Introduction to ASP.NET

ASP.NET is a web application framework developed by **Microsoft** for building dynamic websites, web applications, and web services. It runs on the .NET platform and allows developers to create interactive web pages using languages such as C# or VB.NET.

Features of ASP.NET

- Open-source and cross-platform (ASP.NET Core).
- Supports rapid application development.
- Built-in security features like authentication and authorization.
- Server-side processing.
- Easy integration with databases.
- Supports MVC, Web Forms, and Web API.

Advantages

- High performance.
- Easy deployment.
- Rich set of server controls.
- Better security.
- Code reusability.

---

2. ASP.NET and HTML Controls

Controls are elements used to create the user interface of a web page.

A. HTML Controls

HTML controls are standard HTML elements that can be accessed on the server by adding the runat="server" attribute.

Examples

<input type="text" id="txtName" runat="server" />

<input type="button" value="Submit" runat="server" />

Common HTML Controls

Control	Description
TextBox (<input type="text">)	Accepts text input
Button	Performs an action when clicked
CheckBox	Allows multiple selections
RadioButton	Allows one selection from a group
Select (<select>)	Displays a drop-down list

B. ASP.NET Server Controls

ASP.NET provides server controls that automatically generate HTML and offer additional features.

Examples

<asp:TextBox ID="txtName" runat="server"></asp:TextBox>

<asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />

Common ASP.NET Controls

Control	Purpose
TextBox	Input text
Label	Display text
Button	Trigger an event
CheckBox	Yes/No selection
RadioButton	Single choice
DropDownList	List of options
GridView	Display data in tabular form
Calendar	Select a date

The image shows **Common ASP.NET Controls**. Below is the explanation, syntax, and examples for each control.

1. TextBox Control

Purpose: Used to accept input from the user.

Syntax

<asp:TextBox ID="txtName" runat="server"></asp:TextBox>

Example

Name:

<asp:TextBox ID="txtName" runat="server"></asp:TextBox>

Output: User can type text such as "John".

VB.Net code

Protected Sub btnShow_Click(sender As Object, e As EventArgs) Handles btnShow.Click
lblResult.Text = "Welcome " & txtName.Text End Sub

---

2. Label Control

Purpose: Displays text on a web page.

Syntax

<asp:Label ID="lblMessage" runat="server" Text="Welcome"></asp:Label>

Example

<asp:Label ID="lblMessage" runat="server"

Text="Welcome to ASP.NET"></asp:Label>

Output:
Welcome to ASP.NET

VB.Net code

Protected Sub btnShow_Click(sender As Object, e As EventArgs) Handles btnShow.Click
lblResult.Text = "Welcome " & txtName.Text End Sub

---

3. Button Control

Purpose: Triggers an event when clicked.

Syntax

<asp:Button ID="btnSubmit" runat="server"

Text="Submit" OnClick="btnSubmit_Click" />

Example

<asp:Button ID="btnSubmit" runat="server"

Text="Submit" OnClick="btnSubmit_Click" />

VB.net Code

Protected Sub btnClick_Click(sender As Object, e As EventArgs) Handles btnClick.Click
lblOutput.Text = "Button Clicked Successfully"

End Sub

---

4. CheckBox Control

Purpose: Allows Yes/No or True/False selection.

Syntax

<asp:CheckBox ID="chkAgree" runat="server"

Text="I Agree" />

Example

<asp:CheckBox ID="chkAgree" runat="server"

Text="Accept Terms and Conditions" />

VB.Net Code

Protected Sub btnCheck_Click(sender As Object, e As EventArgs) Handles btnCheck.Click

If chkAgree.Checked Then lblStatus.Text = "Accepted" Else lblStatus.Text = "Not Accepted"

End If

End Sub

---

5. RadioButton Control

Purpose: Allows a single choice from a group.

Syntax

<asp:RadioButton ID="rbMale" runat="server"

GroupName="Gender" Text="Male" />

Example

```
<asp:RadioButton ID="rbMale" runat="server"
   GroupName="Gender" Text="Male" />
<asp:RadioButton ID="rbFemale" runat="server"
   GroupName="Gender" Text="Female" />
```

VB.Net Code

```
Protected Sub btnGender_Click(sender As Object, e As EventArgs) Handles btnGender.Click
If rbMale.Checked Then lblGender.Text = "Male Selected"
ElseIf rbFemale.Checked
Then lblGender.Text = "Female Selected"
End If
End Sub
```

DropDownList

ASPX

```
<asp:DropDownList ID="ddlCity" runat="server">
   <asp:ListItem>Pune</asp:ListItem>
   <asp:ListItem>Mumbai</asp:ListItem>
   <asp:ListItem>Latur</asp:ListItem>
</asp:DropDownList>
<asp:Button ID="btnCity" runat="server" Text="Show City" />
<asp:Label ID="lblCity" runat="server"></asp:Label>
```

VB.NET

```
Protected Sub btnCity_Click(sender As Object, e As EventArgs) Handles btnCity.Click
   lblCity.Text = "Selected City: " & ddlCity.SelectedItem.Text
End Sub
```

GridView

ASPX

```
<asp:GridView ID="GridView1" runat="server">
</asp:GridView>
```

VB.NET

```
Imports System.Data

Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load

   If Not IsPostBack Then

      Dim dt As New DataTable()

      dt.Columns.Add("ID")

      dt.Columns.Add("Name")

      dt.Rows.Add("1", "Rahul")

      dt.Rows.Add("2", "Priya")

      GridView1.DataSource = dt

      GridView1.DataBind()

   End If

End Sub
```

Output

ID	Name
1	Rahul
2	Priya

Calendar

ASPX

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

<asp:Label ID="lblDate" runat="server"></asp:Label>
```

VB.NET

```
Protected Sub Calendar1_SelectionChanged(sender As Object, e As EventArgs) Handles
Calendar1.SelectionChanged


   lblDate.Text = "Selected Date: " &

            Calendar1.SelectedDate.ToShortDateString()
```

End Sub

Difference Between HTML Controls and ASP.NET Controls

HTML Controls	ASP.NET Controls
Standard HTML elements	Server-side controls
Limited functionality	Rich functionality
Need JavaScript for many tasks	Built-in server-side features
Faster rendering	Supports events and state management

---

3. ASP.NET Events and Event Handlers

An **event** is an action performed by the user or the system, such as clicking a button, changing text, or loading a page.

An **event handler** is a method that executes automatically when an event occurs.

Common ASP.NET Events

Event	Description
Page_Load	Occurs when the page loads
Click	Triggered when a button is clicked
TextChanged	Occurs when text changes
SelectedIndexChanged	Occurs when a selection changes
CheckedChanged	Occurs when a checkbox state changes

Example: Button Click Event

ASPX Page

```
<asp:TextBox ID="txtName" runat="server"></asp:TextBox>


<asp:Button ID="btnSubmit"
    runat="server"
    Text="Submit"
```

OnClick="btnSubmit_Click" />

<asp:Label ID="lblMessage" runat="server"></asp:Label>

VB.Net Code

Protected Sub btnSubmit_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btnSubmit.Click lblMessage.Text = "Welcome " & txtName.Text

End Sub

Output:

- If the user enters **Rahul** and clicks **Submit**, the label displays:

Welcome Rahul

Page Life Cycle Events

1. **Page_Init** – Initializes page controls.
2. **Page_Load** – Loads the page and its controls.
3. **Button_Click** – Executes when the button is clicked.
4. **Page_PreRender** – Occurs just before the page is rendered.
5. **Page_Unload** – Cleans up resources after the page is processed.

Feature	**HTML Controls**	**ASP.NET Web Controls**
Definition	Standard HTML elements used in web pages.	Server-side controls provided by ASP.NET.
Runs on Server	No (unless runat="server" is added).	Yes, always requires runat="server".
Processing	Processed mainly by the browser.	Processed on the server before being sent to the browser.
Properties	Limited HTML attributes.	Rich set of properties, methods, and events.
Events	Client-side events (JavaScript).	Server-side events such as Click, TextChanged, etc.

Feature	HTML Controls	ASP.NET Web Controls
State Management	Does not automatically maintain state.	Supports ViewState to maintain values across postbacks.
Programming	Less server-side functionality.	Easier to manipulate in C# or VB.NET code-behind.

Examples

HTML Control

<input type="text" id="txtName">

<input type="button" value="Submit">

HTML Control with runat="server"

<input type="text" id="txtName" runat="server">

ASP.NET Web Controls

<asp:TextBox ID="txtName" runat="server"></asp:TextBox>

<asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />

Advantages of HTML Controls

- Simple and lightweight.
- Faster rendering.
- Ideal for static pages.
- Works directly with standard HTML and JavaScript.

Advantages of ASP.NET Web Controls

- Rich server-side functionality.
- Built-in event handling.
- Automatic state management using ViewState.
- Easier data binding and validation.
- Consistent programming model across controls.

Summary

- **HTML Controls** are standard HTML elements and are best for simple, lightweight web pages.

- **ASP.NET Web Controls** are server controls with built-in features such as event handling, state management, and data binding, making them suitable for dynamic, data-driven ASP.NET applications.

ASP.NET Events and Event Handlers

What is an Event?

An **event** is an action that occurs on a web page. Events can be triggered by the user or by the system.

Examples:

- Clicking a button
- Changing text in a textbox
- Selecting an item from a dropdown list
- Loading a page

In ASP.NET Web Forms, events are handled on the **server**.

What is an Event Handler?

An **event handler** is a method (function) that is automatically executed when an event occurs.

Syntax

```
protected void ControlName_EventName(object sender, EventArgs e)
{
   // Code to execute
}
```

- **sender** → The control that raised the event.
- **EventArgs e** → Contains information about the event.

ASP.NET Event Life Cycle

When a page is requested, ASP.NET processes several events in order:

1. Page_Init
2. Page_Load
3. Control Events (Button Click, Text Changed, etc.)

4. Page_PreRender
5. Page_Unload

Common ASP.NET Events

Event	**Description**
Page_Load	Occurs when the page is loaded.
Click	Occurs when a button is clicked.
TextChanged	Occurs when textbox text changes.
SelectedIndexChanged	Occurs when dropdown selection changes.
CheckedChanged	Occurs when checkbox state changes.

Program 1: Button Click Event

Default.aspx

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


<!DOCTYPE html>

<html>

<head>

  <title>Button Click Example</title>

</head>

<body>

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

    <asp:Label ID="lblMessage" runat="server" Text=""></asp:Label>

    <br /><br />

    <asp:Button ID="btnClick"

      runat="server"

      Text="Click Me"
```

```
        OnClick="btnClick_Click" />
    </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
public partial class _Default : System.Web.UI.Page
{
    protected void btnClick_Click(object sender, EventArgs e)
    {
        lblMessage.Text = "Welcome to ASP.NET!";
    }
}
```

Output

Before clicking:

[Click Me]

After clicking:

Welcome to ASP.NET!

Program 2: TextBox TextChanged Event

Default.aspx

```
<asp:TextBox ID="txtName"
    runat="server"
    AutoPostBack="true"
    OnTextChanged="txtName_TextChanged">
</asp:TextBox>

<br /><br />
```

```
<asp:Label ID="lblName" runat="server"></asp:Label>
```

Default.aspx.cs

```
protected void txtName_TextChanged(object sender, EventArgs e)
{
    lblName.Text = "Hello " + txtName.Text;
}
```

Output

Input:

Rahul

Output:

Hello Rahul

Program 3: Page Load Event

Default.aspx.cs

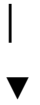
```
using System;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            lblMessage.Text = "Page Loaded Successfully!";
        }
    }
}
```

Output

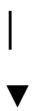
Page Loaded Successfully!

How Event Handling Works

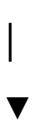
User Clicks Button



Button Click Event Occurs



btnClick_Click() Event Handler Executes



Response Sent to Browser

Advantages of Event Handling

- Makes web pages interactive.
- Executes code only when a specific action occurs.
- Simplifies server-side programming.
- Supports user input validation and processing.
- Enables dynamic updates to page content.

Viva/Exam Questions

1. What is an event in ASP.NET?
An event is an action performed by the user or system, such as clicking a button or loading a page.

2. What is an event handler?
An event handler is a method that executes automatically when an event occurs.

3. Why is AutoPostBack="true" used?
It sends the page back to the server immediately when the control's value changes, allowing the associated server-side event (such as TextChanged) to execute.

4. What is IsPostBack?
IsPostBack is a property that indicates whether the page is being loaded for the first time (false) or in response to a postback caused by a control event (true).

Key Points to Remember

- **Event**: An action such as Click, Load, or TextChanged.
- **Event Handler**: The method that responds to the event.
- **runat="server"**: Required for ASP.NET server controls.
- **AutoPostBack="true"**: Triggers an immediate postback for controls like TextBox and DropDownList.
- **IsPostBack**: Prevents code from running repeatedly on every postback.