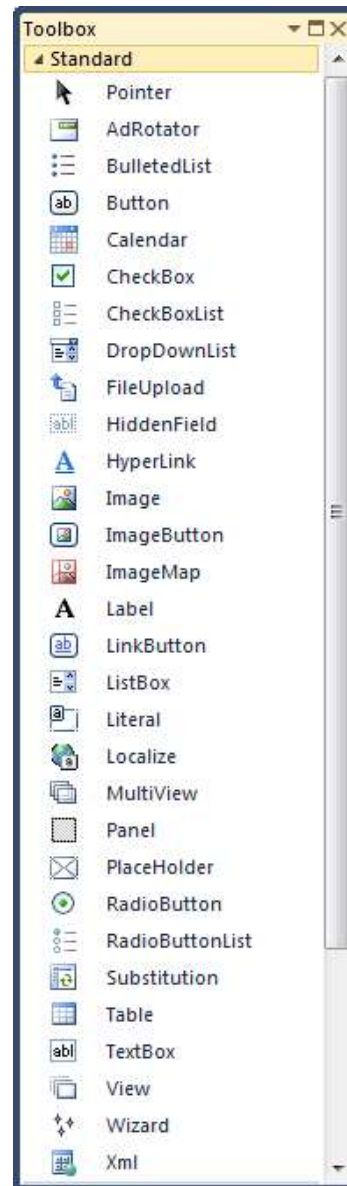
public partial class ASP_HIDDENFIELD : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Response.Write("HiddenField Value is - "+HiddenField1.Value);
    }
}
```

A red box labeled **OUTPUT** is positioned above a browser window. The browser window shows the rendered output of the page: "HiddenField Value is - Ankit Rami" and a "Get Value" button. The browser's address bar indicates the URL is `localhost:49881/Unit-2/ASP_HIDDENFIELD.a...`. The Visual Studio status bar at the bottom shows the file is at line 18, column 2, and the system tray shows the time as 14:46 on 11-07-2024.

## 📢 ASP.NET Standard Controls - II:

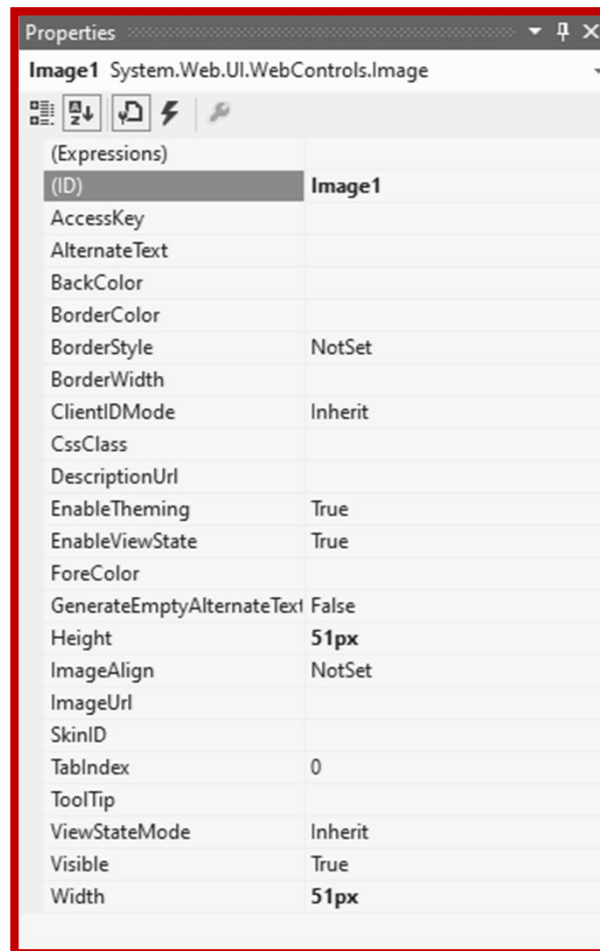
- Image
- ImageButton
- ListBox
- DropDownList
- HyperLink
- CheckBox
- CheckBoxList
- RadioButton
- RadioButtonList
- Table
- Panel
- Calendar
- AdRotator



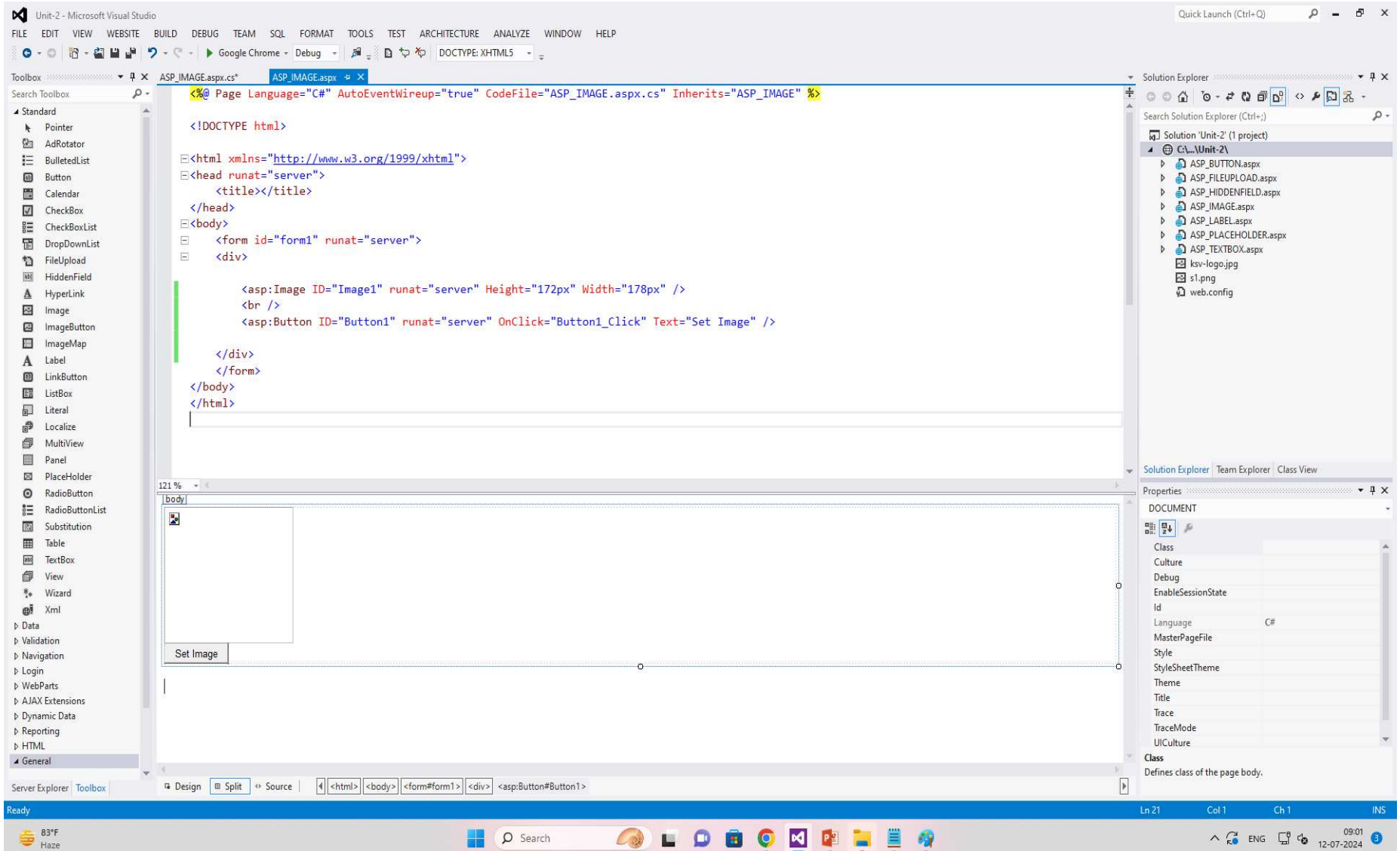
## ASP.NET Standard Controls - II: Image

- ✓ ASP:Image is a Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ This control is used to display Image on the web forms.

Ex- `<asp:Image ID="Image1" runat="server" />`



# ASP.NET Standard Controls - II: Image



The screenshot displays the Microsoft Visual Studio IDE with the following components:

- Code View:** Shows the ASP.NET page code for `ASP_IMAGE.aspx`. The code includes a page directive, a DOCTYPE declaration, a head section with a title, and a body section containing a form with an image and a button.

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

<!DOCTYPE html>

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

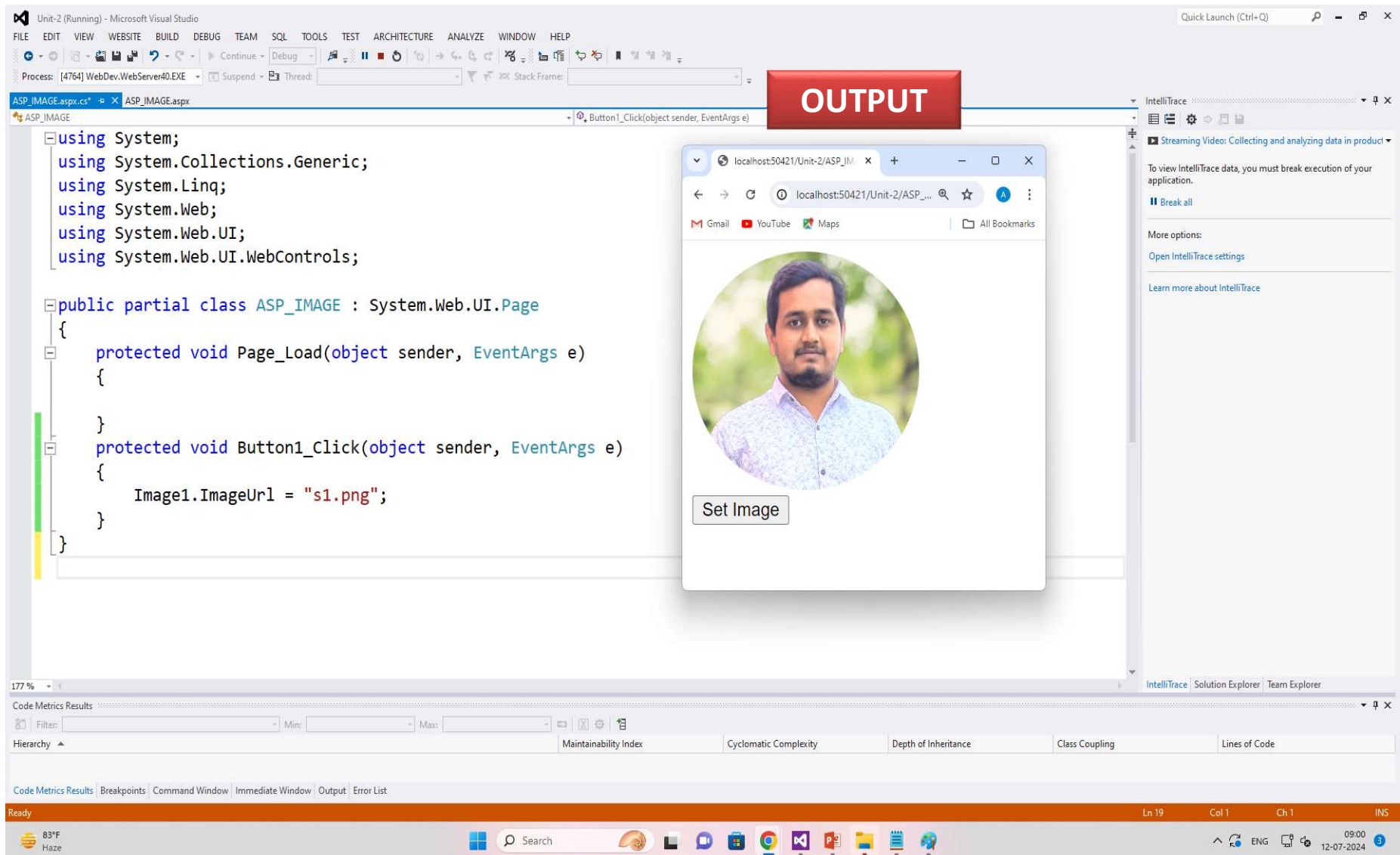
<asp:Image ID="Image1" runat="server" Height="172px" Width="178px" />
<br />
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Set Image" />

</div>
</form>
</body>
</html>
```
- Design View:** Shows a visual representation of the page. It features a large rectangular area representing the image and a button labeled "Set Image" below it.
- Toolbox:** Lists various ASP.NET controls such as Pointer, AdRotator, BulletedList, Button, Calendar, CheckBox, etc.
- Solution Explorer:** Shows the project structure for "Unit-2", including files like `ASP_BUTTON.aspx`, `ASP_FILEUPLOAD.aspx`, `ASP_HIDDENFIELD.aspx`, `ASP_IMAGE.aspx`, `ASP_LABEL.aspx`, `ASP_PLACEHOLDER.aspx`, `ASP_TEXTBOX.aspx`, `ksv-logo.jpg`, `s1.png`, and `web.config`.
- Properties Window:** Shows the properties for the selected control, including Class, Culture, Debug, EnableSessionState, Id, Language (C#), MasterPageFile, Style, StyleSheetTheme, Theme, Title, Trace, TraceMode, and UICulture.



# 📢 ASP.NET Standard Controls - II: Image

**OUTPUT**



The screenshot displays the Visual Studio IDE with the following components:

- Code Editor:** Shows the code for `ASP_IMAGE.aspx.cs`.
 

```

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

public partial class ASP_IMAGE : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Image1.ImageUrl = "s1.png";
    }
}

```
- Browser Output:** A browser window at `localhost:50421/Unit-2/ASP_IMAGE.aspx` shows a circular image of a man and a button labeled "Set Image".
- IntelliTrace:** A panel on the right showing "Streaming Video: Collecting and analyzing data in product..." and a "Break all" button.
- Code Metrics Results:** A panel at the bottom showing various metrics like Maintainability Index, Cyclomatic Complexity, etc.



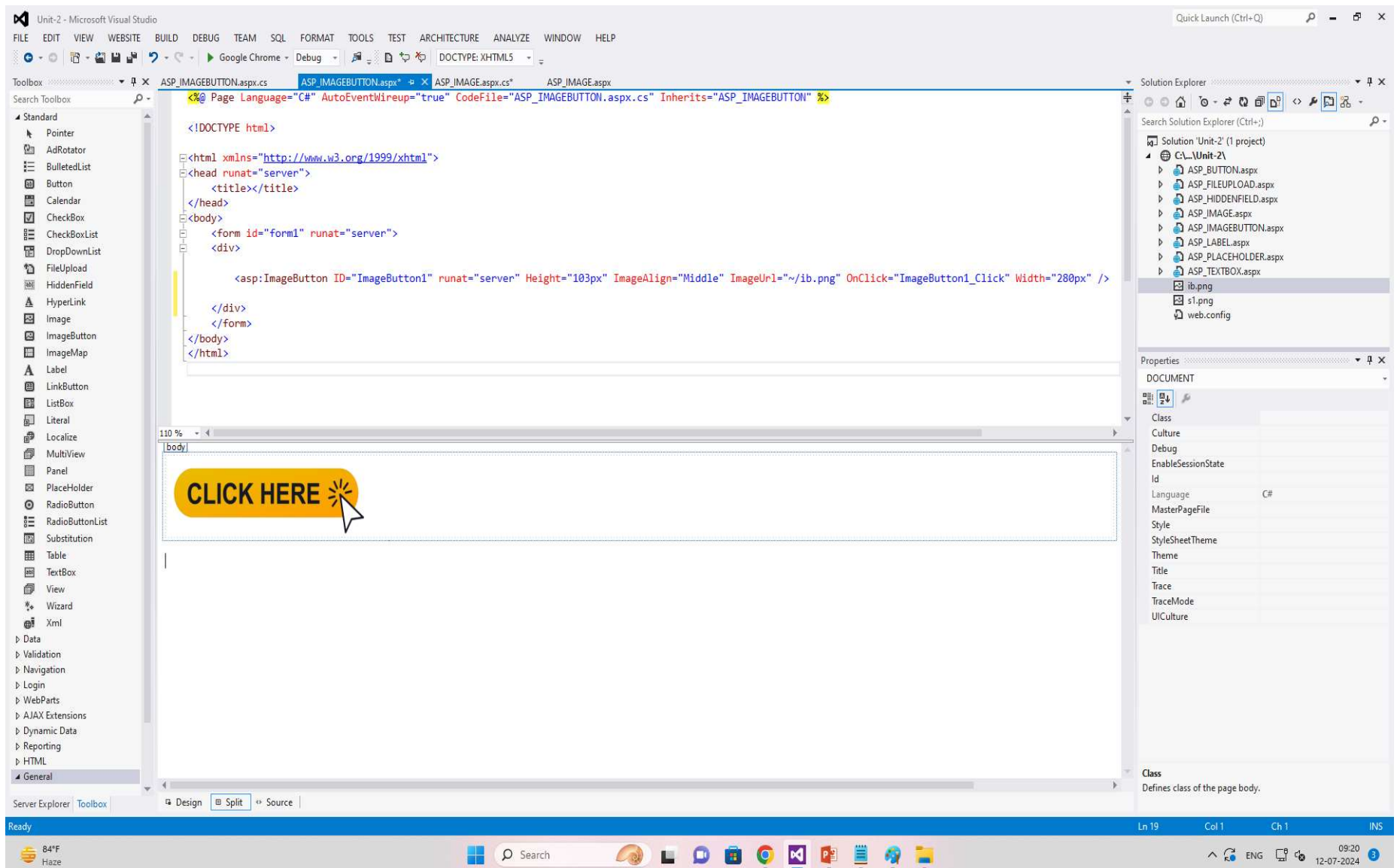
## ASP.NET Standard Controls - II: ImageButton

- ✓ ASP:ImageButton is a Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ This control is used to display image on a button control.

Ex- `<asp:ImageButton ID="ImageButton1" runat="server" />`



# 📢 ASP.NET Standard Controls - II: ImageButton

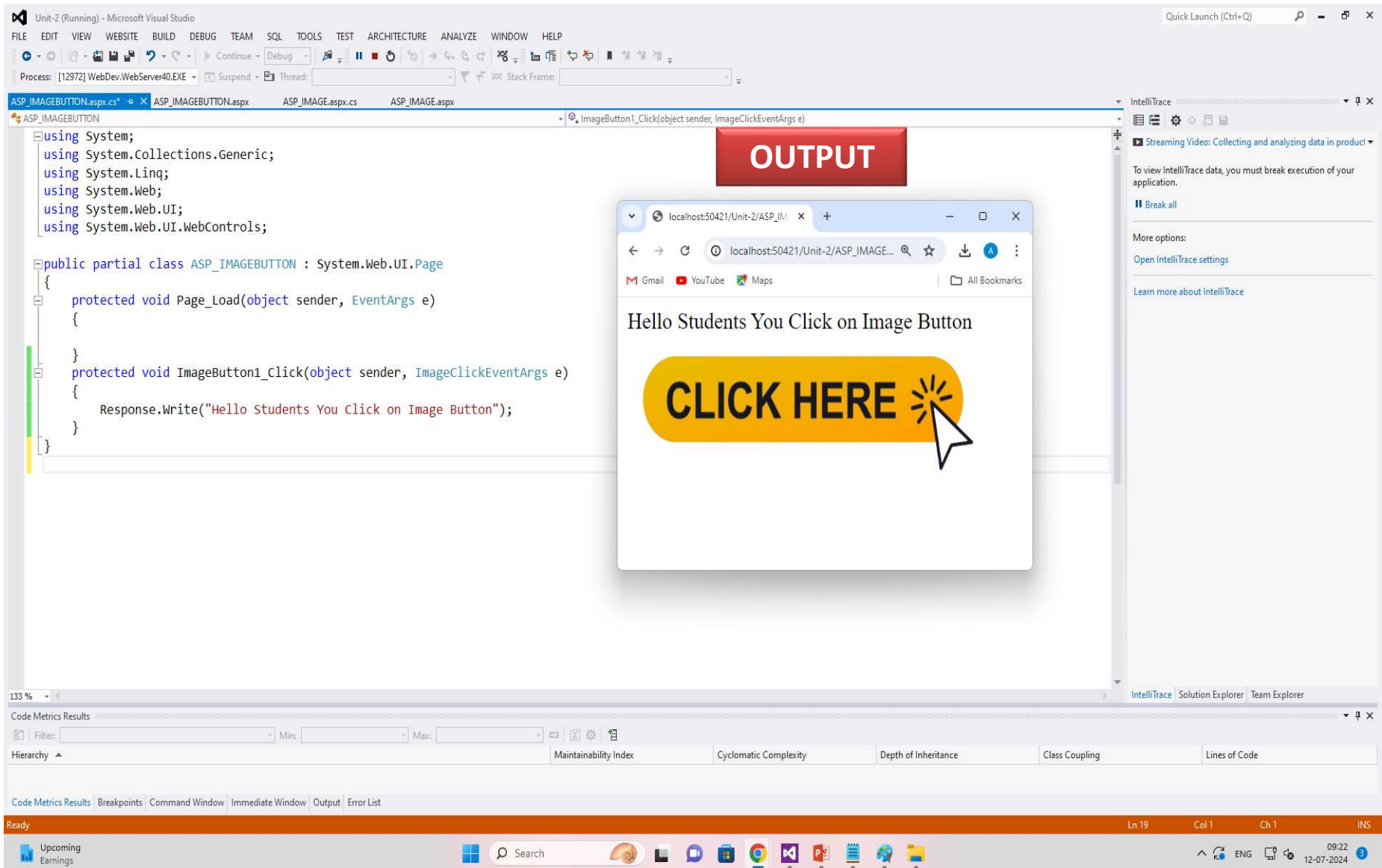


The screenshot displays the Microsoft Visual Studio IDE with the following components:

- Toolbox:** The 'Image' category is expanded, showing the 'ImageButton' control.
- Source View:** Shows the HTML markup for the ImageButton:
 

```
<asp:ImageButton ID="ImageButton1" runat="server" Height="103px" ImageAlign="Middle" ImageUrl="~/ib.png" OnClick="ImageButton1_Click" Width="280px" />
```
- Design View:** Shows a yellow button with the text 'CLICK HERE' and a mouse cursor pointing to it.
- Solution Explorer:** Shows the project structure with files like ASP\_BUTTON.aspx, ASP\_FILEUPLOAD.aspx, ASP\_HIDDENFIELD.aspx, ASP\_IMAGE.aspx, ASP\_IMAGEBUTTON.aspx, ASP\_LABEL.aspx, ASP\_PLACEHOLDER.aspx, and ASP\_TEXTBOX.aspx.
- Properties Window:** Shows the 'DOCUMENT' class properties, including Class, Culture, Debug, EnableSessionState, Id, Language (C#), MasterPageFile, Style, StyleSheetTheme, Theme, Title, Trace, TraceMode, and UI Culture.

# ASP.NET Standard Controls - II: ImageButton




The screenshot displays the Visual Studio IDE with a C# code file for an ASP.NET page. The code includes the following:

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

public partial class ASP_IMAGEBUTTON : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
    {
        Response.Write("Hello Students You Click on Image Button");
    }
}
```

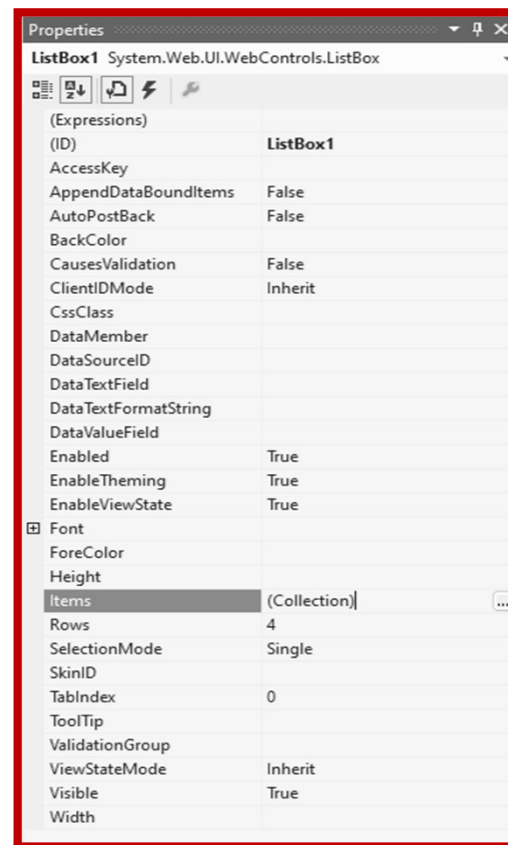
The browser window shows the rendered output:

localhost:50421/Unit-2/ASP\_IM...  
localhost:50421/Unit-2/ASP\_IMAGE...  
Gmail YouTube Maps All Bookmarks  
Hello Students You Click on Image Button  
CLICK HERE 

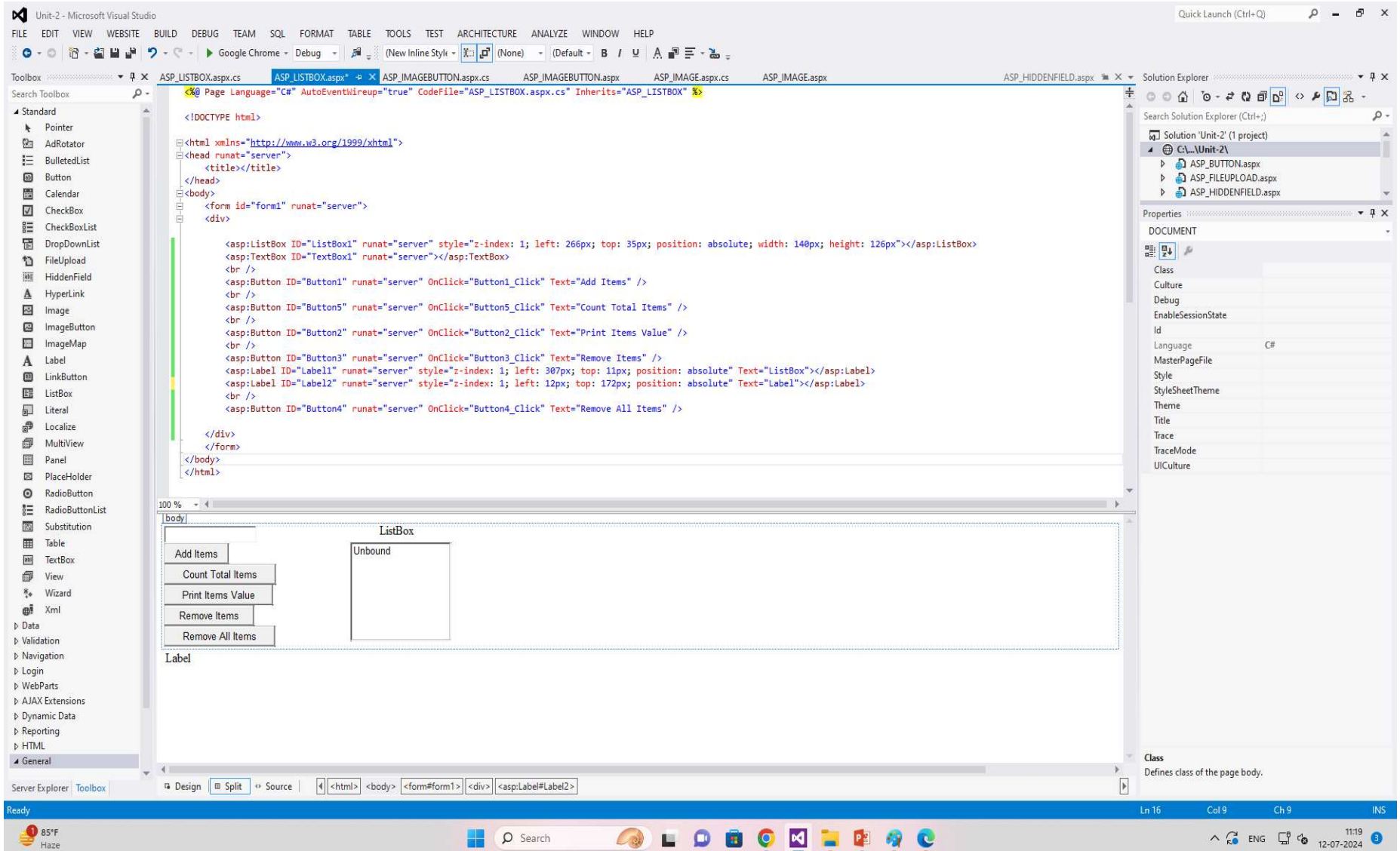
The bottom status bar shows: Ready | Ln 19 | Col 1 | Ch 1 | INS | 09:22 | 12-07-2024

## ASP.NET Standard Controls - II: ListBox

- ✓ ASP:ListBox is a Microsoft Visual Studio Standard Control Tools.
  - ✓ It is always run on Server.
  - ✓ The ListBox Control provides us a user interface that will display the List of the items. From there, the users can select one or more items from the List. We can use the ListBox to show the multiple columns, and these columns can contain images and other controls.
- Ex- `<asp:ListBox ID="ListBox1" runat="server"></asp:ListBox>`



# 📢 ASP.NET Standard Controls - II: ListBox

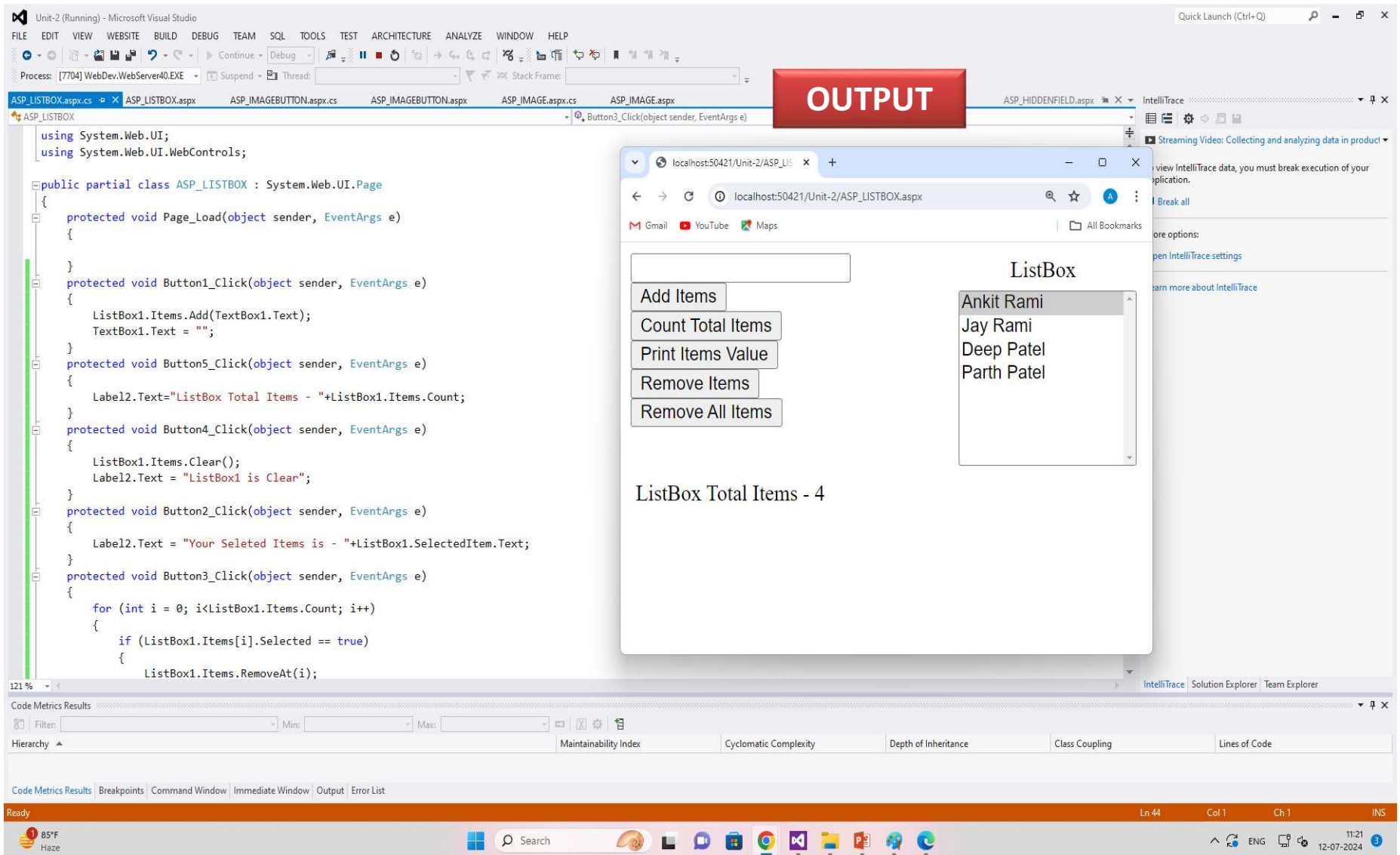


The screenshot displays the Microsoft Visual Studio IDE with the following components:

- Code Editor:** Shows the source code for `ASP_LISTBOX.aspx.cs`. The code includes a form with a `ListBox` control, a `TextBox`, and several buttons for adding, counting, printing, removing, and clearing items. The `ListBox` is configured with `runat="server"` and specific styling attributes.
- Design View:** Shows the rendered output of the code. It features a form with a `ListBox` control (labeled "Unbound"), a `TextBox`, and five buttons: "Add Items", "Count Total Items", "Print Items Value", "Remove Items", and "Remove All Items".
- Toolbox:** Lists various ASP.NET standard controls, with `ListBox` selected.
- Solution Explorer:** Shows the project structure for "Unit-2", including `ASP_BUTTON.aspx`, `ASP_FILEUPLOAD.aspx`, and `ASP_HIDDENFIELD.aspx`.
- Properties Window:** Shows the properties for the selected `ListBox` control, including `Class`, `Culture`, `Debug`, `EnableSessionState`, `Id`, `Language`, `MasterPageFile`, `Style`, `StyleSheetTheme`, `Theme`, `Title`, `Trace`, `TraceMode`, and `UILanguage`.



# ASP.NET Standard Controls - II: ListBox



The screenshot displays the Visual Studio IDE with a C# code file for `ASP_LISTBOX.aspx`. The code implements a web page with a text input, a list box, and several buttons. The browser window shows the rendered page with the following elements:

- A text input field.
- Buttons: "Add Items", "Count Total Items", "Print Items Value", "Remove Items", and "Remove All Items".
- A **ListBox** containing the items: "Ankit Rami", "Jay Rami", "Deep Patel", and "Parth Patel".
- A label displaying "ListBox Total Items - 4".

```
using System.Web.UI;
using System.Web.UI.WebControls;

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

    protected void Button1_Click(object sender, EventArgs e)
    {
        ListBox1.Items.Add(TextBox1.Text);
        TextBox1.Text = "";
    }

    protected void Button5_Click(object sender, EventArgs e)
    {
        Label12.Text="ListBox Total Items - "+ListBox1.Items.Count;
    }

    protected void Button4_Click(object sender, EventArgs e)
    {
        ListBox1.Items.Clear();
        Label12.Text = "ListBox1 is Clear";
    }

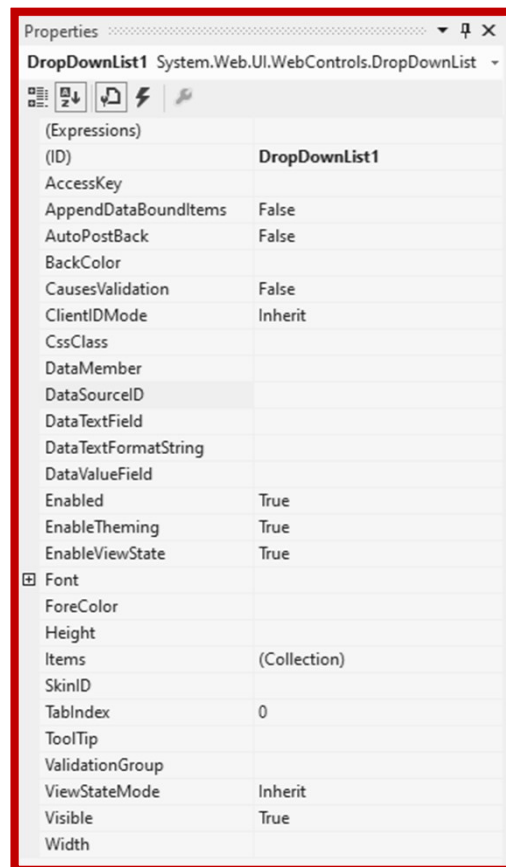
    protected void Button2_Click(object sender, EventArgs e)
    {
        Label12.Text = "Your Seleted Items is - "+ListBox1.SelectedItem.Text;
    }

    protected void Button3_Click(object sender, EventArgs e)
    {
        for (int i = 0; i<ListBox1.Items.Count; i++)
        {
            if (ListBox1.Items[i].Selected == true)
            {
                ListBox1.Items.RemoveAt(i);
            }
        }
    }
}
```

## ASP.NET Standard Controls - II: DropDownList

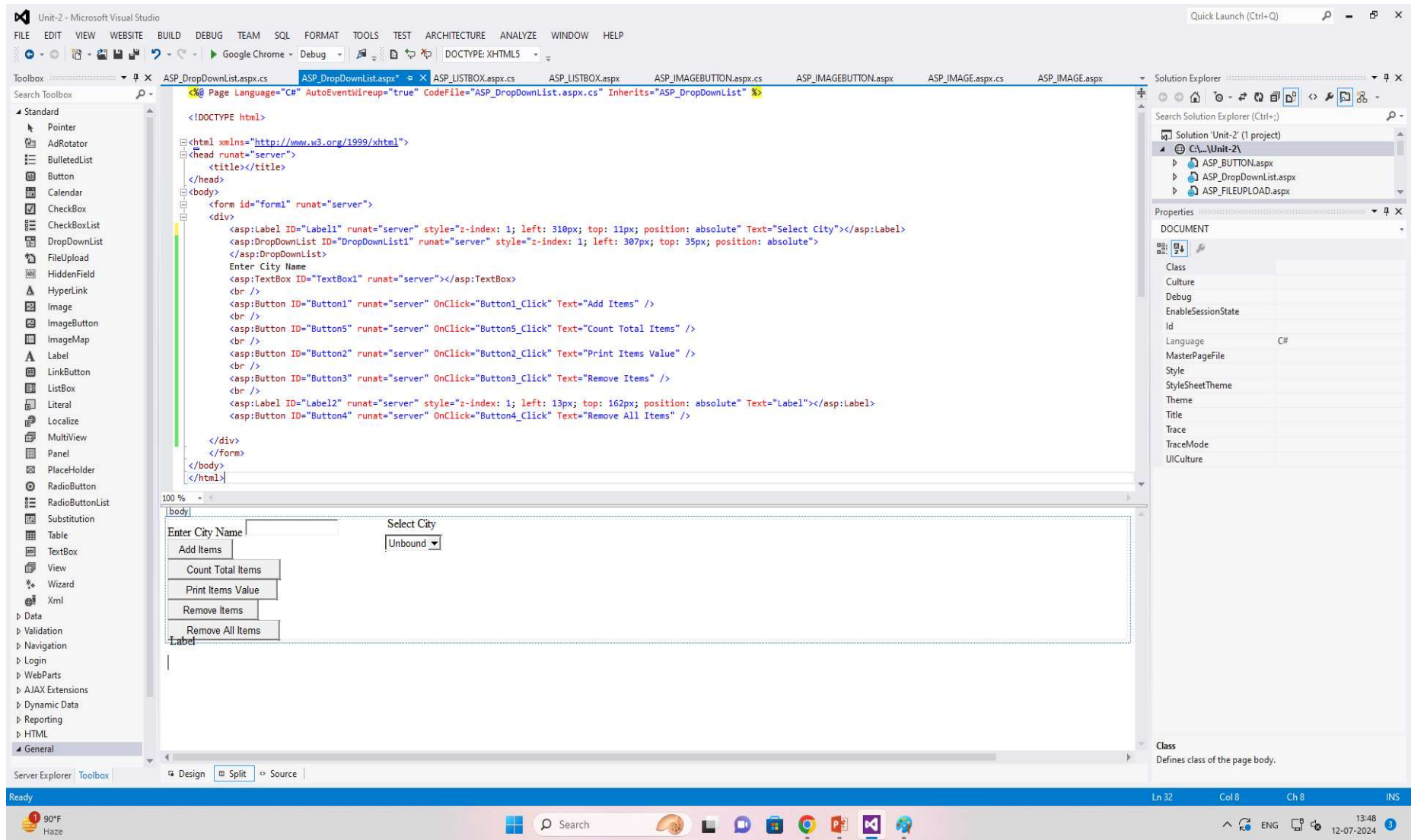
- ✓ ASP:DropDownList is a Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ The DropDownList control is a web server element that creates the drop-down menu and lets users select a single item from various options. You can add any number of items to the DropDownList.

Ex- `<asp:DropDownList ID="DropDownList1" runat="server">`  
`</asp:DropDownList>`





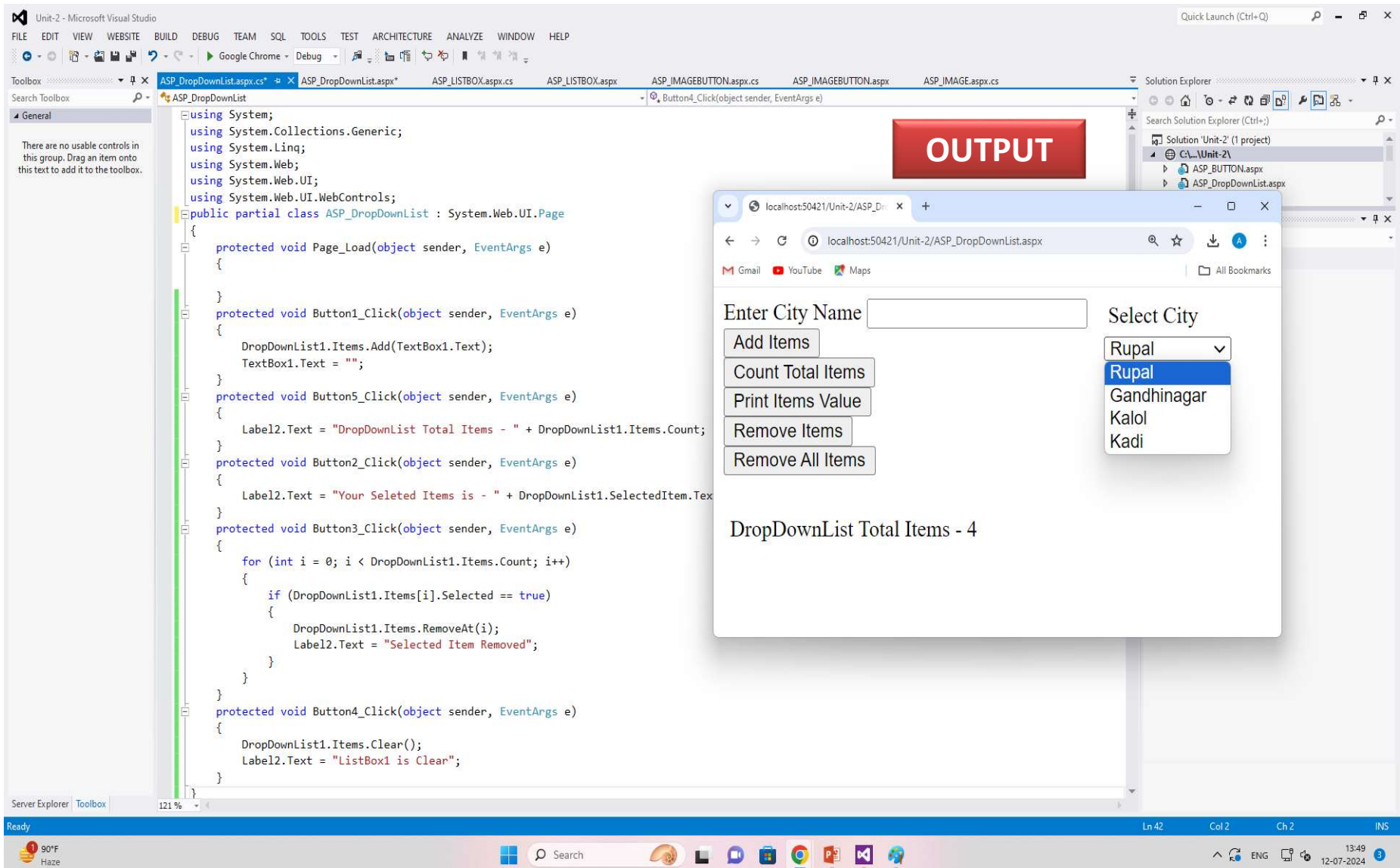
# 📣 ASP.NET Standard Controls - II: DropDownList



The screenshot displays the Microsoft Visual Studio IDE with the following components:

- Code View:** Shows the HTML and ASP.NET code for a page. The code includes a form with a label "Select City", a DropDownList control, a text box for "Enter City Name", and several buttons for "Add Items", "Count Total Items", "Print Items Value", "Remove Items", and "Remove All Items".
- Design View:** Shows a visual representation of the page layout. It includes a text box labeled "Enter City Name", a "Select City" label, a dropdown menu currently showing "Unbound", and a vertical stack of buttons: "Add Items", "Count Total Items", "Print Items Value", "Remove Items", and "Remove All Items".
- Toolbox:** Lists various ASP.NET standard controls such as Pointer, AdRotator, BulletedList, Button, Calendar, CheckBox, DropDownList, FileUpload, HiddenField, HyperLink, Image, ImageButton, ImageMap, Label, LinkButton, ListBox, Literal, Localize, MultiView, Panel, Placeholder, RadioButton, and Substitution.
- Solution Explorer:** Shows the project structure for "Unit-2", including files like ASP\_BUTTON.aspx, ASP\_DropDownList.aspx, and ASP\_FILEUPLOAD.aspx.
- Properties Window:** Shows the "DOCUMENT" class properties, including Class, Culture, Debug, EnableSessionState, Id, Language (C#), MasterPageFile, Style, StyleSheetTheme, Theme, Title, Trace, TraceMode, and UICulture.

# 📢 ASP.NET Standard Controls - II: DropDownList



The screenshot displays the Visual Studio IDE with a C# code file for an ASP.NET page. The code implements a DropDownList control with several event handlers. A red 'OUTPUT' box is overlaid on the code, and a browser window shows the rendered output.

```

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

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

    protected void Button1_Click(object sender, EventArgs e)
    {
        DropDownList1.Items.Add(TextBox1.Text);
        TextBox1.Text = "";
    }

    protected void Button5_Click(object sender, EventArgs e)
    {
        Label2.Text = "DropDownList Total Items - " + DropDownList1.Items.Count;
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        Label2.Text = "Your Selected Items is - " + DropDownList1.SelectedItem.Text;
    }

    protected void Button3_Click(object sender, EventArgs e)
    {
        for (int i = 0; i < DropDownList1.Items.Count; i++)
        {
            if (DropDownList1.Items[i].Selected == true)
            {
                DropDownList1.Items.RemoveAt(i);
                Label2.Text = "Selected Item Removed";
            }
        }
    }

    protected void Button4_Click(object sender, EventArgs e)
    {
        DropDownList1.Items.Clear();
        Label2.Text = "ListBox is Clear";
    }
}
  
```

**OUTPUT**

Enter City Name

Add Items

Count Total Items

Print Items Value

Remove Items

Remove All Items

Select City

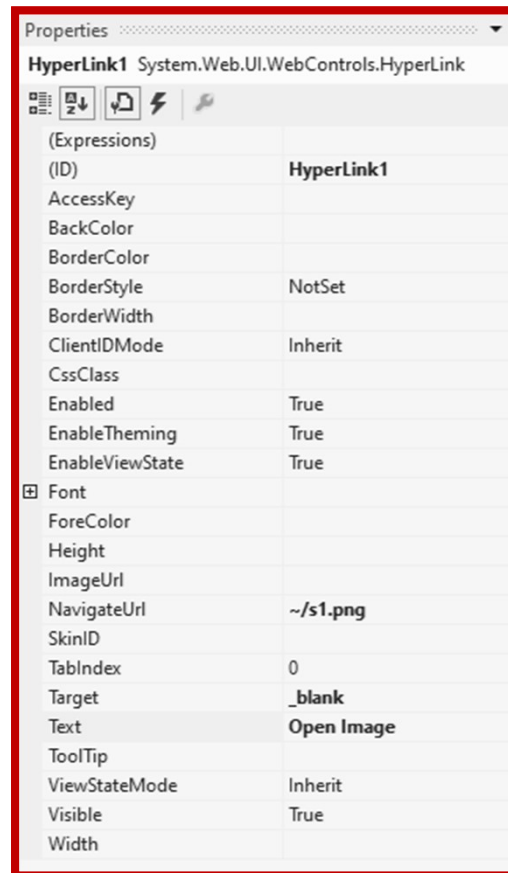
- Rupal
- Rupal
- Gandhinagar
- Kalol
- Kadi

DropDownList Total Items - 4

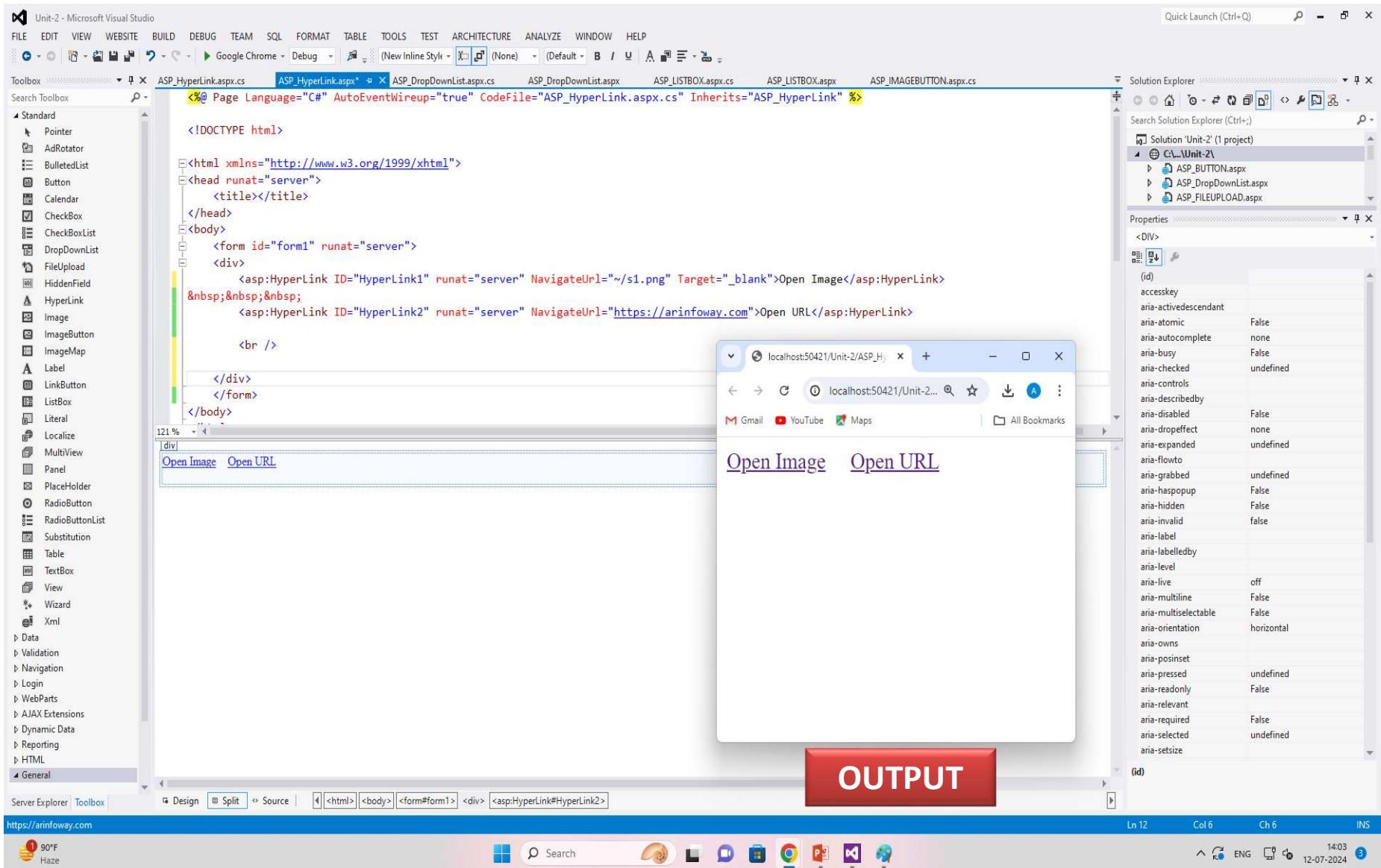
## ASP.NET Standard Controls - II: HyperLink

- ✓ ASP:HyperLink is a Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ It is a control that is used to create a hyperlink. It responds to a click event. We can use it to refer any web page on the server.

Ex- `<asp:HyperLink ID="HyperLink1" runat="server" NavigateUrl="~/s1.png" Target="_blank">Open Image</asp:HyperLink>`



# 📢 ASP.NET Standard Controls - II: HyperLink



The screenshot displays the Visual Studio IDE with the following components:

- Code Editor:** Shows the ASP.NET code for `ASP_HyperLink.aspx.cs`. The code includes a page directive, a DOCTYPE declaration, and a form containing two `<asp:HyperLink>` controls. The first control is configured with `NavigateUrl="~/s1.png"` and `Target="_blank"`. The second control is configured with `NavigateUrl="https://arinfoway.com"`.
- Browser Preview:** A window titled `localhost:50421/Unit-2/ASP_Hy...` shows the rendered output: `Open Image` and `Open URL`, both displayed as underlined blue text.
- Properties Window:** Shows the properties for the selected `<DIV>` control, including attributes like `accesskey`, `aria-activedescendant`, and `aria-autocomplete`.
- Server Explorer:** Shows the project structure for `Unit-2`, including `ASP_BUTTON.aspx`, `ASP_DropDownList.aspx`, and `ASP_FILEUPLOAD.aspx`.

**OUTPUT**

## ASP.NET Standard Controls - II: CheckBox

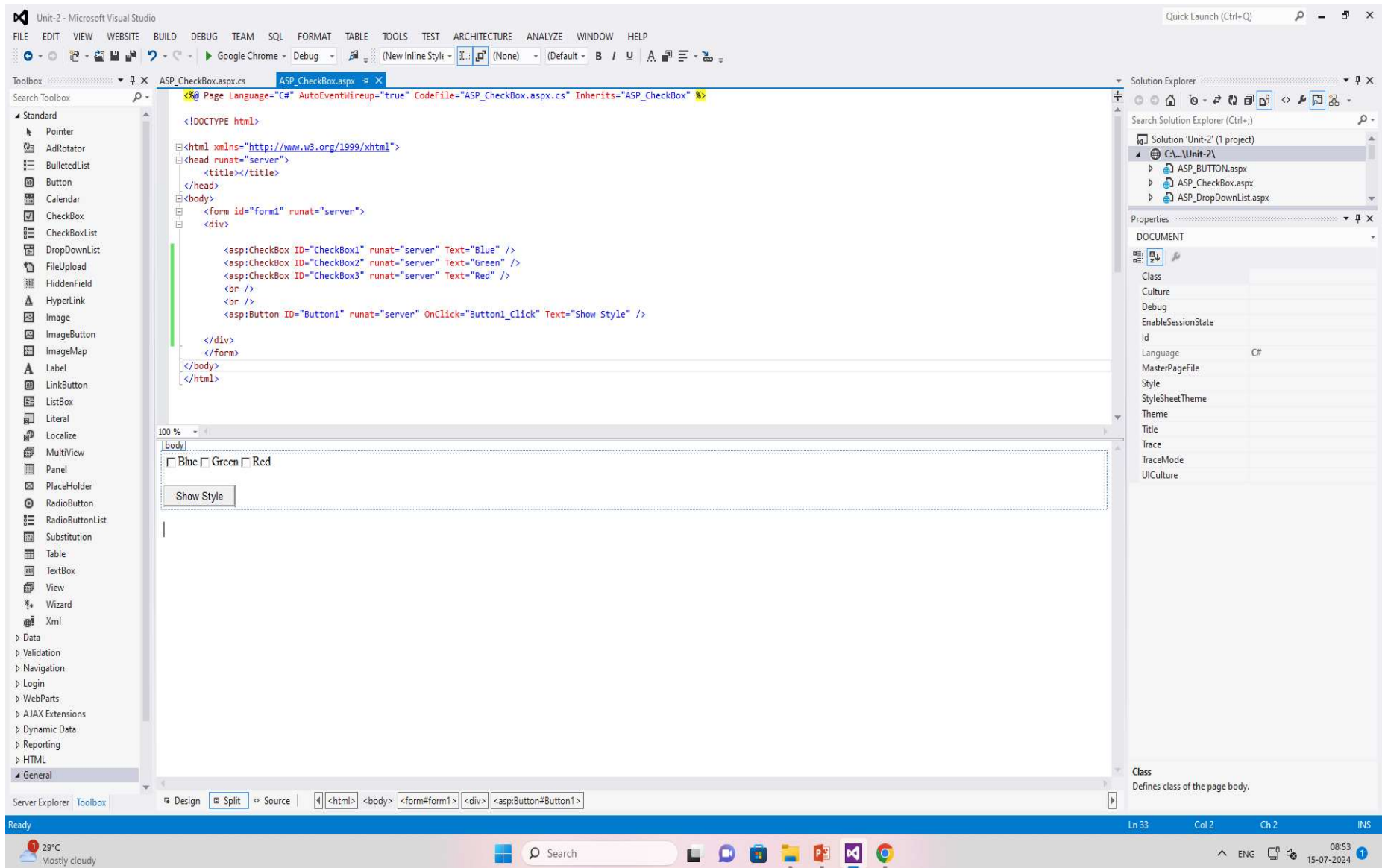
- ✓ ASP:CheckBox is a Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ It is used to get multiple inputs from the user. It allows user to select choices from the set of choices.
- ✓ It takes user input in yes or no format. It is useful when we want multiple choices from the user.

Ex- `<asp:CheckBox ID="CheckBox1" runat="server" />`





# 📢 ASP.NET Standard Controls - II: CheckBox



The screenshot displays the Microsoft Visual Studio environment for an ASP.NET project named 'Unit-2'. The main window shows the code for 'ASP\_CheckBox.aspx.cs', which includes the following code:

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

<asp:CheckBox ID="CheckBox1" runat="server" Text="Blue" />
<asp:CheckBox ID="CheckBox2" runat="server" Text="Green" />
<asp:CheckBox ID="CheckBox3" runat="server" Text="Red" />
<br />
<br />
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Show Style" />

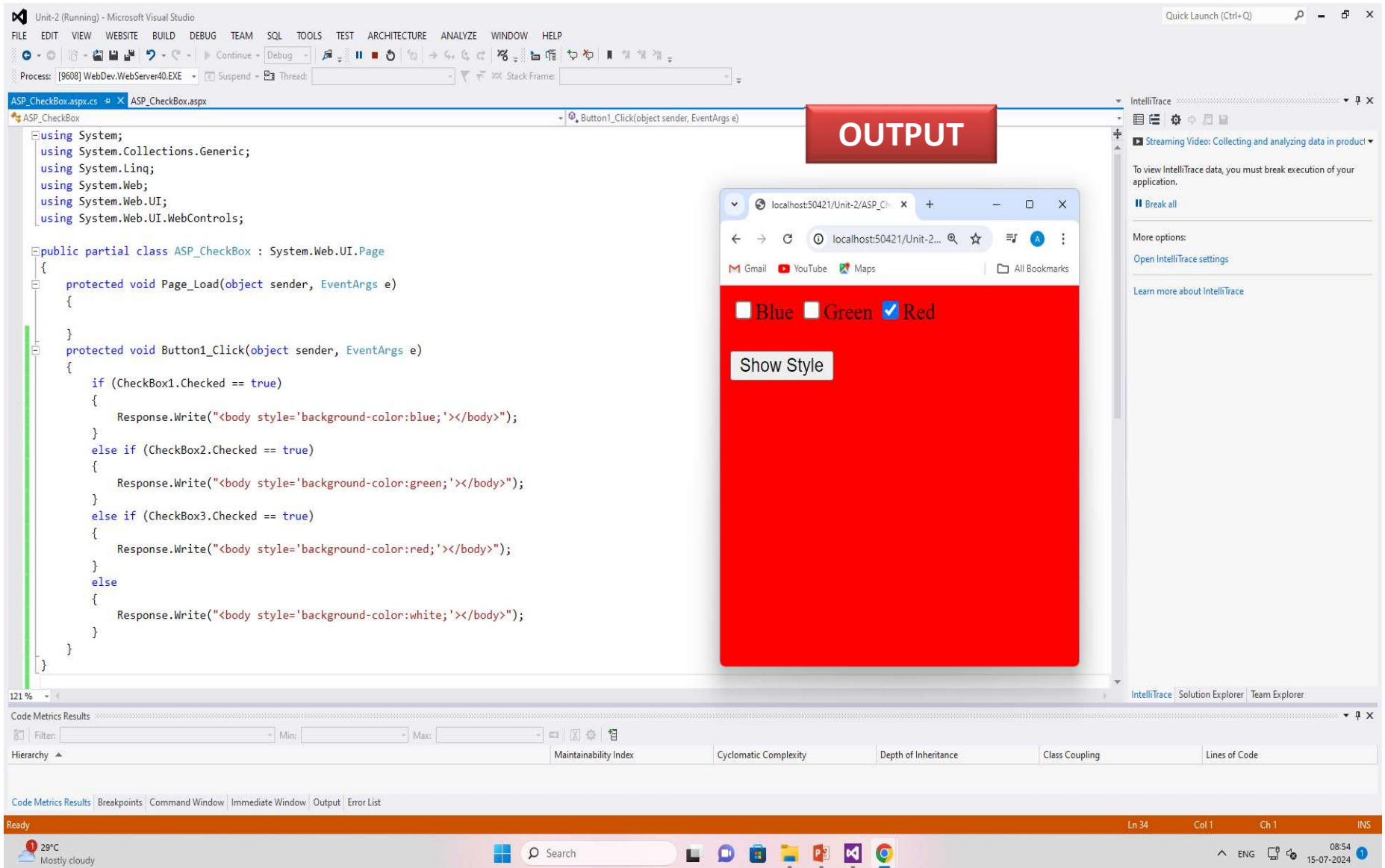
</div>
</form>
</body>
</html>
```

The preview window shows the rendered output of the code, which is a form with three checkboxes labeled 'Blue', 'Green', and 'Red', and a 'Show Style' button below them.

The Properties window on the right shows the 'DOCUMENT' properties for the page, including Class, Culture, Debug, EnableSessionState, Id, Language (C#), MasterPageFile, Style, StyleSheetTheme, Theme, Title, Trace, TraceMode, and UICulture.



# ASP.NET Standard Controls - II: CheckBox



The screenshot displays the Visual Studio IDE with a C# code file named `ASP_CheckBox.aspx.cs` and a browser window showing the rendered output. A red box labeled "OUTPUT" is overlaid on the browser window.

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

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

    protected void Button1_Click(object sender, EventArgs e)
    {
        if (CheckBox1.Checked == true)
        {
            Response.Write("<body style='background-color:blue;'></body>");
        }
        else if (CheckBox2.Checked == true)
        {
            Response.Write("<body style='background-color:green;'></body>");
        }
        else if (CheckBox3.Checked == true)
        {
            Response.Write("<body style='background-color:red;'></body>");
        }
        else
        {
            Response.Write("<body style='background-color:white;'></body>");
        }
    }
}
```

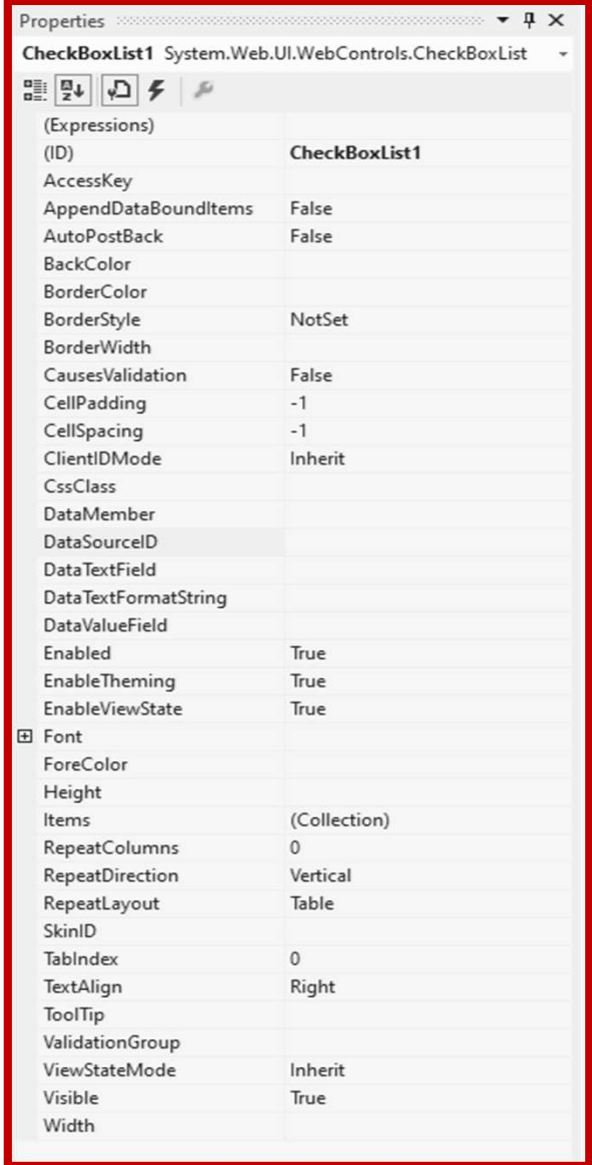
The browser window shows the rendered output of the code, which is a red background. The output includes three checkboxes:  Blue,  Green, and  Red. Below the checkboxes is a button labeled "Show Style".

## ASP.NET Standard Controls - II: CheckBoxList

- ✓ ASP:CheckBoxList is a Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ It is a web control that can be used to collate the items that can be checked, thus giving the user the ability to select multiple items simultaneously.
- ✓ This list of items in the CheckBoxList can be dynamically generated using the Data Binding functions.

Ex-

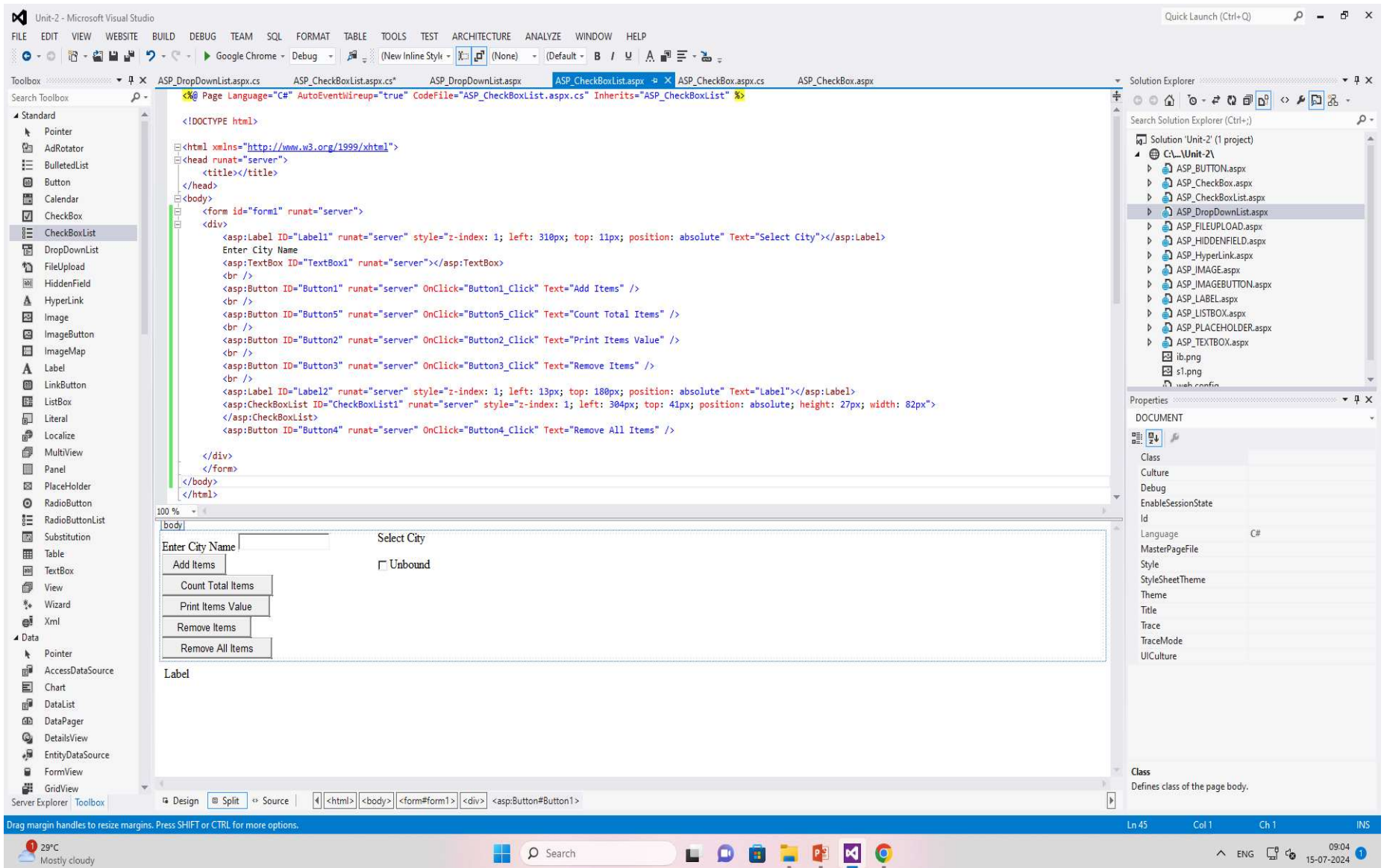
```
<asp:CheckBoxList ID="CheckBoxList1" runat="server">
</asp:CheckBoxList>
```



The screenshot shows the Properties window for a **CheckBoxList1** control. The control is identified as `System.Web.UI.WebControls.CheckBoxList`. The properties are listed in a table below:

(Expressions)	
(ID)	CheckBoxList1
AccessKey	
AppendDataBoundItems	False
AutoPostBack	False
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
CausesValidation	False
CellPadding	-1
CellSpacing	-1
ClientIDMode	Inherit
CssClass	
DataMember	
DataSourceID	
DataTextField	
DataTextFormatString	
DataValueField	
Enabled	True
EnableTheming	True
EnableViewState	True
<b>Font</b>	
ForeColor	
Height	
Items	(Collection)
RepeatColumns	0
RepeatDirection	Vertical
RepeatLayout	Table
SkinID	
TabIndex	0
TextAlign	Right
ToolTip	
ValidationGroup	
ViewStateMode	Inherit
Visible	True
Width	

# 📢 ASP.NET Standard Controls - II: CheckBoxList



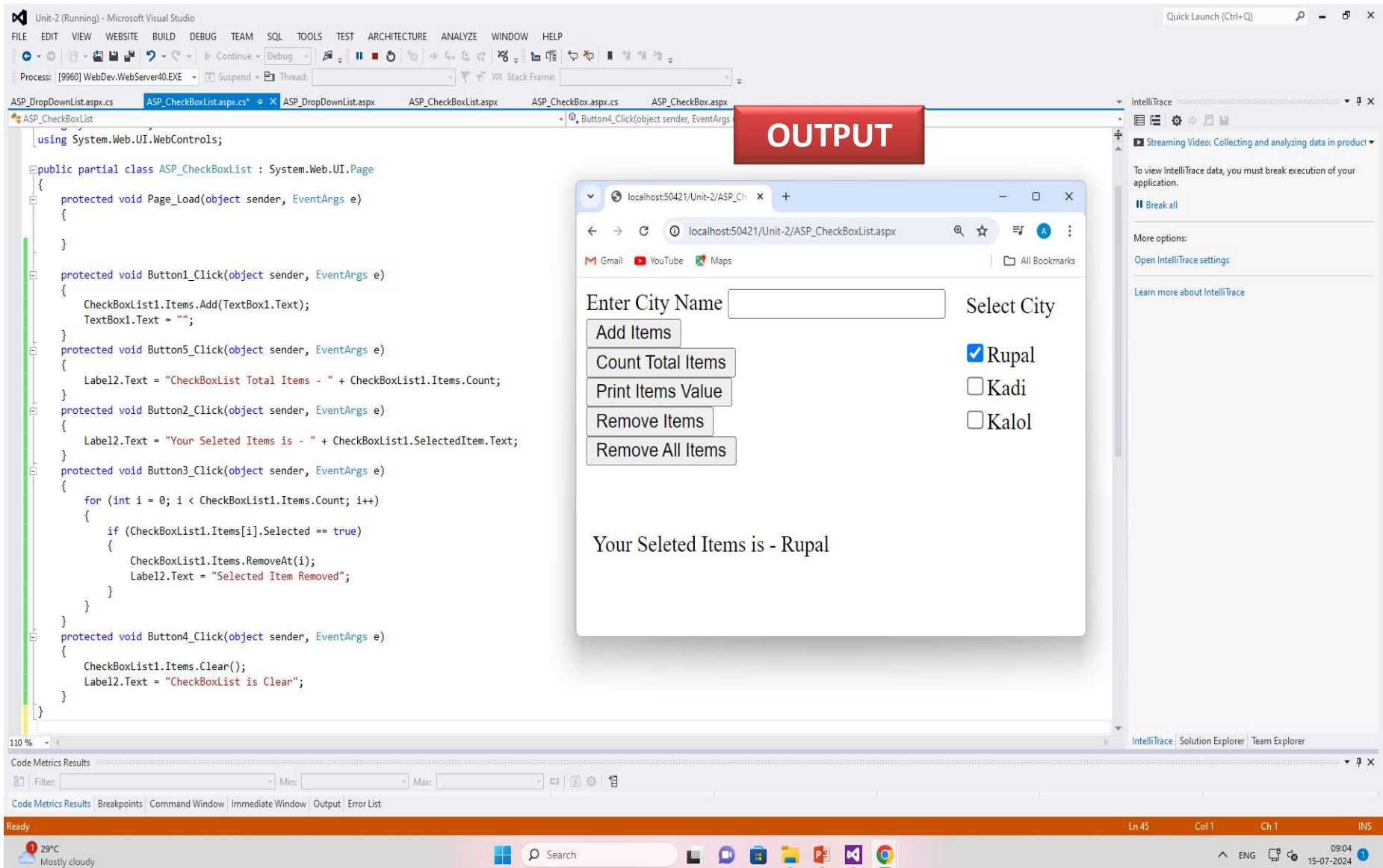
The screenshot displays the Microsoft Visual Studio environment for a project named 'Unit-2'. The main window is split into three panes:

- Toolbox:** Shows various ASP.NET Standard Controls. The 'CheckBoxList' control is highlighted.
- Code View:** Displays the ASPX markup for 'ASP\_CheckBoxList.aspx'. The code includes a form with a label 'Select City', a text box for 'Enter City Name', and several buttons: 'Add Items', 'Count Total Items', 'Print Items Value', 'Remove Items', and 'Remove All Items'. A 'CheckBoxList' control is also present, with a 'Label' property set to 'Label'.
- Design View:** Shows the rendered web page. It features a text box labeled 'Enter City Name', a 'Select City' label, and a 'Unbound' checkbox. Below these are five buttons: 'Add Items', 'Count Total Items', 'Print Items Value', 'Remove Items', and 'Remove All Items'. A 'Label' control is also visible.

The bottom status bar shows the current line and column numbers: Ln 45, Col 1, Ch 1, INS.

# 📢 ASP.NET Standard Controls - II: CheckBoxList

OUTPUT



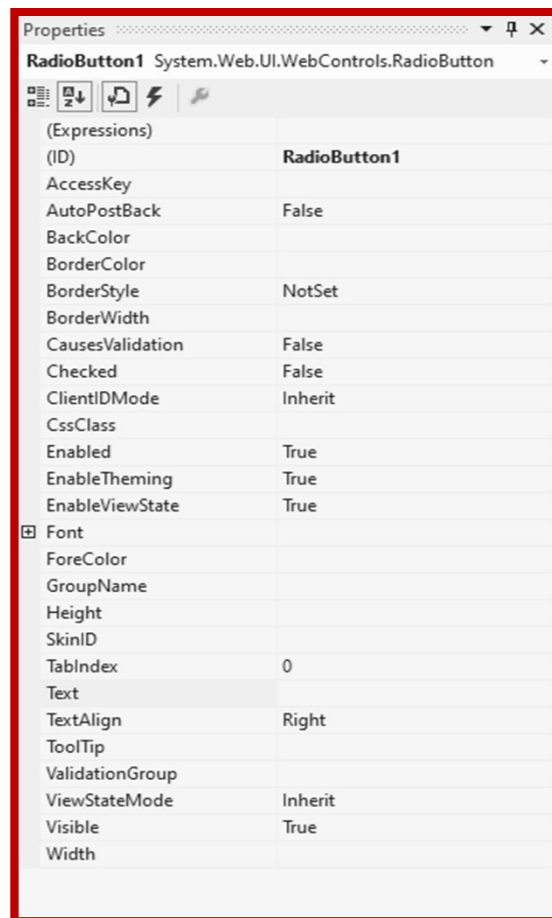
The screenshot displays the Visual Studio IDE with the following components:

- Code Editor:** Shows the code for `ASP_CheckBoxList` class, including `Page_Load`, `Button1_Click`, `Button5_Click`, `Button2_Click`, `Button3_Click`, and `Button4_Click` methods.
- Browser Output:** A browser window at `localhost:50421/Unit-2/ASP_CheckBoxList.aspx` shows a form with:
  - Input field: "Enter City Name" with a text box.
  - Buttons: "Add Items", "Count Total Items", "Print Items Value", "Remove Items", "Remove All Items".
  - CheckBoxList: "Select City" with options:  Rupal,  Kadi,  Kalol.
  - Output: "Your Selected Items is - Rupal".
- IntelliTrace:** Shows a message: "To view IntelliTrace data, you must break execution of your application." with a "Break all" button.
- Code Metrics Results:** Filtered view at the bottom of the IDE.

## ASP.NET Standard Controls - II: RadioButton

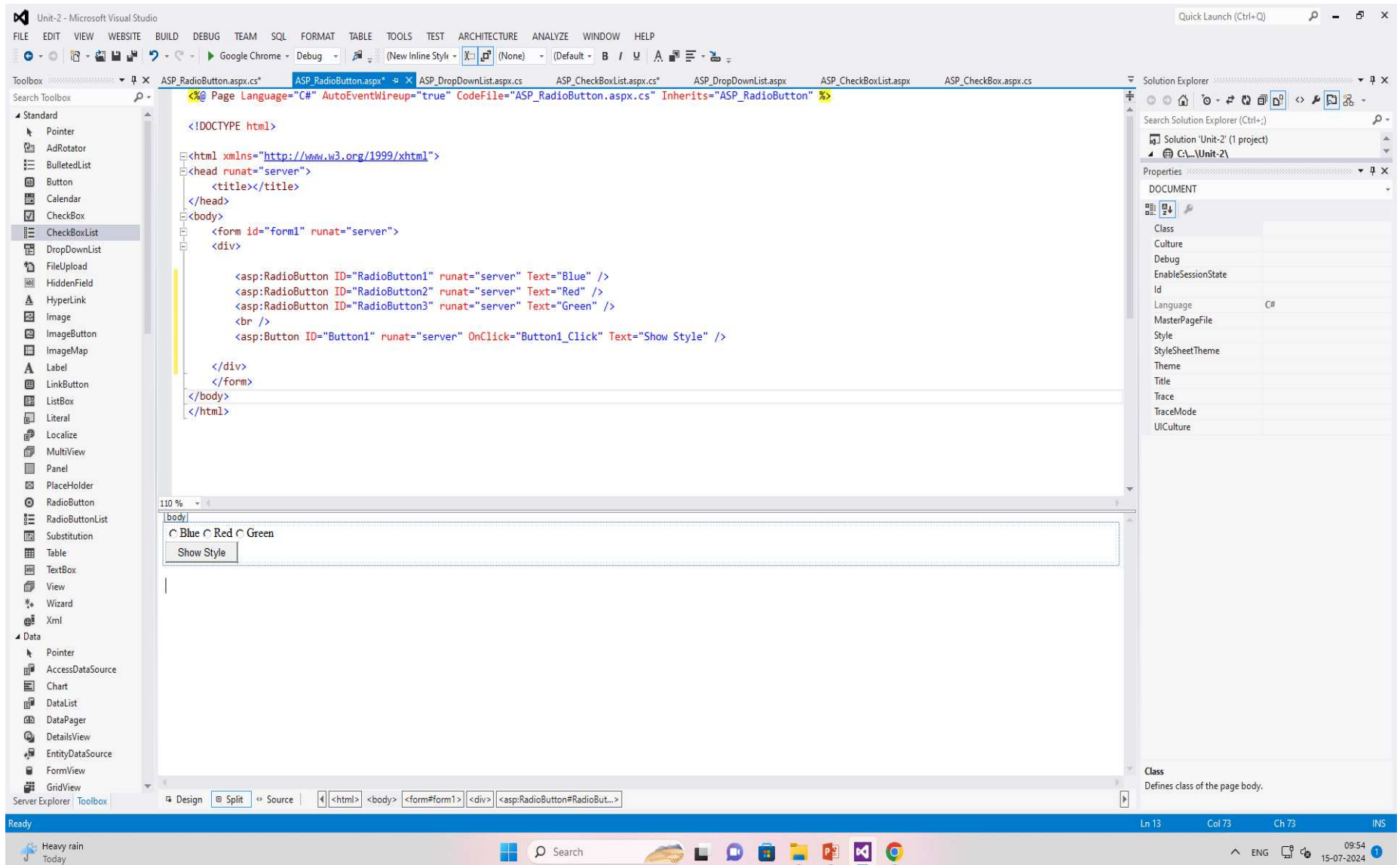
- ✓ ASP:RadioButton is a Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ It is an input control which is used to takes input from the user. It allows user to select a choice from the group of choices.

Ex- `<asp:RadioButton ID="RadioButton1" runat="server" />`





# 📢 ASP.NET Standard Controls - II: RadioButton



The screenshot displays the Visual Studio IDE with the following components:

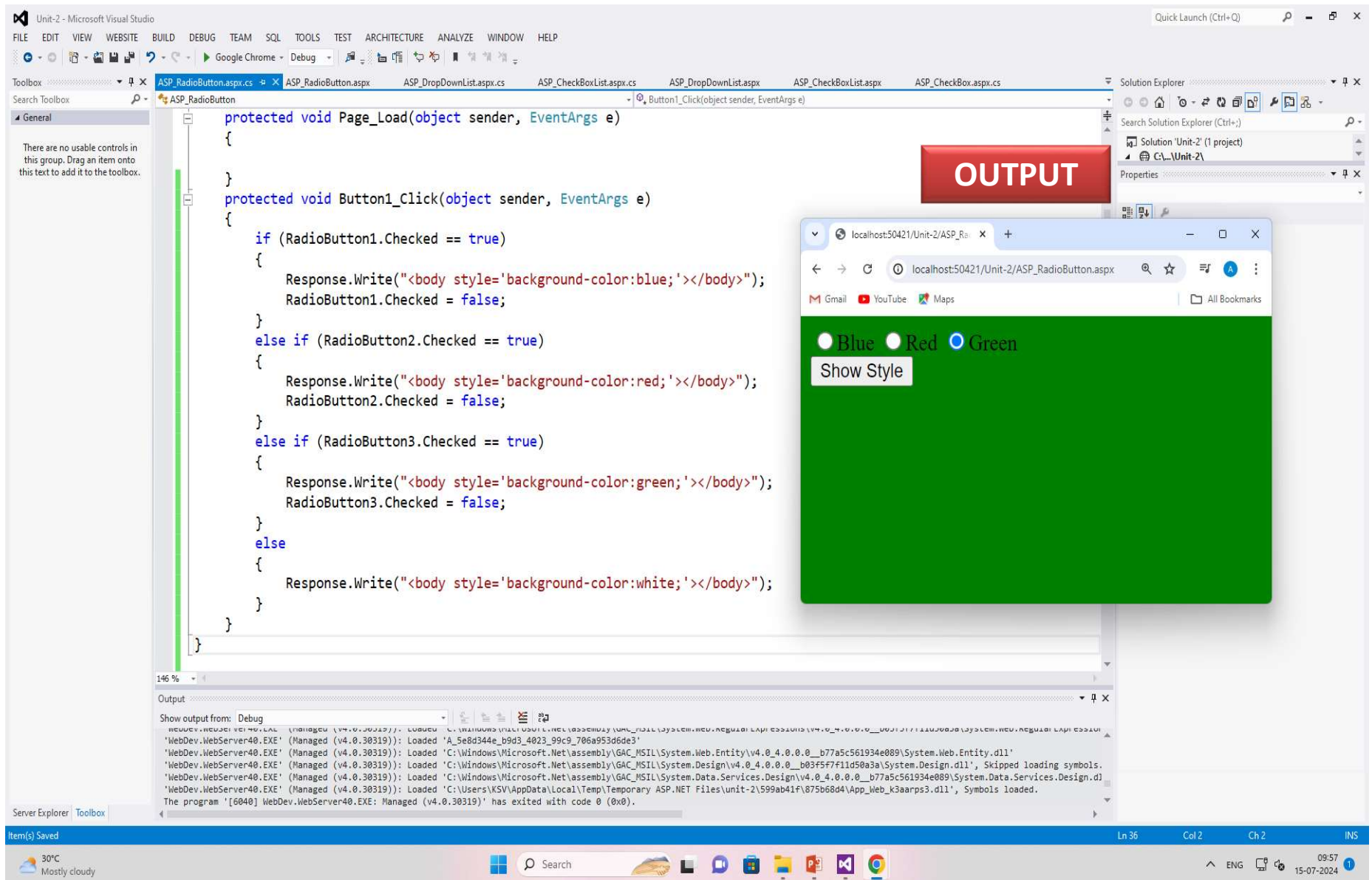
- Code Editor:** Shows the source code for `ASP_RadioButton.aspx`. The code includes a form with three radio buttons and a button:
 

```

      <!DOCTYPE html>
      <html xmlns="http://www.w3.org/1999/xhtml">
      <head runat="server">
      <title></title>
      </head>
      <body>
      <form id="form1" runat="server">
      <div>
      <asp:RadioButton ID="RadioButton1" runat="server" Text="Blue" />
      <asp:RadioButton ID="RadioButton2" runat="server" Text="Red" />
      <asp:RadioButton ID="RadioButton3" runat="server" Text="Green" />
      <br />
      <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Show Style" />
      </div>
      </form>
      </body>
      </html>
      
```
- Design View:** Shows the rendered output of the code, displaying three radio buttons labeled "Blue", "Red", and "Green", and a "Show Style" button below them.
- Toolbox:** Lists various ASP.NET standard controls, with "RadioButton" selected.
- Properties Window:** Shows the properties for the selected control, including Class, Culture, Debug, EnableSessionState, Id, Language (C#), MasterPageFile, Style, StyleSheetTheme, Theme, Title, Trace, TraceMode, and UICulture.



# 📢 ASP.NET Standard Controls - II: RadioButton



The screenshot displays the Visual Studio IDE with a C# code file named `ASP_RadioButton.aspx.cs`. The code implements a `Page_Load` method and a `Button1_Click` event handler. The event handler uses conditional logic to check which radio button is selected and writes the corresponding background color to the response. A red box labeled "OUTPUT" points to the browser window showing the rendered page.

```

protected void Page_Load(object sender, EventArgs e)
{
}

protected void Button1_Click(object sender, EventArgs e)
{
    if (RadioButton1.Checked == true)
    {
        Response.Write("<body style='background-color:blue;'></body>");
        RadioButton1.Checked = false;
    }
    else if (RadioButton2.Checked == true)
    {
        Response.Write("<body style='background-color:red;'></body>");
        RadioButton2.Checked = false;
    }
    else if (RadioButton3.Checked == true)
    {
        Response.Write("<body style='background-color:green;'></body>");
        RadioButton3.Checked = false;
    }
    else
    {
        Response.Write("<body style='background-color:white;'></body>");
    }
}
  
```

The browser window shows the output of the application, which is a green background. The page contains three radio buttons labeled "Blue", "Red", and "Green", with the "Green" button selected. Below the radio buttons is a button labeled "Show Style".

The Output window at the bottom shows the debug output, including the following message:

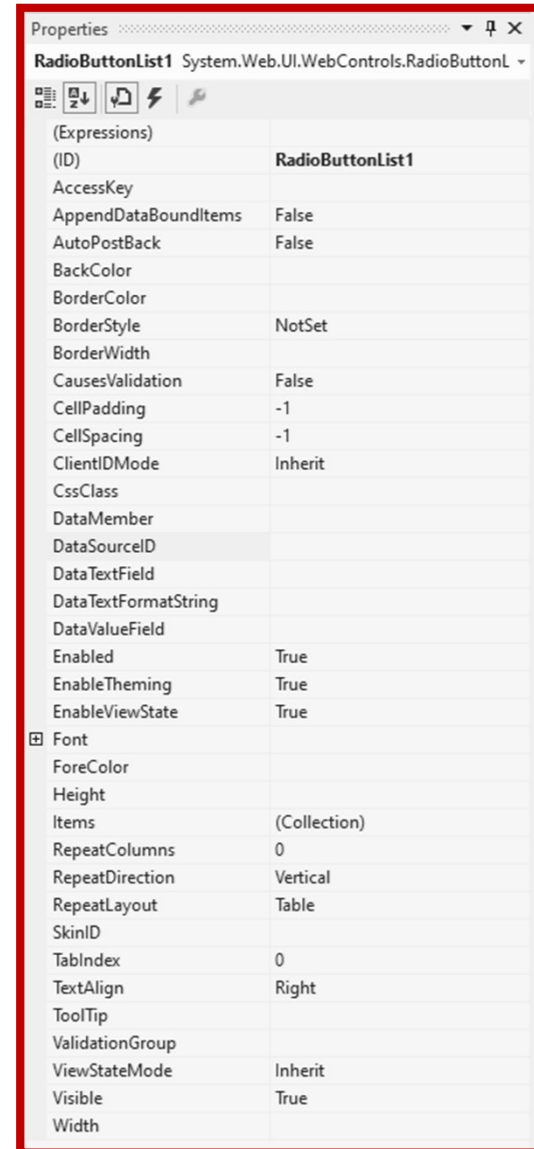
```

The program '[6040] WebDev.WebServer40.EXE: Managed (v4.0.30319)' has exited with code 0 (0x0).
  
```

## ASP.NET Standard Controls - II: RadioButtonList

- ✓ ASP:RadioButtonList is a Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ This immensely helps programmers by providing a single selection radio button group.
- ✓ This group can also be generated dynamically with the help of data binding.

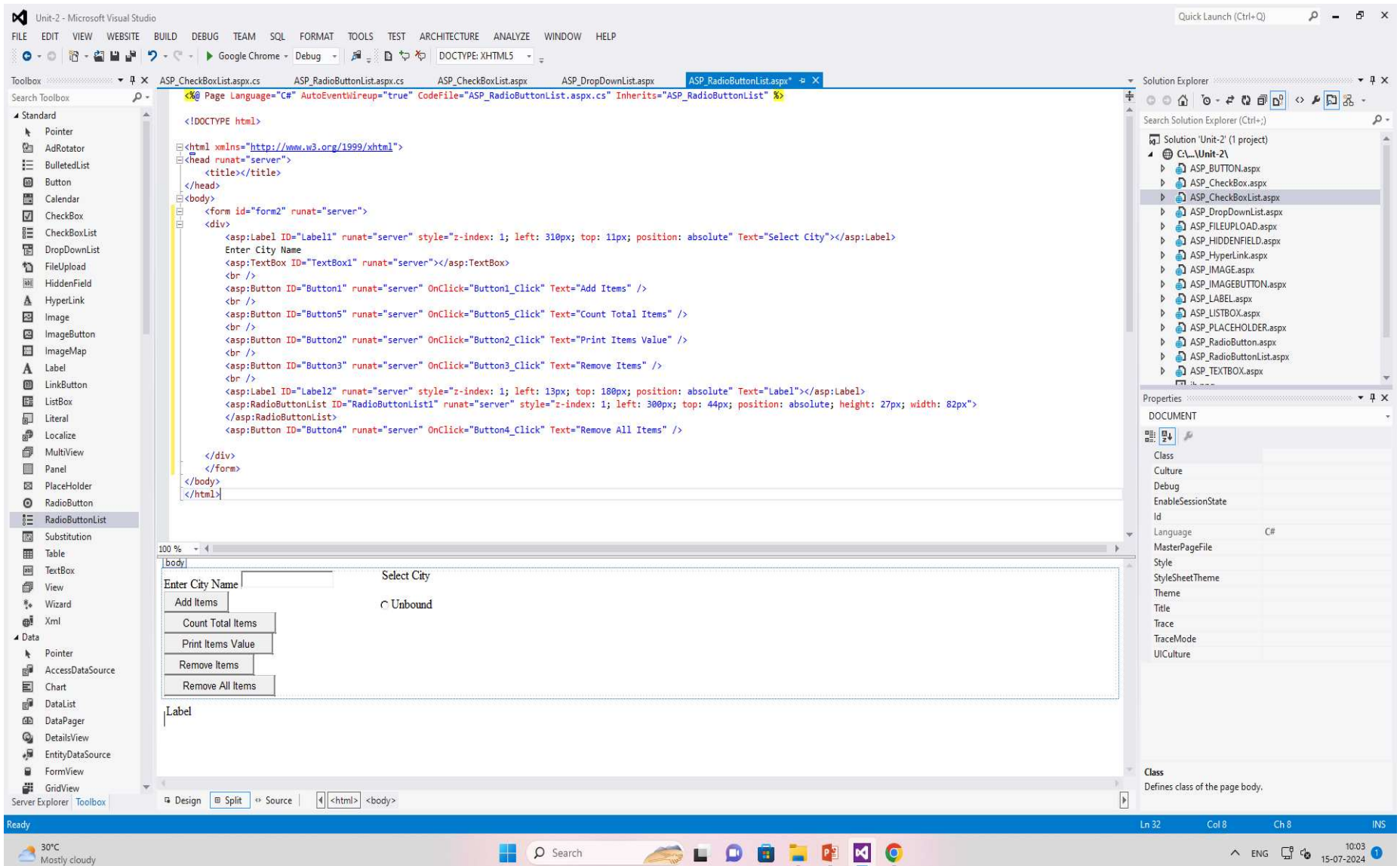
Ex- `<asp:RadioButtonList ID="RadioButtonList1" runat="server"> </asp:RadioButtonList>`



The screenshot shows the Properties window for a `RadioButtonList1` control. The control is of type `System.Web.UI.WebControls.RadioButtonList`. The properties are listed in a table below.

Properties	
RadioButtonsList1 System.Web.UI.WebControls.RadioButtonList	
(Expressions)	
(ID)	RadioButtonsList1
AccessKey	
AppendDataBoundItems	False
AutoPostBack	False
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
CausesValidation	False
CellPadding	-1
CellSpacing	-1
ClientIDMode	Inherit
CssClass	
DataMember	
DataSourceID	
DataTextField	
DataTextFormatString	
DataValueField	
Enabled	True
EnableTheming	True
EnableViewState	True
Font	
ForeColor	
Height	
Items	(Collection)
RepeatColumns	0
RepeatDirection	Vertical
RepeatLayout	Table
SkinID	
TabIndex	0
TextAlign	Right
ToolTip	
ValidationGroup	
ViewStateMode	Inherit
Visible	True
Width	

# 📢 ASP.NET Standard Controls - II: RadioButtonList

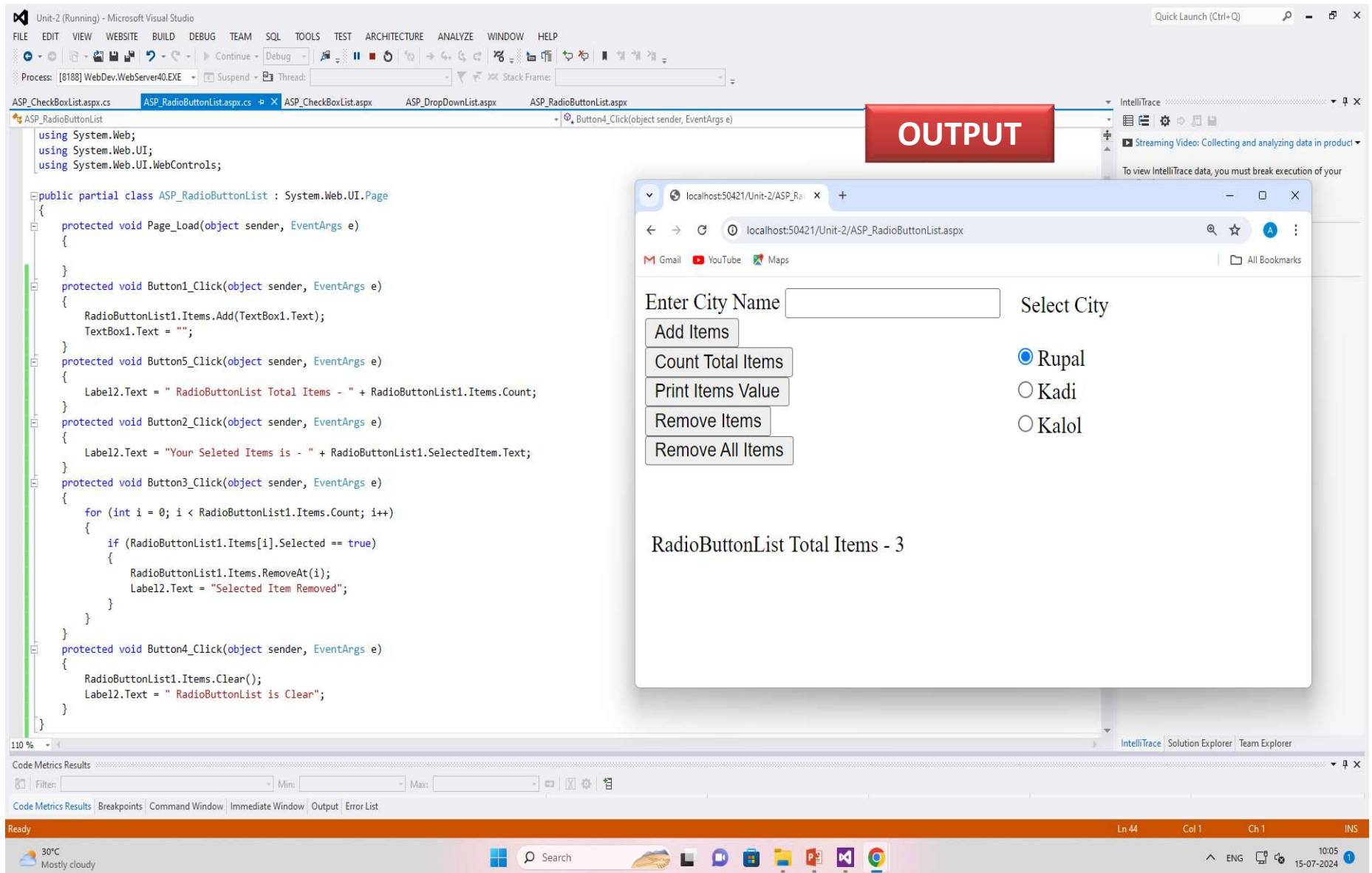


The screenshot displays the Visual Studio IDE with the following components:

- Code Editor:** Shows the source code for `ASP_RadioButtonList.aspx`. The code includes a form with a text box for "Enter City Name", a "Select City" label, and a `RadioButtonList` control. It also contains several buttons for "Add Items", "Count Total Items", "Print Items Value", "Remove Items", and "Remove All Items".
- Toolbox:** The `RadioButtonList` control is selected in the Standard toolbox.
- Design View:** Shows the rendered output of the code, including the text box, label, radio button list, and buttons.
- Properties Window:** Shows the properties for the selected `RadioButtonList` control, such as `Class`, `Culture`, `Language`, and `UICulture`.

# 📢 ASP.NET Standard Controls - II: RadioButtonList

OUTPUT



The screenshot displays the Visual Studio IDE with a C# code file for `ASP_RadioButtonList.aspx.cs` and a browser window showing the rendered web page.

```

using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class ASP_RadioButtonList : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        RadioButtonList1.Items.Add(TextBox1.Text);
        TextBox1.Text = "";
    }
    protected void Button5_Click(object sender, EventArgs e)
    {
        Label2.Text = "RadioButtonList Total Items - " + RadioButtonList1.Items.Count;
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        Label2.Text = "Your Selected Items is - " + RadioButtonList1.SelectedItem.Text;
    }
    protected void Button3_Click(object sender, EventArgs e)
    {
        for (int i = 0; i < RadioButtonList1.Items.Count; i++)
        {
            if (RadioButtonList1.Items[i].Selected == true)
            {
                RadioButtonList1.Items.RemoveAt(i);
                Label2.Text = "Selected Item Removed";
            }
        }
    }
    protected void Button4_Click(object sender, EventArgs e)
    {
        RadioButtonList1.Items.Clear();
        Label2.Text = "RadioButtonList is Clear";
    }
}
  
```

The browser window shows the following rendered page:

Enter City Name  Select City

Add Items

Count Total Items

Print Items Value

Remove Items

Remove All Items

Rupal

Kadi

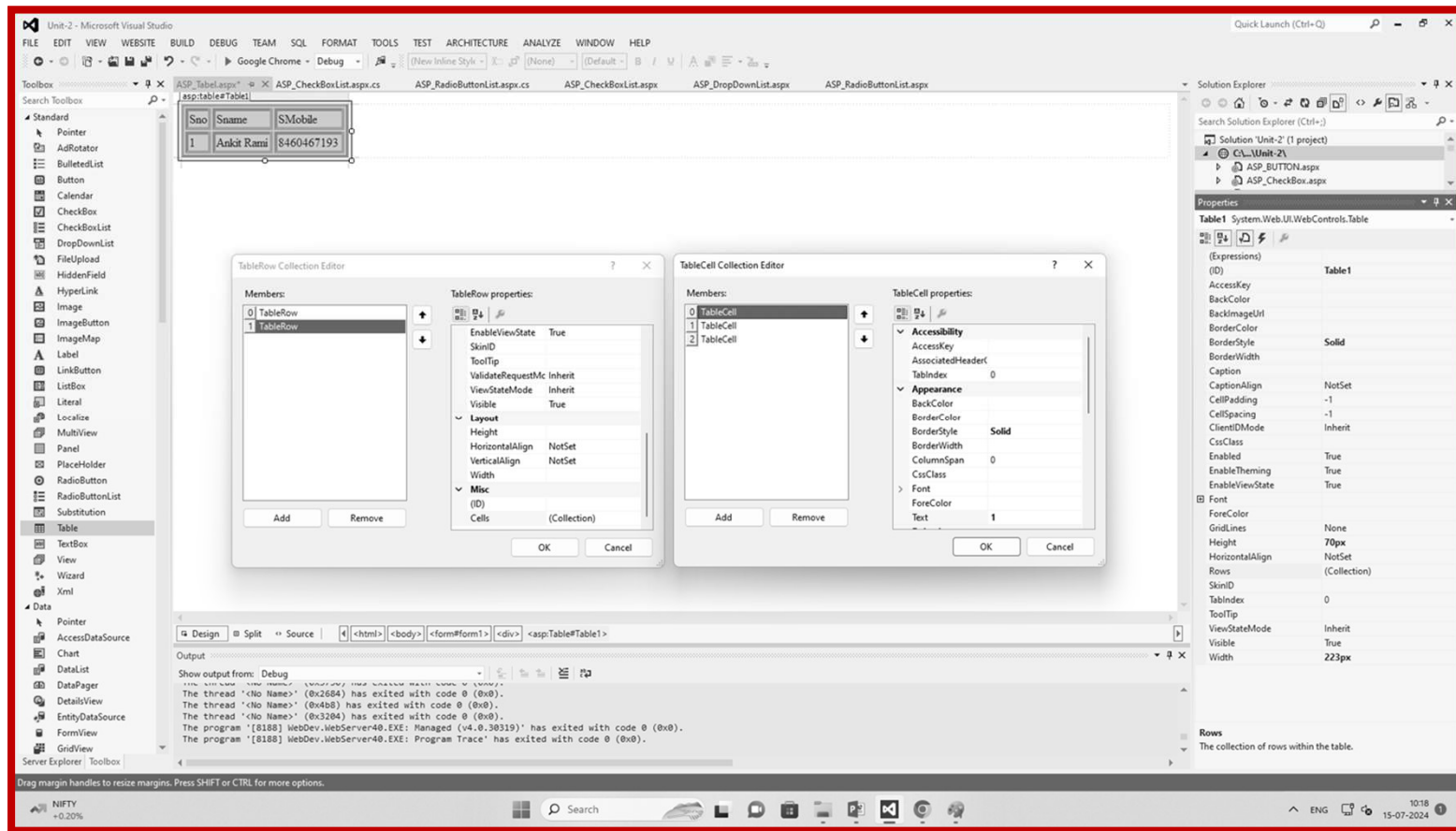
Kalol

RadioButtonList Total Items - 3



# 📢 ASP.NET Standard Controls - II: Table

- ✓ ASP:Table is a Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ Table control is used to structure a web pages. In other words to divide a page into several rows and columns to arrange the information or images. When it is rendered on the page, it is implemented through <table> HTML tag.
- ✓ asp:Table control is we can dynamically add rows or columns at the runtime or change the appearance of the table. Ex- <asp:Table ID="Table2" runat="server"></asp:Table>



The screenshot shows the Visual Studio IDE with a web page containing a table. The table has the following data:

Sno	Sname	SMobile
1	Ankit Rami	8460467193

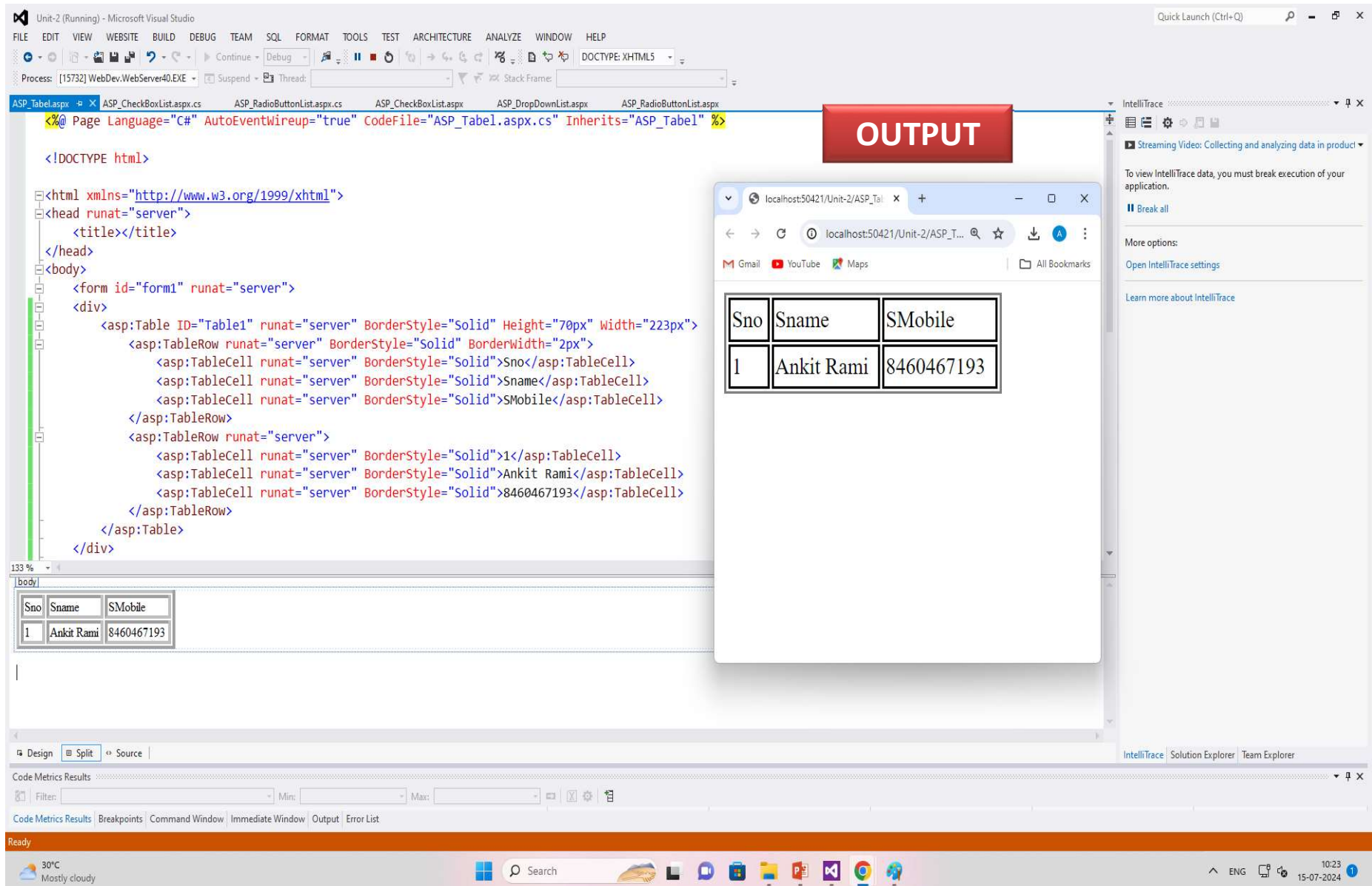
Two configuration windows are open:

- TableRow Collection Editor:** Shows a list of members with one 'TableRow' selected. Properties include: EnableViewState (True), SkinID, ToolTip, ValidateRequestMc (Inherit), ViewStateMode (Inherit), Visible (True), and a 'Cells' collection.
- TableCell Collection Editor:** Shows a list of members with three 'TableCell' items. Properties include: Accessibility, AccessKey, AssociatedHeader, TabIndex (0), and Appearance (BorderStyle: Solid, BorderWidth, ColumnSpan: 0, CssClass, Font: ForeColor, Text: 1).

The Properties window on the right shows the 'Table1' properties, including: ID (Table1), AccessKey, BackColor, BackImageLit, BorderColor, BorderStyle (Solid), BorderWidth, Caption, CaptionAlign (NotSet), CellPadding (-1), CellSpacing (-1), ClientIDMode (Inherit), CssClass, Enabled (True), EnableTheming (True), and EnableViewState (True).

# 📢 ASP.NET Standard Controls - II: Table

OUTPUT



The screenshot displays the Visual Studio IDE with the following components:

- Code Editor:** Shows the source code for `ASP_Table.aspx`. The code defines an ASP.NET table control with the following structure:
 

```

      <!DOCTYPE html>
      <html xmlns="http://www.w3.org/1999/xhtml">
      <head runat="server">
      <title></title>
      </head>
      <body>
      <form id="form1" runat="server">
      <div>
      <asp:Table ID="Table1" runat="server" BorderStyle="Solid" Height="70px" Width="223px">
      <asp:TableRow runat="server" BorderStyle="Solid" BorderWidth="2px">
      <asp:TableCell runat="server" BorderStyle="Solid">Sno</asp:TableCell>
      <asp:TableCell runat="server" BorderStyle="Solid">Sname</asp:TableCell>
      <asp:TableCell runat="server" BorderStyle="Solid">SMobile</asp:TableCell>
      </asp:TableRow>
      <asp:TableRow runat="server">
      <asp:TableCell runat="server" BorderStyle="Solid">1</asp:TableCell>
      <asp:TableCell runat="server" BorderStyle="Solid">Ankit Rami</asp:TableCell>
      <asp:TableCell runat="server" BorderStyle="Solid">8460467193</asp:TableCell>
      </asp:TableRow>
      </asp:Table>
      </div>
      </form>
      </body>
      </html>
      
```
- Browser Window:** Shows the rendered output of the table at `localhost:50421/Unit-2/ASP_T...`. The table has a solid border and contains the following data:
 

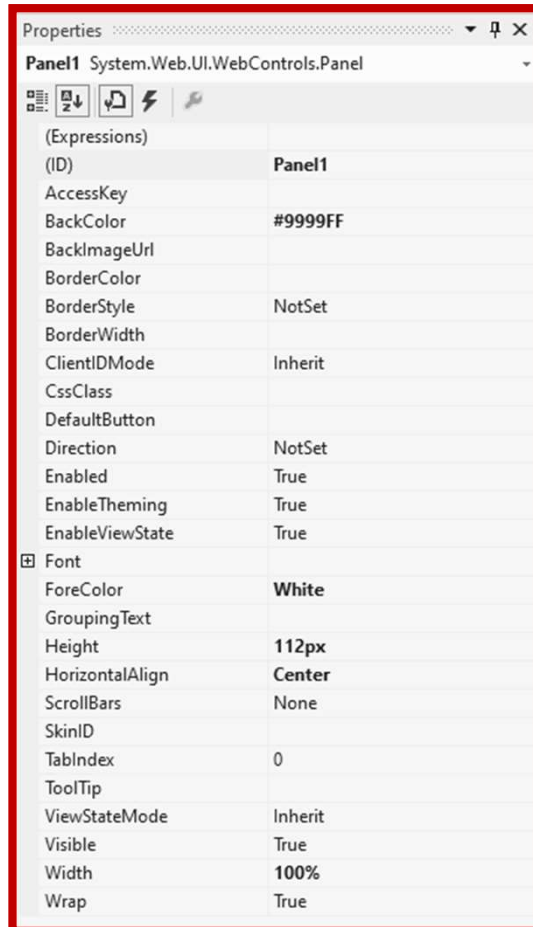
Sno	Sname	SMobile
1	Ankit Rami	8460467193
- Design View:** Located at the bottom left, it shows a visual representation of the table structure.
- IntelliTrace:** A sidebar on the right showing options to view data and break execution.
- Code Metrics Results:** A panel at the bottom of the IDE.



## ASP.NET Standard Controls - II: Panel

- ✓ ASP:Panel is a Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ The Panel control is used as a container for other controls. This control is often used to generate controls by code and to display or hide groups of controls as per the condition.

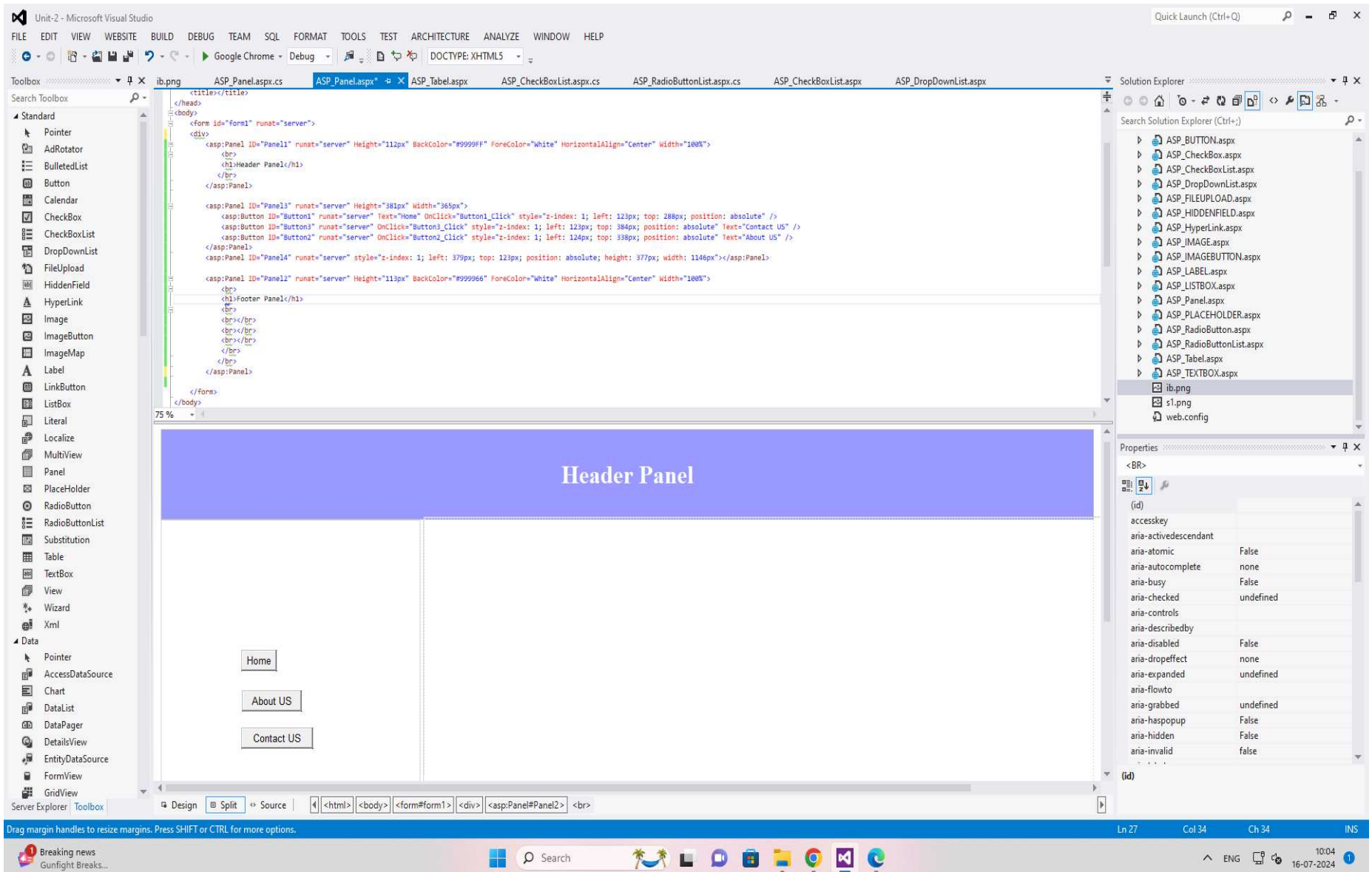
Ex- `<asp:Panel ID="Panel1" runat="server"></asp:Panel>`



The screenshot shows the Properties window for a Panel control. The control is identified as 'Panel1' of type 'System.Web.UI.WebControls.Panel'. The properties are listed in a table format, with some grouped under expandable headers like '(Expressions)', 'Font', and 'Font'.

(Expressions)	
(ID)	Panel1
AccessKey	
BackColor	#9999FF
BackImageUrl	
BorderColor	
BorderStyle	NotSet
BorderWidth	
ClientIDMode	Inherit
CssClass	
DefaultButton	
Direction	NotSet
Enabled	True
EnableTheming	True
EnableViewState	True
Font	
ForeColor	White
GroupingText	
Height	112px
HorizontalAlign	Center
ScrollBars	None
SkinID	
TabIndex	0
ToolTip	
ViewStateMode	Inherit
Visible	True
Width	100%
Wrap	True

# 📢 ASP.NET Standard Controls - II: Panel



The screenshot displays the Microsoft Visual Studio IDE with the following components:

- Code Editor:** Shows the ASP.NET code for `ASP_Panel.aspx`. The code defines a form with a header panel, a main content area with three buttons (Home, About US, Contact US), and a footer panel.

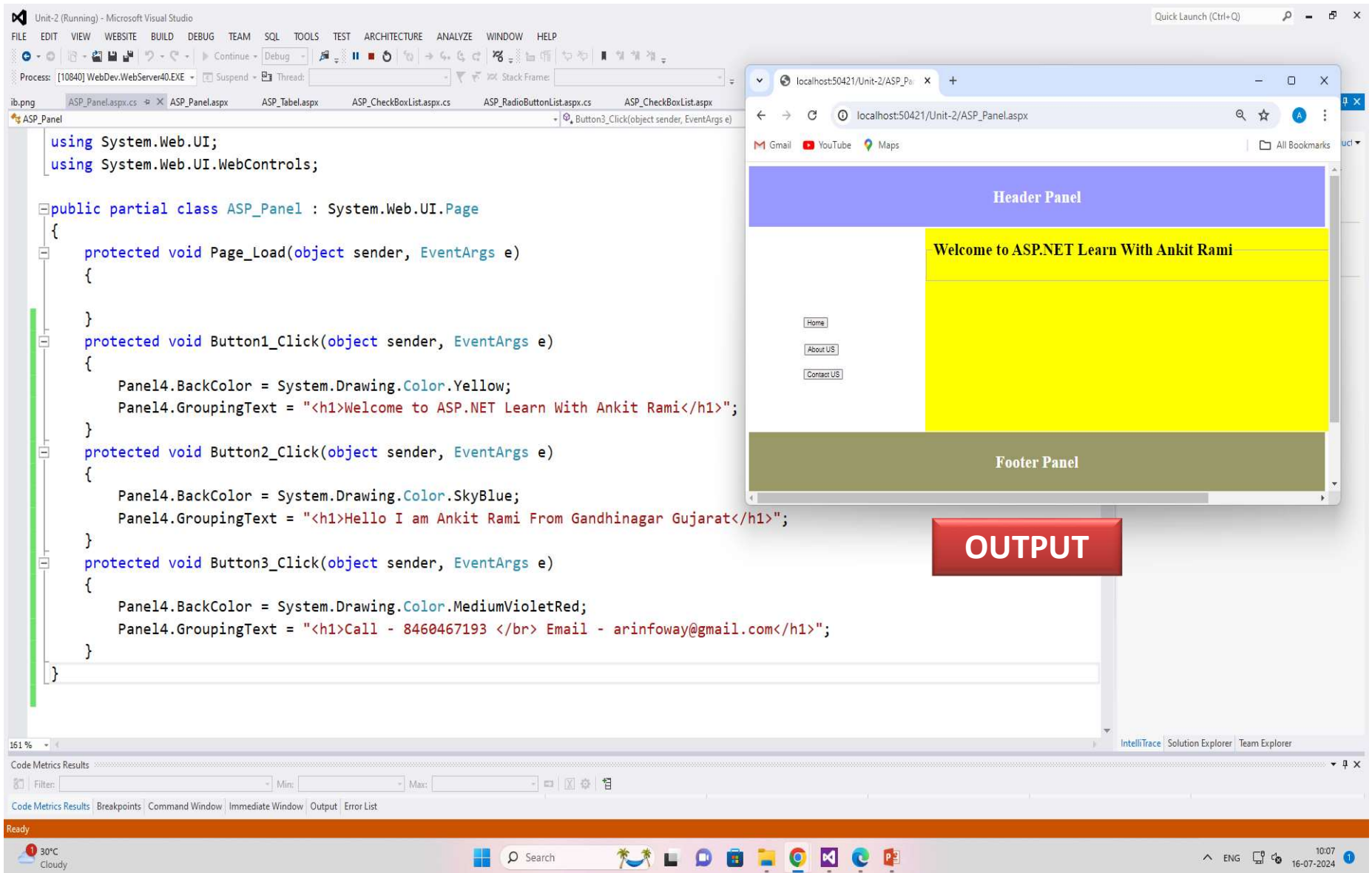
```

<title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:Panel ID="Panel1" runat="server" Height="112px" BackColor="#9999FF" ForeColor="White" HorizontalAlign="Center" Width="100%">
        <br>
        <h1>Header Panel</h1>
      </asp:Panel>
      <asp:Panel ID="Panel3" runat="server" Height="383px" Width="366px">
        <asp:Button ID="Button1" runat="server" Text="Home" OnClick="Button1_Click" style="z-index: 1; left: 123px; top: 288px; position: absolute;" />
        <asp:Button ID="Button3" runat="server" OnClick="Button3_Click" style="z-index: 1; left: 123px; top: 384px; position: absolute;" Text="Contact US" />
        <asp:Button ID="Button2" runat="server" OnClick="Button2_Click" style="z-index: 1; left: 124px; top: 338px; position: absolute;" Text="About US" />
      </asp:Panel>
      <asp:Panel ID="Panel4" runat="server" style="z-index: 1; left: 379px; top: 123px; position: absolute; height: 377px; width: 1146px;"</asp:Panel>
      <asp:Panel ID="Panel2" runat="server" Height="113px" BackColor="#999966" ForeColor="White" HorizontalAlign="Center" Width="100%">
        <br>
        <h1>Footer Panel</h1>
      </asp:Panel>
    </div>
  </form>
</body>

```

- Design View:** Shows the rendered output of the code. It features a blue header panel with the text "Header Panel", a white content area with three buttons labeled "Home", "About US", and "Contact US", and a light blue footer panel.
- Toolbox:** Lists various ASP.NET standard controls such as Button, TextBox, Image, and Panel.
- Solution Explorer:** Shows the project structure, including files like `ASP_BUTTON.aspx`, `ASP_Panel.aspx`, and `web.config`.
- Properties Window:** Displays the properties for the selected `<BR>` control, including attributes like `accesskey`, `aria-activedescendant`, and `aria-atomic`.

# 📢 ASP.NET Standard Controls - II: Panel



The screenshot displays the Visual Studio IDE with a code editor on the left and a browser window on the right. The code editor shows the following C# code for the `ASP_Panel` class:

```

using System.Web.UI;
using System.Web.UI.WebControls;

public partial class ASP_Panel : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Panel4.BackColor = System.Drawing.Color.Yellow;
        Panel4.GroupingText = "<h1>Welcome to ASP.NET Learn With Ankit Rami</h1>";
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        Panel4.BackColor = System.Drawing.Color.SkyBlue;
        Panel4.GroupingText = "<h1>Hello I am Ankit Rami From Gandhinagar Gujarat</h1>";
    }
    protected void Button3_Click(object sender, EventArgs e)
    {
        Panel4.BackColor = System.Drawing.Color.MediumVioletRed;
        Panel4.GroupingText = "<h1>Call - 8460467193 </br> Email - arinfoway@gmail.com</h1>";
    }
}

```

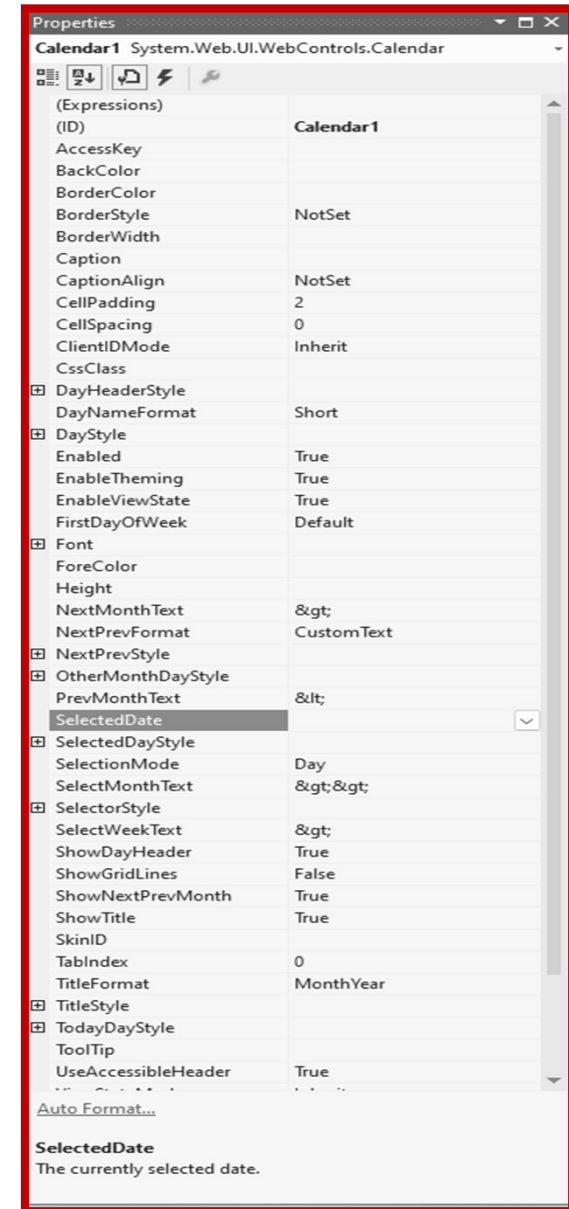
The browser window shows the rendered output of the page. It features a blue header panel, a yellow main content area, and a green footer panel. The content area contains a heading "Welcome to ASP.NET Learn With Ankit Rami" and three buttons: "Home", "About US", and "Contact US".

**OUTPUT**

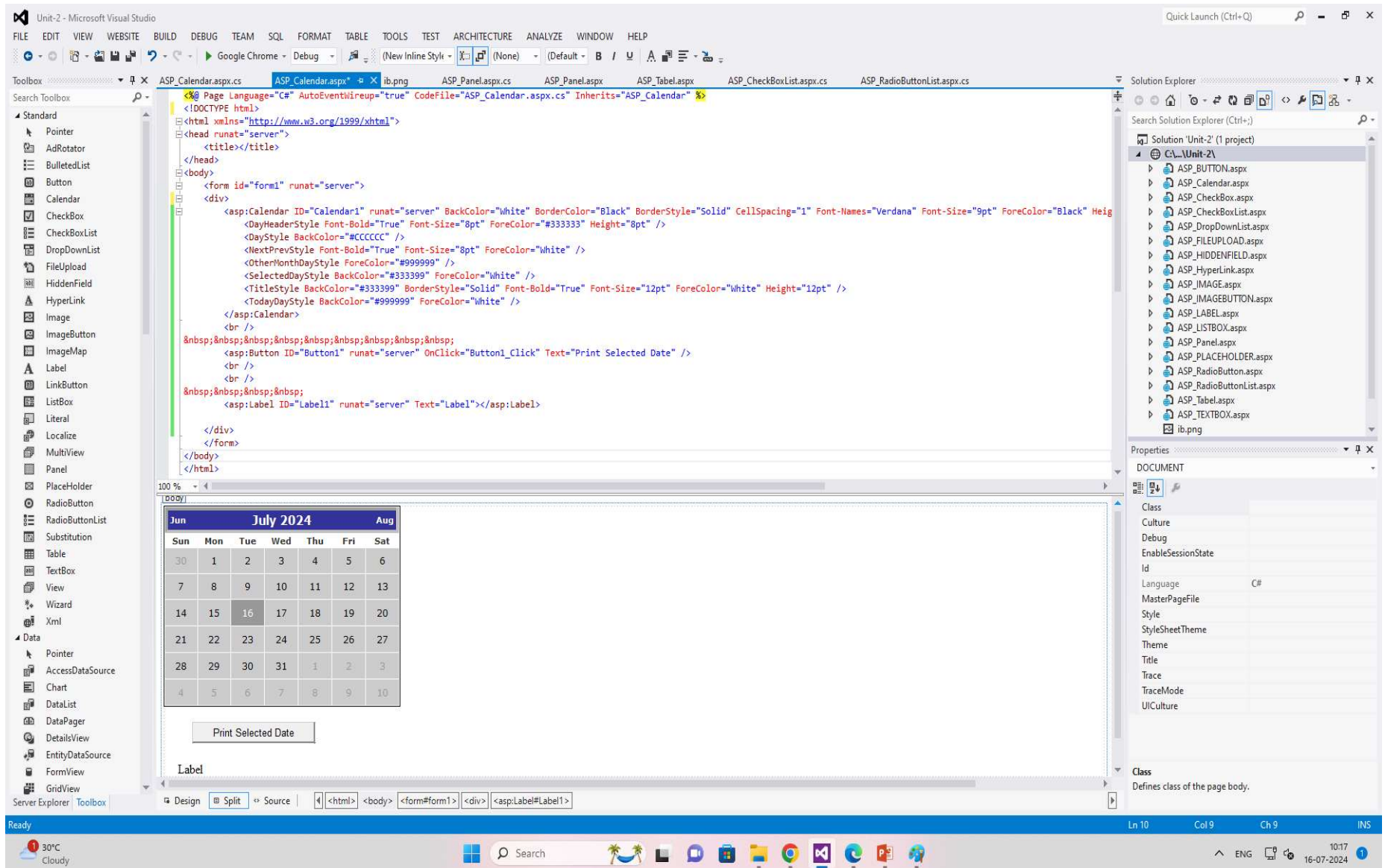
## 📢 ASP.NET Standard Controls - II: Calendar

- ✓ ASP:Calendar is a Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ It is used to display selectable date in a calendar. It also shows data associated with specific date. This control displays a calendar through which users can move to any day in any year.
- ✓ We can also set Selected Date property that shows specified date in the calendar.

Ex- `<asp:Calendar ID="Calendar1" runat="server">  
</asp:Calendar>`



# 📢 ASP.NET Standard Controls - II: Calendar



The screenshot displays the Microsoft Visual Studio environment. The central code editor shows the following ASP.NET code for a calendar control:

```

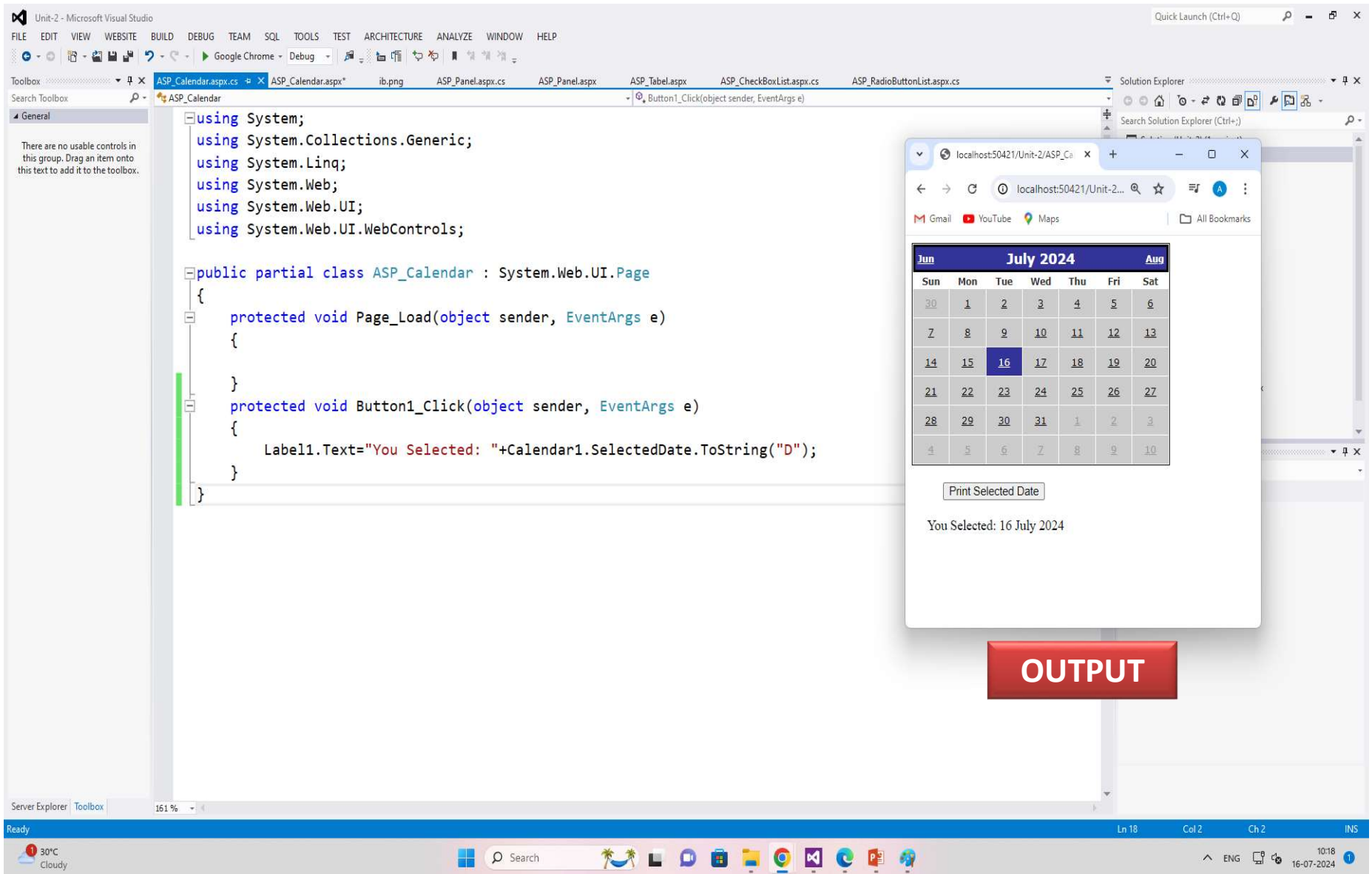
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ASP_Calendar.aspx.cs" Inherits="ASP_Calendar" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:Calendar ID="Calendar1" runat="server" BackColor="White" BorderColor="Black" BorderStyle="Solid" CellSpacing="1" Font-Names="Verdana" Font-Size="9pt" ForeColor="Black" Height="
<DayHeaderStyle Font-Bold="True" Font-Size="8pt" ForeColor="#333333" Height="8pt" />
<DayStyle BackColor="#CCCCCC" />
<NextPrevStyle Font-Bold="True" Font-Size="8pt" ForeColor="White" />
<OtherMonthDayStyle ForeColor="#999999" />
<SelectedDayStyle BackColor="#333399" ForeColor="White" />
<TitleStyle BackColor="#333399" BorderStyle="Solid" Font-Bold="True" Font-Size="12pt" ForeColor="White" Height="12pt" />
<TodayDayStyle BackColor="#999999" ForeColor="White" />
</asp:Calendar>
<br />
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Print Selected Date" />
<br />
<br />
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
</div>
</form>
</body>
</html>
  
```

The Design view at the bottom shows a calendar for July 2024. The calendar is a table with columns for days of the week (Sun to Sat) and rows for dates. The selected date is 16th July 2024. Below the calendar is a button labeled "Print Selected Date" and a label.

Jun	July 2024						Aug
Sun	Mon	Tue	Wed	Thu	Fri	Sat	
30	1	2	3	4	5	6	
7	8	9	10	11	12	13	
14	15	16	17	18	19	20	
21	22	23	24	25	26	27	
28	29	30	31	1	2	3	
4	5	6	7	8	9	10	



# 📢 ASP.NET Standard Controls - II: Calendar



The screenshot displays the Visual Studio IDE with a C# code file for an ASP.NET page named `ASP_Calendar.aspx.cs`. The code includes the following:

```

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

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

    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text="You Selected: "+Calendar1.SelectedDate.ToString("D");
    }
}

```

Overlaid on the right side of the IDE is a browser window showing the rendered output. It features a calendar for July 2024. The date 16th July is selected and highlighted in blue. Below the calendar is a "Print Selected Date" button and a text label that reads "You Selected: 16 July 2024".

**OUTPUT**



## 📢 ASP.NET Standard Controls - II: AdRotator

- ✓ ASP:AdRotator is a Microsoft Visual Studio Standard Control Tools.
- ✓ It is always run on Server.
- ✓ The AdRotator control randomly selects banner graphics from a list, which is specified in an external XML schedule file. This external XML schedule file is called the advertisement file.
- ✓ The AdRotator control allows you to specify the advertisement file and the type of window that the link should follow in the AdvertisementFile and the Target property respectively.
- ✓ Ex- `<asp:AdRotator ID="AdRotator1" runat="server" />`



Properties	
AdRotator1 System.Web.UI.WebControls.AdRotator	
(Expressions)	
(ID)	AdRotator1
AccessKey	
AdvertisementFile	
AlternateTextField	AlternateText
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
ClientIDMode	Inherit
CssClass	
DataMember	
DataSourceID	
Enabled	True
EnableTheming	True
EnableViewState	True
ForeColor	
Height	
ImageUrlField	ImageUrl
KeywordFilter	
NavigateUrlField	NavigateUrl
SkinID	
TabIndex	0
Target	_top
ToolTip	
ViewStateMode	Inherit
Visible	True
Width	
<b>AdvertisementFile</b>	
XML file containing advertisements.	

# ASP.NET Standard Controls - II: AdRotator

## The Advertisement File

- ✓ The advertisement file is an XML file, which contains the information about the advertisements to be displayed.
- ✓ Extensible Markup Language (XML) is a W3C standard for text document markup.
- ✓ It is a text-based markup language that enables you to store data in a structured format by using meaningful tags.
- ✓ XML is not a language in itself, like HTML, but a set of rules for creating new markup languages.
- ✓ It is a meta-markup language.
- ✓ It allows developers to create custom tag sets for special uses. It structures, stores, and transports the information.

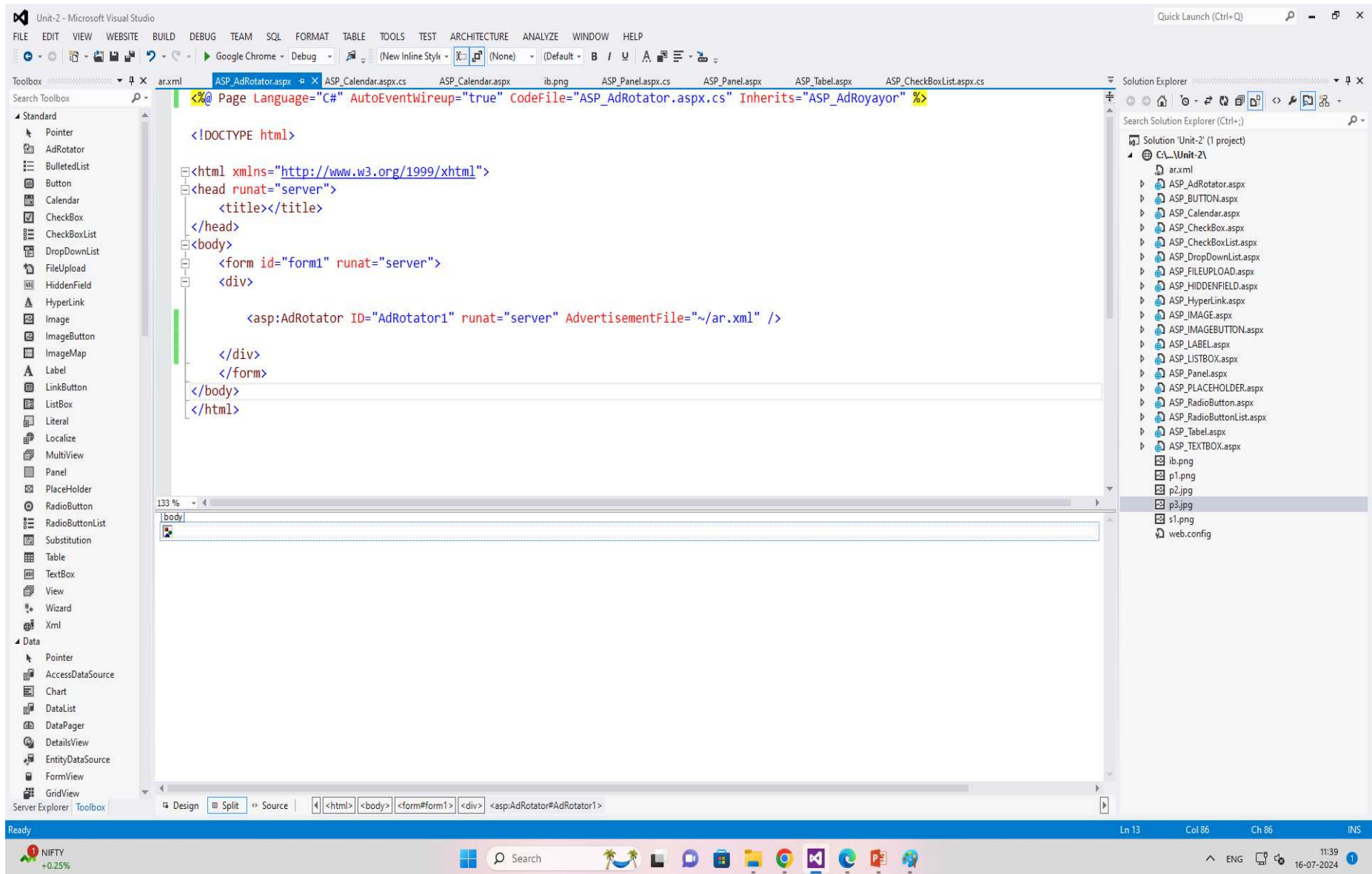
Element	Description
Advertisements	Encloses the advertisement file.
Ad	Delineates separate ad.
ImageUrl	The path of image that will be displayed.
NavigateUrl	The link that will be followed when the user clicks the ad.
AlternateText	The text that will be displayed instead of the picture if it cannot be displayed.
Keyword	Keyword identifying a group of advertisements. This is used for filtering.
Impressions	The number indicating how often an advertisement will appear.
Height	Height of the image to be displayed.
Width	Width of the image to be displayed.

### Syntax

```

<Advertisements>
  <Ad>
    <ImageUrl> Path </ImageUrl>
    <NavigateUrl> Link </NavigateUrl>
    <AlternateText>
      Details
    </AlternateText>
    <Impressions> Numbers </Impressions>
    <Keyword> Filtering Keyword </Keyword>
  </Ad>
</Advertisements>
  
```

# 📢 ASP.NET Standard Controls - II: AdRotator



The screenshot displays the Microsoft Visual Studio IDE with the following components:

- Toolbox:** Shows various ASP.NET standard controls under the 'Standard' and 'Data' categories. The 'AdRotator' control is highlighted in the 'Standard' category.
- Code Editor:** Shows the source code for `ASP_AdRotator.aspx`. The code includes:
 

```

      <%@ Page Language="C#" AutoEventWireup="true" CodeFile="ASP_AdRotator.aspx.cs" Inherits="ASP_AdRoyayor" %>

      <!DOCTYPE html>

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

      <asp:AdRotator ID="AdRotator1" runat="server" AdvertisementFile="~/ar.xml" />

      </div>
      </form>
      </body>
      </html>
      
```
- Solution Explorer:** Shows the project structure for 'Unit-2', listing various ASP.NET controls like `ASP_Button.aspx`, `ASP_Calendar.aspx`, `ASP_Calendar.aspx.cs`, `ASP_Panel.aspx`, `ASP_Panel.aspx.cs`, `ASP_Tabel.aspx`, `ASP_Tabel.aspx.cs`, `ASP_CheckBoxList.aspx`, `ASP_CheckBoxList.aspx.cs`, `ASP_FileUpload.aspx`, `ASP_FileUpload.aspx.cs`, `ASP_HiddenField.aspx`, `ASP_HiddenField.aspx.cs`, `ASP_HyperLink.aspx`, `ASP_HyperLink.aspx.cs`, `ASP_Image.aspx`, `ASP_ImageButton.aspx`, `ASP_ImageButton.aspx.cs`, `ASP_Label.aspx`, `ASP_ListBox.aspx`, `ASP_ListBox.aspx.cs`, `ASP_Panel.aspx`, `ASP_Panel.aspx.cs`, `ASP_Placeholder.aspx`, `ASP_Placeholder.aspx.cs`, `ASP_RadioButton.aspx`, `ASP_RadioButton.aspx.cs`, `ASP_RadioButtonList.aspx`, `ASP_RadioButtonList.aspx.cs`, `ASP_Tabel.aspx`, `ASP_Tabel.aspx.cs`, `ASP_TextBox.aspx`, `ASP_TextBox.aspx.cs`, `ib.png`, `p1.png`, `p2.jpg`, `p3.jpg`, `s1.png`, and `web.config`.
- Design View:** Shows a preview of the rendered page, displaying the `asp:AdRotator` control in the center of the form.
- Taskbar:** Shows the Windows taskbar with the Start button, search bar, and various application icons. The system tray shows the time as 11:39 and the date as 16-07-2024.

# ASP.NET Standard Controls - II: AdRotator

The screenshot displays the Visual Studio IDE with the following components:

- Code Editor:** Shows the XML configuration for the `AdRotator` control in `ar.xml`. The code defines three advertisements:
 

```

            <?xml version="1.0" encoding="utf-8" ?>
            <Advertisements>
              <Ad>
                <ImageUrl>p1.png</ImageUrl>
                <NavigateUrl>https://prochipper.in/</NavigateUrl>
                <AlternateText>Home Theater</AlternateText>
                <Impressions>20</Impressions>
                <Keyword>prochipper</Keyword>
              </Ad>
              <Ad>
                <ImageUrl>p2.jpg</ImageUrl>
                <NavigateUrl>https://prochipper.in/</NavigateUrl>
                <AlternateText>Laptop</AlternateText>
                <Impressions>20</Impressions>
                <Keyword>prochipper</Keyword>
              </Ad>
              <Ad>
                <ImageUrl>p3.jpg</ImageUrl>
                <NavigateUrl>https://prochipper.in/</NavigateUrl>
                <AlternateText>Desktop Computer</AlternateText>
                <Impressions>20</Impressions>
                <Keyword>prochipper</Keyword>
              </Ad>
            </Advertisements>
            
```
- Browser Preview:** Three browser windows show the rendered output of the `AdRotator` control. Red arrows point from the corresponding XML elements to the browser windows:
  - The first window shows an advertisement for "Vardayini Infosys" featuring a home theater system, with the alternate text "Home Theater" visible.
  - The second window shows an advertisement for "Vardayini Infosys" featuring a laptop, with the alternate text "Laptop" visible.
  - The third window shows an advertisement for "Vardayini Infosys" featuring a desktop computer, with the alternate text "Desktop Computer" visible.
- Solution Explorer:** Lists the project files, including `ar.xml`, `ASP_AdRotator.aspx`, and various ASP.NET controls.
- Taskbar:** Shows the Windows taskbar with the system tray displaying the time as 11:40 on 16-07-2024.

**OUTPUT**




## Reference Link for E-Learning



**AMIT ACADEMY**  
for Computer Education

- ❖ <https://www.javatpoint.com/asp-net-tutorial>
- ❖ <https://www.javatpoint.com/c-sharp-tutorial>
- ❖ [https://www.geeksforgeeks.org/csharp-programming-language/?ref=header\\_search](https://www.geeksforgeeks.org/csharp-programming-language/?ref=header_search)
- ❖ <https://www.ankitweblogic.com/asp/index.php>
- ❖ <https://www.tutorialspoint.com/asp.net/>
- ❖ <https://amit.arinfoway.com>
- ❖ <https://www.youtube.com/channel/UCWbJh2iQ8w-8nrU0Xpjpw7g>





### AR INFOWAY

@ankitramijoinar · 853 subscribers · 124 videos

Hello friends, my name is Ankit and welcome to my ARINFOWAY channel. ...more

[arinfoway.com](https://amit.arinfoway.com) and 2 more links

 Subscribed 



Like



Comment



Share

SUBSCRIBE



**More Information Join  
YouTube Channel**

**View More**