

CS-32: Programming with ASP.NET

UNIT – 1

Framework and Web Contents
Validation Controls

Dharmesh Kapadiya
Dhruvita Savaliya

TOPICS :

- Overview of Asp.NET Framework
- Client Server Architecture
- Application Web Servers
- Types of Files in Asp.NET
- Types of controls in Asp.NET
- Page Architecture
- Web form
- Introduction to standard Controls
(Buttons, Textbox, Checkbox, Lable, Panel, Listbox, Dropdownlist etc.)
- Running an Asp.Net Application, File Upload Control
- What is Validation?
 - Client Side Validation
 - Server Side Validation
- Types (RequieredField Validator, Range Validator, CompareField Validator, RegularExpression Validator, Custom Validator, ValidationSummery Control)

BASIC TERMS :

➤ What is Web Page?

- A document which can be displayed in a web browser such as Firefox, Google Chrome, Opera, Microsoft Edge, or Apple Safari.
- These are also often called just "pages."

➤ What is Website?

- A collection of web pages which are grouped together and usually connected together in various ways.
- Often called a "website" or a "site". it is having static content.

➤ What is Web App or Web Application?

- Web application main motive is to take input from the user like from visitors to subscribe channel, online image edit service etc. it is having dynamic content.
- To access web app we required web server, Application server & database.

- **What is Web Server?**

A web server is a software program that serves web pages to web users (browsers).

A web server delivers requested web pages to users who enter the URL in a web browser.

Every computer on the internet that contains a web site must have a web server program.

Examples of web servers: IIS, Apache

- **What is Application server?**

Application servers work as an intermediary between databases, which store application data, and web clients.

They also communicate with web servers, which deliver content to the web client.

Overview of Asp.NET Framework :

- ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites.
- It allows you to use a full featured programming language such as C# or VB.NET to build web applications easily.
- ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices.
- ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation.
- ASP.NET is a part of Microsoft .Net platform.

- ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework.
- These codes can use the entire hierarchy of classes in .Net framework.
- The ASP.NET application codes can be written in any of the following languages:
 - C#
 - Visual Basic.Net
 - Jscript
 - J#
- ASP.NET is used to produce interactive, data-driven web applications over the internet.
- It consists of a large number of controls such as text boxes, buttons, and labels for assembling, configuring, and manipulating code to create HTML pages.

- **What is ASP.NET?**
- The ASP stands for **Active Server Pages**, and **.NET is Network Enabled Technologies**.
- Asp.net was released in January 2002.
 - Latest version of ASP.NET Core is v8.0.0 → November 14, 2023. The next version, .NET 9 → November 12, 2024
- ASP.NET pages have the extension .aspx and are normally written in C#(C sharp).
- ASP.Net is a web development platform provided by Microsoft.
- ASP.Net is designed to work with the HTTP protocol.
- This is the standard protocol used across all web applications.
- It is used for creating powerful dynamic web site, web application and web services.
- It provides fantastic integration of HTML, CSS and JavaScript.
- ASP.Net applications can also be written in a variety of **.Net languages**.
 - These include C#, VB.Net, and J#.
- ASP supports many different development models:
 - 1. Classic ASP, 2. ASP.NET Web Forms, 3. ASP.NET MVC, 4. ASP.NET Web Pages, 5. ASP.NET API, 6. ASP.NET Core.

- **ADVANTAGES OF ASP.NET :**

It uses .net framework.

It is compiled not interpreted

It uses multilanguage

It uses object oriented features

Stores presentation logic and application logic differently

It gives multibrowser facility

It gives multidevice facility

Facility of caching

Inbuilt logic controls

ADO.net with good data formatting

Inbuilt validation control

Strong & easy configuration

Master page

Easy to deploy & configure

Facility to create web services

Uses latest concept like AJAX, Silverlight, LINQ, etc

ASP	ASP.NET	9
ASP is the interpreted language.	ASP.NET is the compiled language.	
ASP uses ADO (ActiveX Data Objects) technology to connect and work with databases.	ASP.NET uses ADO.NET to connect and work with databases.	
ASP is partially object-oriented.	ASP.NET is fully object-oriented.	
In ASP there is no facility to separate design from programming logic.	In ASP.NET it has the option of Code Containment.	
ASP Pages have the file extension .asp .	ASP.NET Pages have the file extension .aspx .	
ASP doesn't have the concept of inheritance .	ASP.NET inherit from class written in code behind.	
ASP pages use scripting language .	ASP.NET uses a full-fledged programming language.	
Error handling is very poor in ASP.	Error handling is very good in ASP.NET.	
In ASP debugging is difficult because the ASP scripts are interpreted .	In ASP.NET debugging is easy .	
ASP is not a configurable language.	In ASP.NET Web.config is used for configuration.	
ASP has maximum four in-built classes i.e. Request, Response, Session and Application.	ASP.NET has more than 2000 in-built classes .	
Custom Controls can not be achieved by ASP.	Custom Controls can be achieved by ASP.NET using @register directive .	

- **WHAT IS .NET?**

- NET stands for Network Enabled Technology (Internet).
- In .NET, dot (.) refers to Object-Oriented, and NET refers to the internet.
- So, the complete .NET means through Object-Oriented we can implement internet-based applications.
- According to Microsoft, .NET is a Free, Cross-Platform, Open-Source developer platform for building many different types of applications.
- With .NET, we can use multiple languages (C#, VB, F#, etc.),
- Editors (Visual Studio, Visual Studio Code, Visual Studio for Mac, OmniSharp, JetBrains Rider, etc), and Libraries to build for Web, Mobile, Desktop, Games, IoT, and more
- **Cross Platform:** Whether you are working in C#, F#, or Visual Basic, your code will run on any compatible operating system. You can build many types of apps with .NET. Some are Cross-Platform, and some target a specific set of operating systems and devices.
- **Libraries:** To extend functionality, Microsoft and others maintain a healthy .NET package ecosystem.
- NuGet(ZIP file) is a package manager built specifically for .NET that contains over 100,000 packages(DLL libraries).

- **WHAT IS A FRAMEWORK?**

- A framework is a software. Or you can say a framework is a collection of many small technologies integrated together to develop applications that can be executed anywhere.
- .NET support 60+ programming languages, 9 are **designed by Microsoft** and the remaining are designed by non-Microsoft.

- **Microsoft-designed** programming languages are as follows:

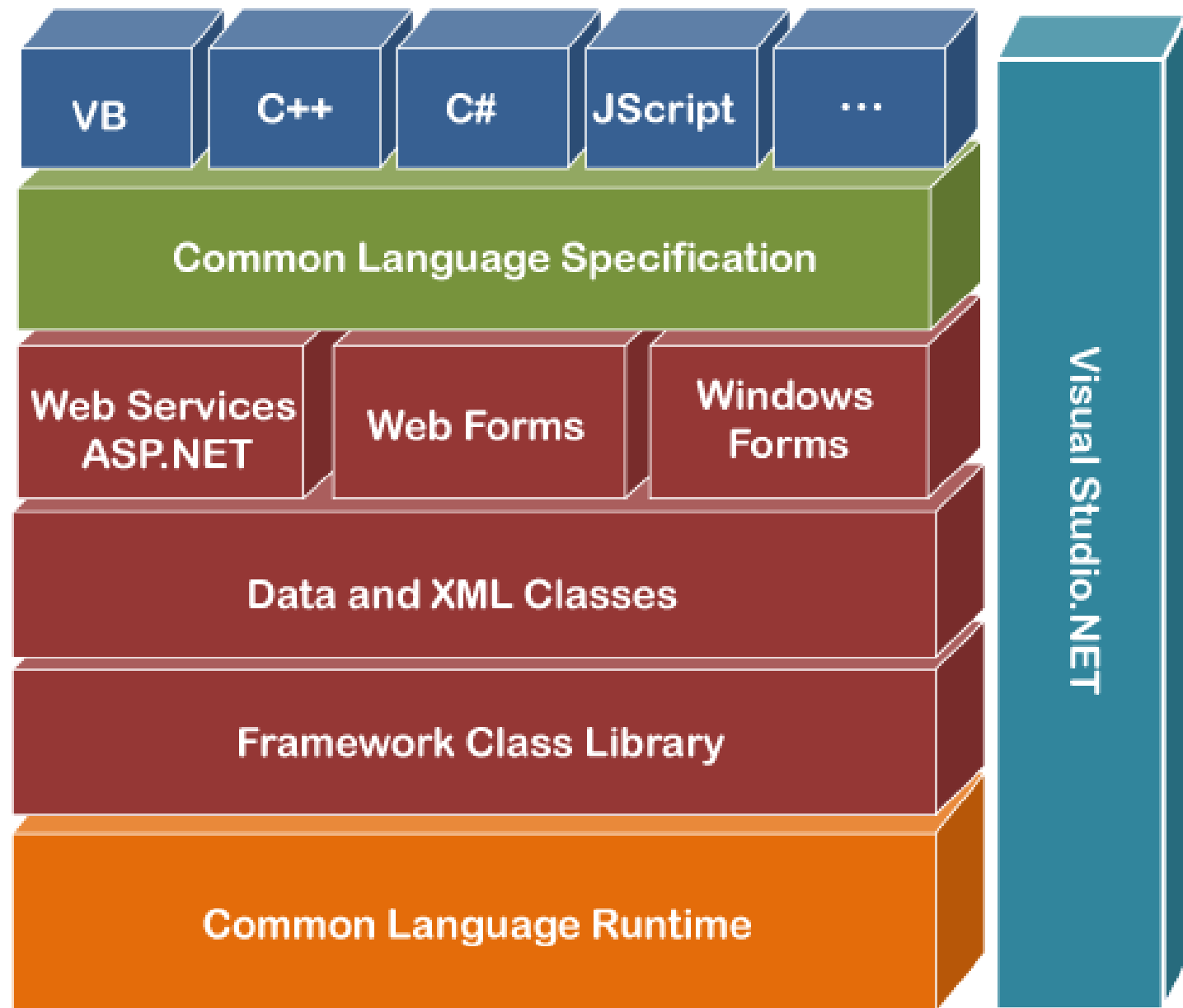
- VB.NET
- C#.NET
- VC++.NET
- J#.NET
- F#.NET
- Jscript.NET
- WindowsPowerShell
- Iron phyton
- Iron Ruby

- **Technologies** supported by the .NET framework are as Follows:

- ASP.NET (Active Server Pages.NET) – MVC, Web API, Core MVC, Core Web API, Core Blazor, etc.
- ADO.NET (Active Data Object.NET)
- WCF (Windows Communication Foundation)
- WPF (Windows Presentation Foundation)
- WWF (Windows Workflow Foundation)
- AJAX (Asynchronous JavaScript and XML)
- LINQ (Language Integrated Query)
- Entity Framework

.NET Framework Architecture :

- **.NET Framework Architecture** is a programming model for the .NET platform that provides an execution environment and integration with various programming languages for simple development and deployment of various Windows and desktop applications.
- It consists of class libraries and reusable components.
- There are Main Two Components of .NET framework :
 1. CLR : Common Language Runtime.
 2. FCL : Framework Class Library.



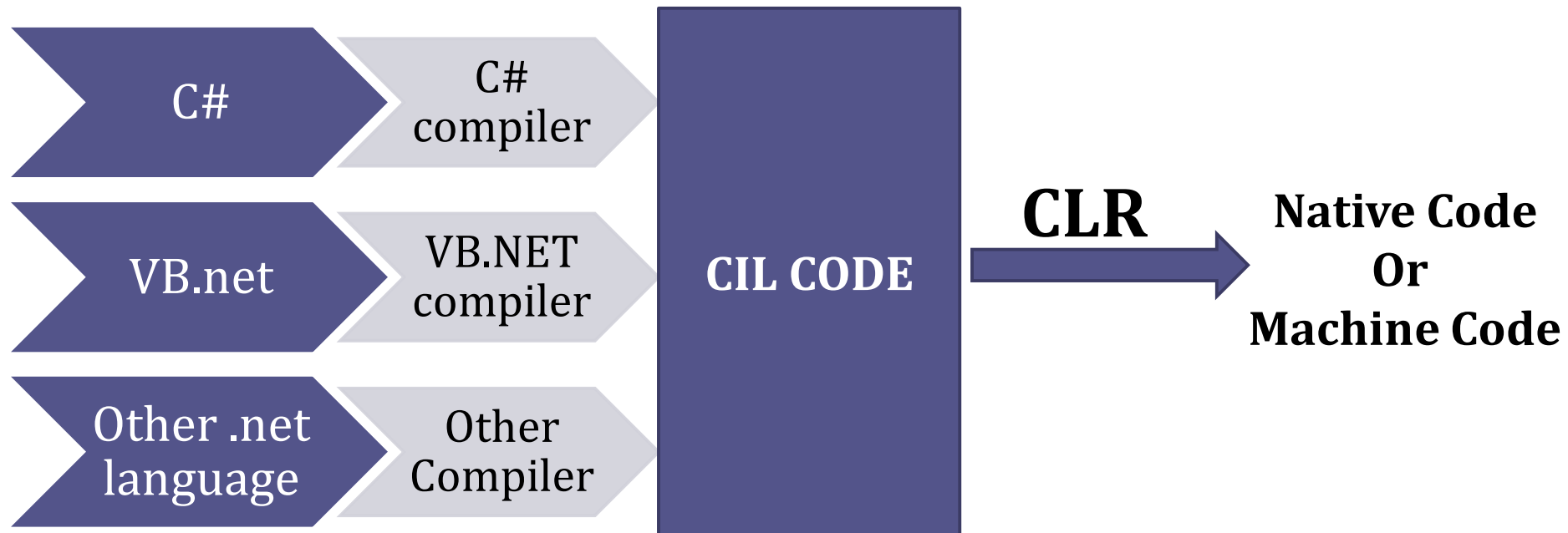
Components of .NET Framework :

1. CLR (Common Language Runtime)
2. CTS (Common Type System)
3. CLS (Common Language Specification)
4. FCL (Framework Class Library)
5. BCL (Base Class Library)
6. .NET Assemblies
7. XML Web Services
8. Window Services

❖ What is Common Language Runtime (CLR):

- CLR is execution engine that loads and executes the program.
- It is the heart of the .NET.
- It is runtime environment.
- CLR is a virtual machine.
- CLR is runtime engine provided by the .NET framework.
- It provides an infrastructure for running programs and allows them to communicate with other parts of the .NET framework.
- It acts as an interface between the framework and operating system.
- A CLR also helps to convert a source code into the byte code, and this byte code is known as CIL (Common Intermediate Language) or MSIL (Microsoft Intermediate Language).
- After converting into a byte code, a CLR uses a JIT(Just In Time) compiler at run time that helps to convert a CIL or MSIL code into the machine or native code.

- .NET Running Environment :



Components of .NET CLR :

- The key components of CLR include the following:
- **Class Loader :**
Used to load all classes at run time.
- **MSIL(Microsoft Intermediate Language) to Native code :**
The Just In Time (JTI) compiler will convert MSIL code into native code.
- **Code Manager :**
It manages the code at run time.
- **Garbage Collector :**
It manages the memory. Collect all unused objects and de-allocate them to reduce memory.
- **Thread Support**
- **Exception Handler :**
It handles exceptions at run time.

❖ Common Type System (CTS) :

- Common Type System (CTS) describes the datatypes that can be used by managed code.
- CTS defines how these types are declared, used and managed in the runtime.
- It facilitates cross-language integration, type safety, and high-performance code execution.
- The rules defined in CTS can be used to define your own classes and values.
- CTS can support two types of category.

1. Value Type

2. Reference Type

- **Value Types :**

Contain the values that need to be stored directly on the stack or allocated inline in a structure. They can be built-in (standard primitive types), user-defined (defined in source code) or enumerations (sets of enumerated values that are represented by labels but stored as a numeric type).

- **Example :**

```
int a;
```

- **Reference Types :**

- Store a reference to the value's memory address and are allocated on the heap. Reference types can be any of the pointer types, interface types or self-describing types (arrays and class types such as user-defined classes, boxed value types and delegates).

- **Example :**

```
student obj = new student();
```

Here, student is a class and obj is object.

❖ Common Language Specification (CLS):

- CLS is a sub set of CTS.
- The Common Language Specification (CLS) is a fundamental set of language features supported by the Common Language Runtime (CLR) of the .NET Framework.
- It means that all of the rules in CTS also apply to the CLS.
- CLS is a part of the specifications of the .NET Framework.
- CLS was designed to support language constructs commonly used by developers and to produce verifiable code, which allows all CLS-compliant languages to ensure the type safety of code.
- CLS includes features common to many object-oriented programming languages.

❖ Framework Class Library (FCL) :

- Framework Class Library is the collection of classes, namespaces, interfaces and value types that are used for .NET applications.
- FCL is provide various types of functionality to the .NET code.
- In C <conio.h>,<stdio.h> are header files, they are added program to use inbuilt functions.

❖ Base Class Library (BCL) :

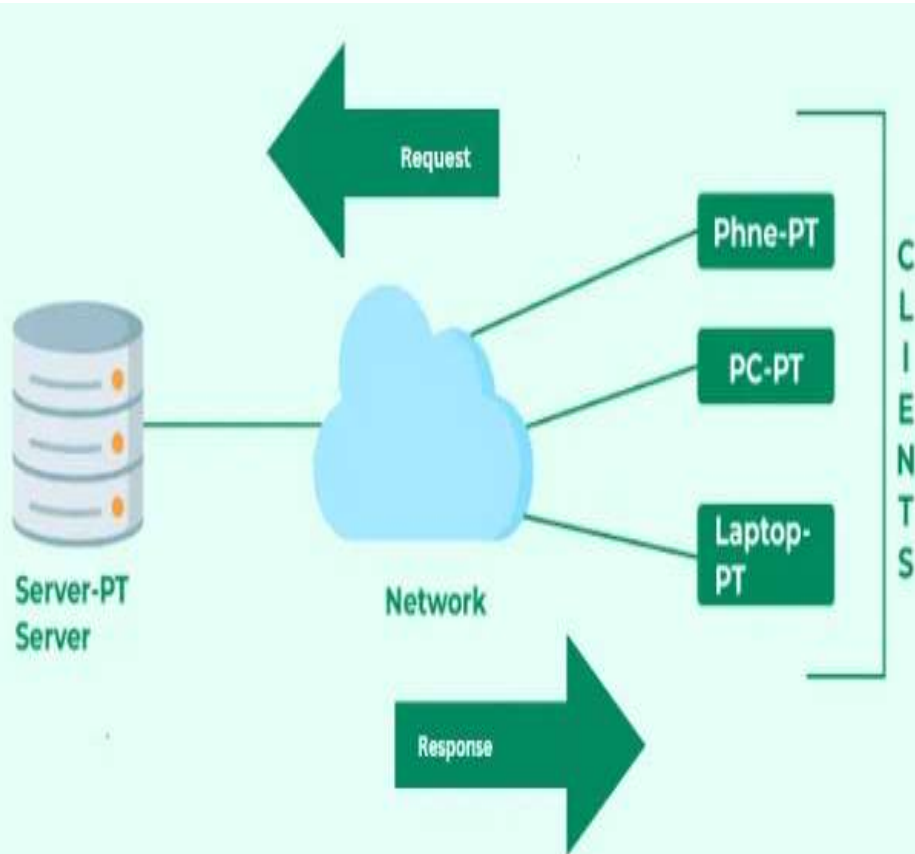
- BCL is Subset of FCL.
- Base Class Library is the sub part of the Framework that provides library support to Common Language Runtime to work properly.
- It includes the System namespace and core types of the .NET framework.
- BCL divides into two parts:
- **User defined class library :**
- **Assemblies** - It is the collection of small parts of deployment an application's part. It contains either the DLL (Dynamic Link Library) or exe (Executable) file.
- **Predefined class library :**
- **Namespace** - It is the collection of predefined class and method that present in .Net.
- In other languages such as, C we used header files, in java we used package
- similarly we used "using system" in .NET, where using is a keyword and system is a namespace.

Client Server Architecture :

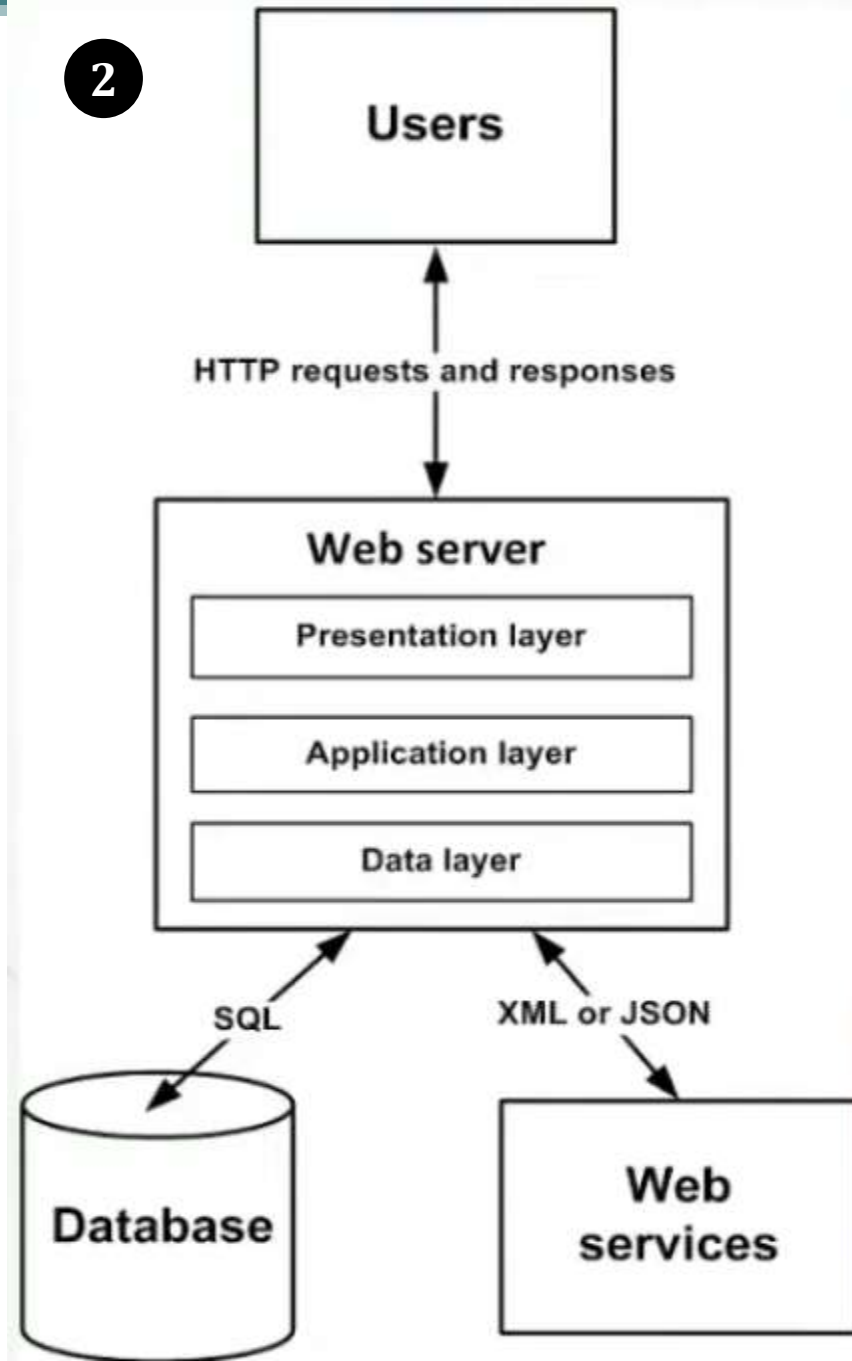
- **Client** – The client can be any computer that requests something from the server. For example – visiting any website we request the webpage from its domain.
- **Server** – a **Server** is a remote computer that provides information (data) or access to particular services, the Server is the computer that is designed to serve the requests to the client. For the same example as above, the client asks for the webpage then the server responds with the webpage to the client.
- In the **client-server architecture**, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and delivers the data packets requested back to the client.
- Clients do not share any of their resources.
- Examples of the Client-Server Model are Email, World Wide Web...

- Client-server architecture, alternatively called a client-server model, is a network application that breaks down tasks and workloads between clients and servers that reside on the same system or are linked by a computer network.
- It is the Client requesting something and the server serving it as long as it is in the database.
- Let's see how client server architecture is manage :
 1. First, the client sends their request via a network-enabled device
 2. Then, the network server accepts and processes the user request
 3. Finally, the server delivers the reply to the client

1

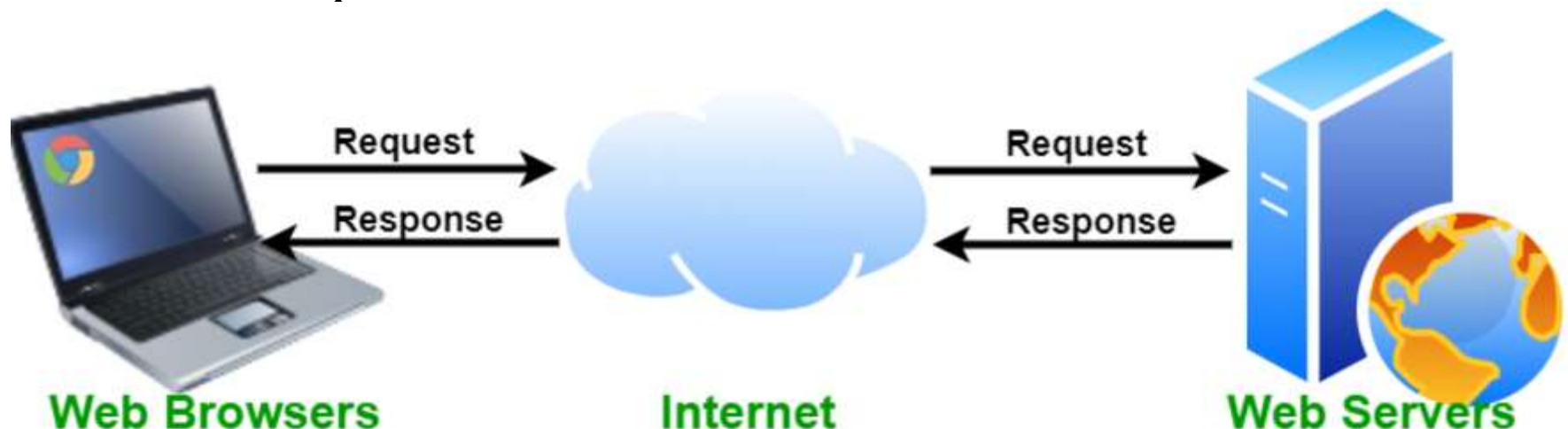
*Client Server Model*

2



Application Web Server :

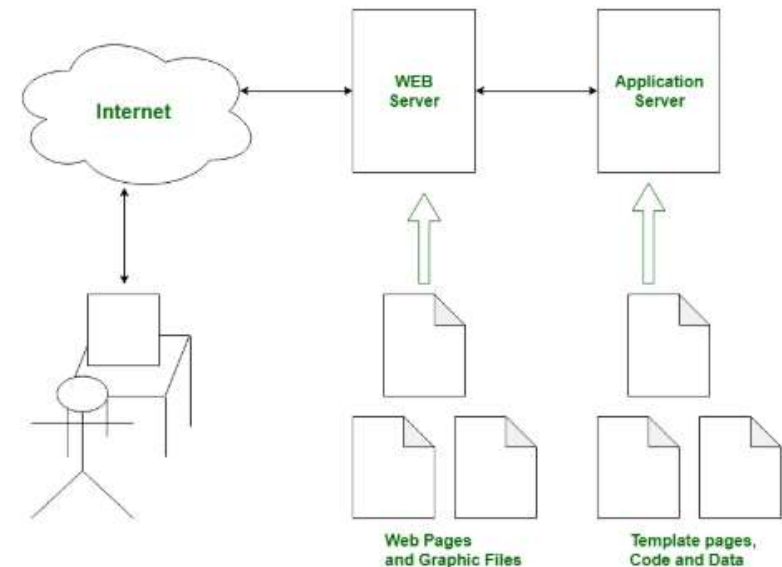
- A **Web Server** is defined as a server which accepts a request for data and sends the relevant document in return.
- In other words, it is a computer program that accepts a request for a specific document and sends it to the client machine.
- Web servers are designed to serve HTTP content to the client computer.
- In most cases, the web servers are the integral parts of the application servers.
- Web servers accept the HTTP requests and interpret them to serve the requested content.



- There are many web servers available in the market both free and paid.
- **Internet Information Services (IIS)**
- **Apache HTTP server**
- **Lighttpd**
- **Jigsaw Server**
- **Sun Java System**

Application Server :

- An Application Server provides a more comprehensive environment for running enterprise applications.
- It includes both a web container and an EJB (Enterprise JavaBeans) container, supporting a wide range of applications, including dynamic content and complex business operations.
- Application servers handle not just HTTP, but also other protocols like RMI (Remote Method Invocation) and RPC (Remote Procedure Calls), making them suitable for hosting both the logic and user interface of applications.
- **Examples of Application Server:**
 - WebLogic
 - JBoss
 - WebSphere
 - GlassFish



Web Server	Application Server
Web server encompasses web container only.	While application server encompasses Web container as well as EJB container .
Web server is useful or fitted for static content .	Whereas application server is fitted for dynamic content .
Web server consumes or utilizes less resources.	While application server utilize more resources.
Web servers arrange the run environment for web applications .	While application servers arrange the run environment for enterprises applications.
In web servers, multithreading is supported.	While in application server, multithreading is not supported.
Web server's capacity is lower than application server.	While application server's capacity is higher than web server.
In web server, HTML and HTTP protocols are used.	While in this, GUI as well as HTTP and RPC/RMI protocols are used.
Processes that are not resource-intensive are supported.	Processes that are resource-intensive are supported.
Transactions and connection pooling is not supported.	Transactions and connection pooling is supported.
Web Server examples are Apache HTTP Server , Nginx .	Application Servers example are JBoss , Glassfish.

How to install IIS ?

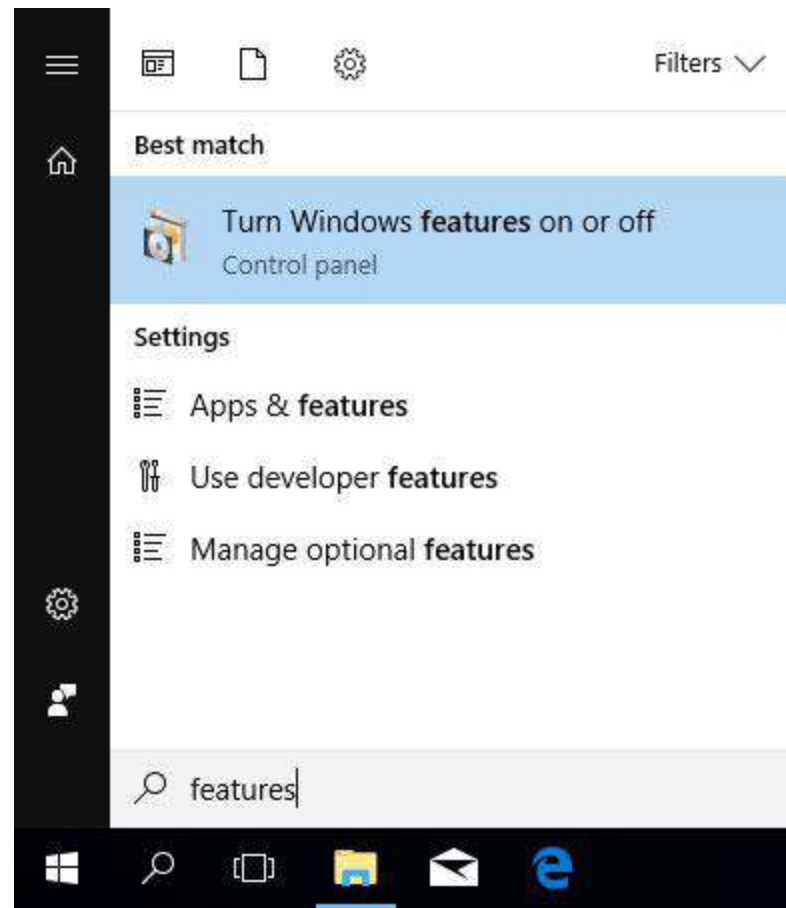
- IIS web server runs on the Microsoft .NET platform on the Windows OS.
- IIS or Internet Information Services is available in most editions of Windows, though disabled by default.

- **Step-1**

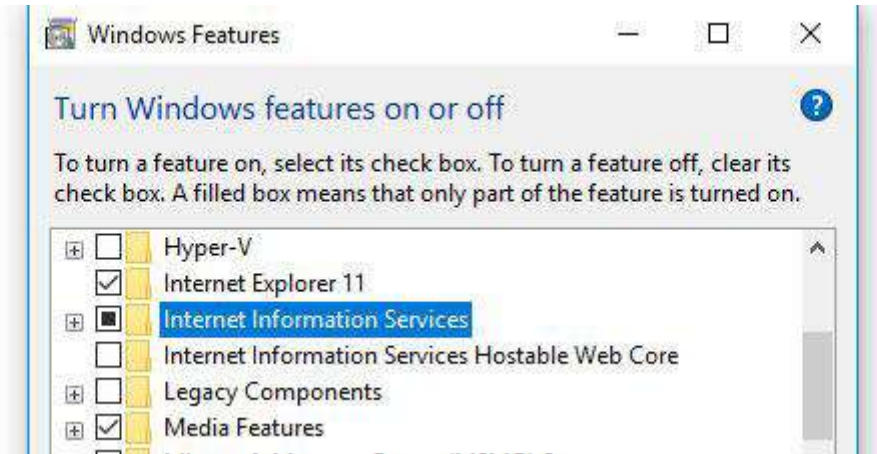
Open the Start menu.

- **Step-2**

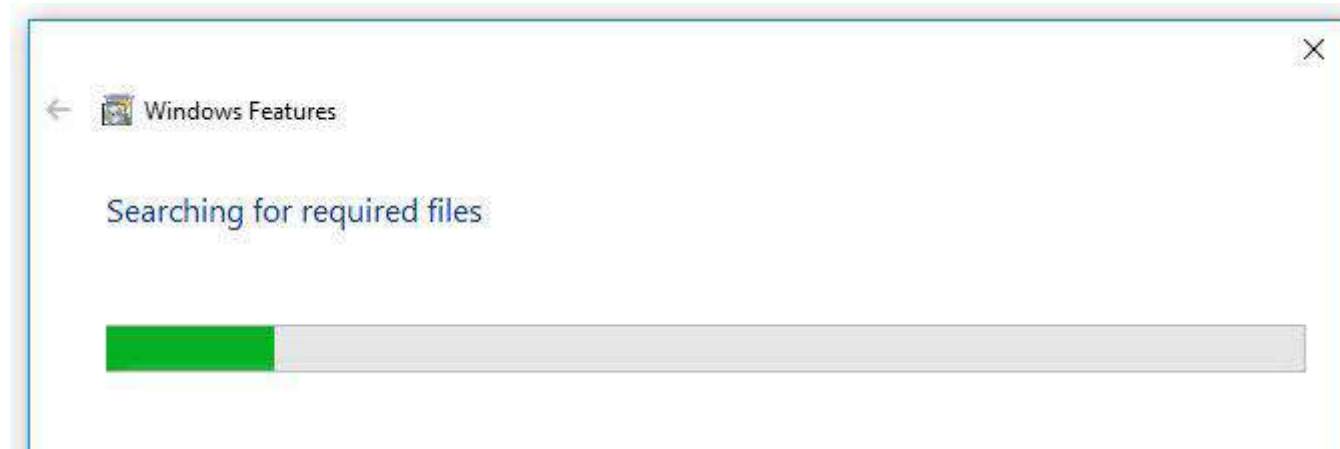
Type features and select Turn Windows features on or off



- **Step-3**
- Tick the Internet Information Services checkbox and hit OK.

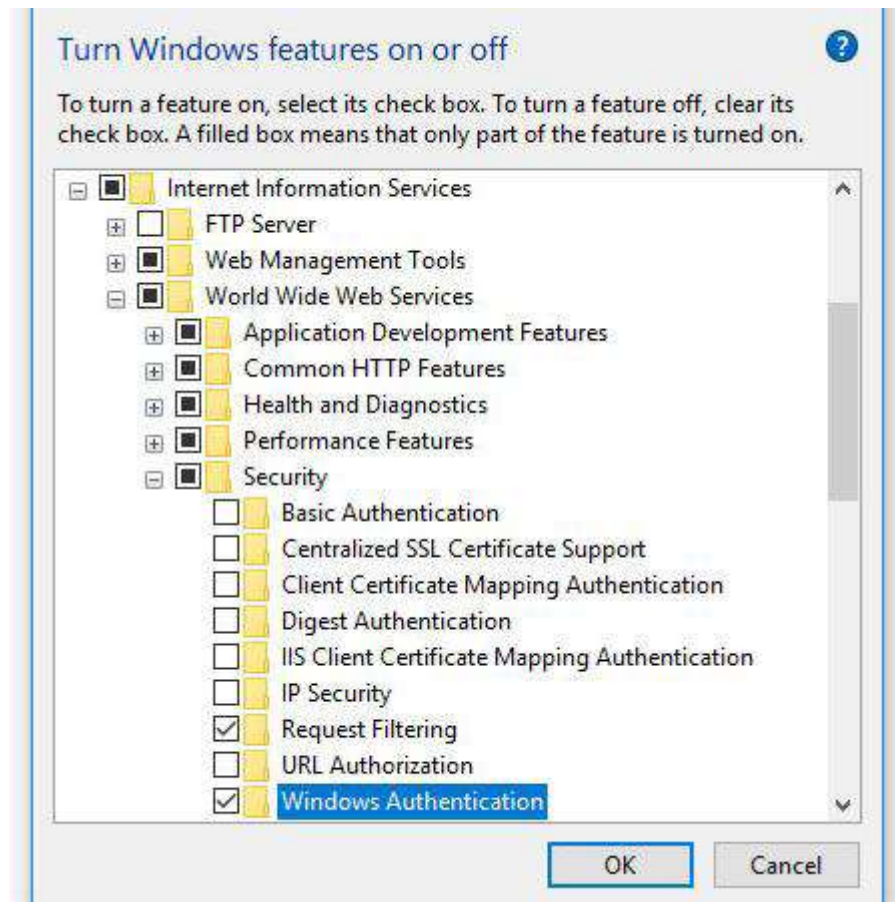


- **Step-4**
- Wait for the installation to complete and hit Close.

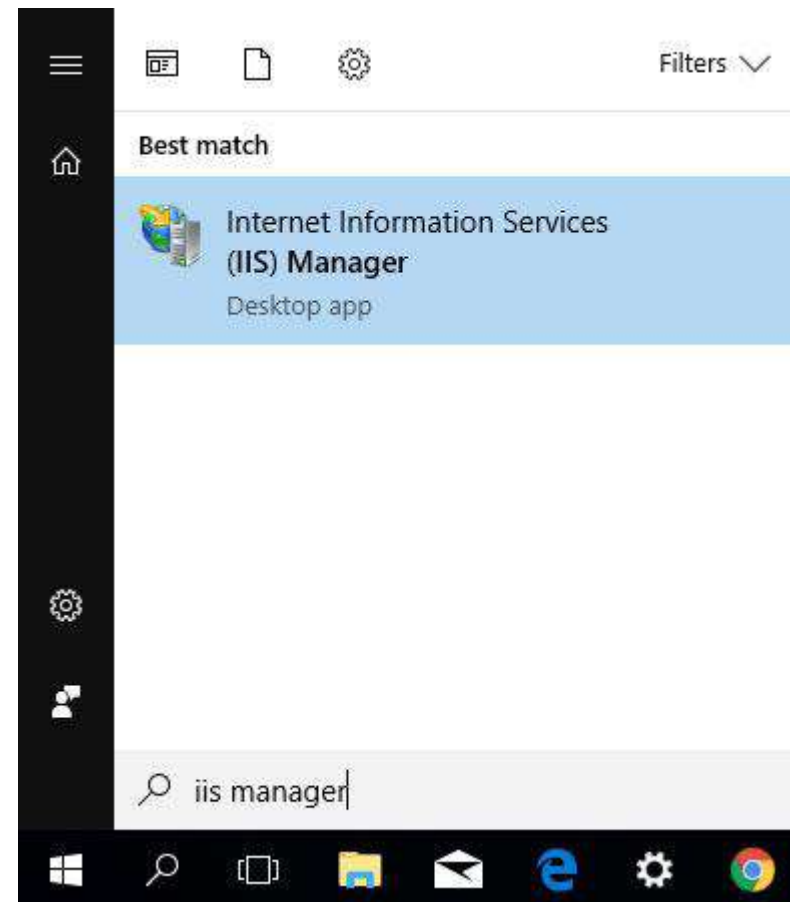


- **Step-5**

If you plan on using integrated Windows authentication in Lansweeper or another website hosted in IIS, tick the Windows Authentication option under Internet Information Services\World Wide Web Services\Security as well and hit OK.



- **Step-6**
- Now IIS manager is ready to work with you.



FILE TYPES USED IN ASP.NET

- .aspx
- .ascx
- .asmx
- .ashx
- .cs
- Web.Config
- Global.asax
- .master

- **.aspx:**

It is basic file structure of web pages that you create In ASP.NET. They contain the GUI(Graphical User Interface) for any web pages. They are also known as Presentation Logic File.

- **.ascx:**

Active server control extension.

It is file that is used to create custom control.

With this you can create custom controls(User Control).

Once created, then this control you can reuse the same control again and again as per our requirement.

- **.asmx:**

Active Server Method File.

It is file that is used to create Web Service.

- **Web.Config:**

It is a XML based configuration file.

- **.ashx:**

ASP.NET Web Handler file.

They are known as Handler file.

It contains a reference to other web pages.

- **.cs:**

C# code.

It can also be .vb, .js form.

VB code it is of the form .vb for javascript it of the form .js.

- **Global.asax:**

It is global application file.

It is used define global variable and global events for given application.

- **.master :**

It refers to a master page that defines the layout of web page in a web application.

It resides Application root or subdirectory.

Types of controls in Asp.NET :

- There are two types of controls available in ASP.NET.
- They are known as:

1. Server Controls

2. Client Controls

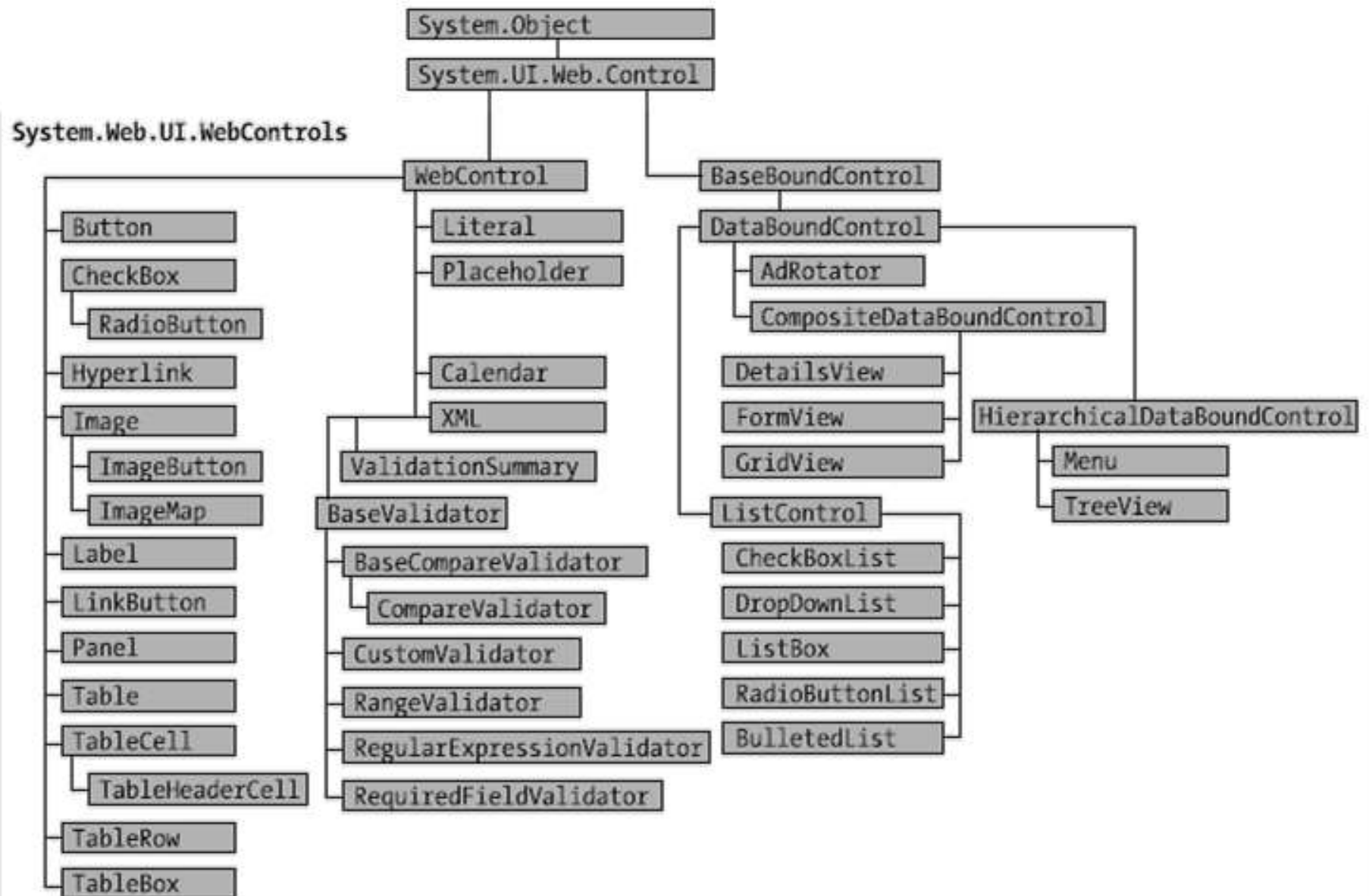
1. Server Controls:

- These are the controls which execute on server side. Most of the controls in ASP.NET controls works or can work as Server Controls. The controls which execute on server side have two very important properties.
- All server controls has one very important attribute called “runat”, which is written as **runat=“server”**.
- **Example :**
- `<asp:TextBox Id="TextBox1" runat="server"></Textbox>`

2. Client Controls :

- Client controls are controls which don't execute on server side. Client controls don't expose the functionality of event handling because event handling is generally done on the control which executes on server side.
- HTML controls are generally called as Client control but they can be converted to Server Control after adding ID and runat attribute to the code.
- The ASP.NET Framework (version 3.5) contains over 70 controls.
- contains over 70 controls.
- **Example :**
- `<input type="text" Id="TextBox1" />`
- These controls can be divided into following groups:

1. **Standard Controls** : The standard controls enable you to render standard form elements such as buttons, input fields, and labels.
2. **Validation Controls** : The validation controls enable you to validate form data before you submit the data to the server. For example, you can use a RequiredFieldValidator control to check whether a user entered a value for a required input field.
3. **Rich Controls** : The rich controls enable you to render things such as calendars, file upload buttons, rotating banner advertisements, and multi-step wizards.
4. **Data Controls** : The data controls enable you to work with data such as database data. For example, you can use these controls to submit new records to a database table or display a list of database records.
5. **Navigation Controls** : The navigation controls enable you to display standard navigation elements such as menus, tree views, and bread crumb trails.
6. **Login Controls** : The login controls enable you to display login, change password, and registration forms.
7. **Web Part Controls** : The Web Part controls enable you to build personalizable portal applications.
8. **HTML Controls** : The HTML controls enable you to convert any HTML tag into a server-side control.
9. **ASP.NET Mobile Controls** : These are the controls which are somewhat similar to Web Controls but these are customized to support mobile clients. Mobile clients can be PDAs, Smart Phones, etc.



Page Architecture :

- The basic components of ASP.NET page are...
 1. Page Directive
 2. Code Declaration Block
 3. ASP.NET controls
 4. Server Side Comments
 5. Server Side Include Directives

1. Page Directive :

- Page directive is used to specify the default programming language for a page.
- Page directives can also be used for tracing page and import directives.

- **Syntax :**

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

1. **Language** : indicates default programming language of the page.
2. **AutoEventWireup** : indicates that the events are enabled on this page.
3. **CodeFile** : indicates the code file associated with this page.
4. **Inherits** : indicates the class file inherited by this page.

- **2. Code Declaration Block :**

- A code declaration block contains all the logic of ASP.NET page and all the global variable declarations, sub routine and functions.
- It must appear with in a **<script Runat="server"> tag.**

- **3. ASP.NET Controls :**

- ASP.NET controls can be used with the text and HTML content of a page.
- The only requirement is that controls should appear with in a **<form runat="server"> tag.**
- All the controls should behave a tag runat="server", which makes the control run and appear on the web page.

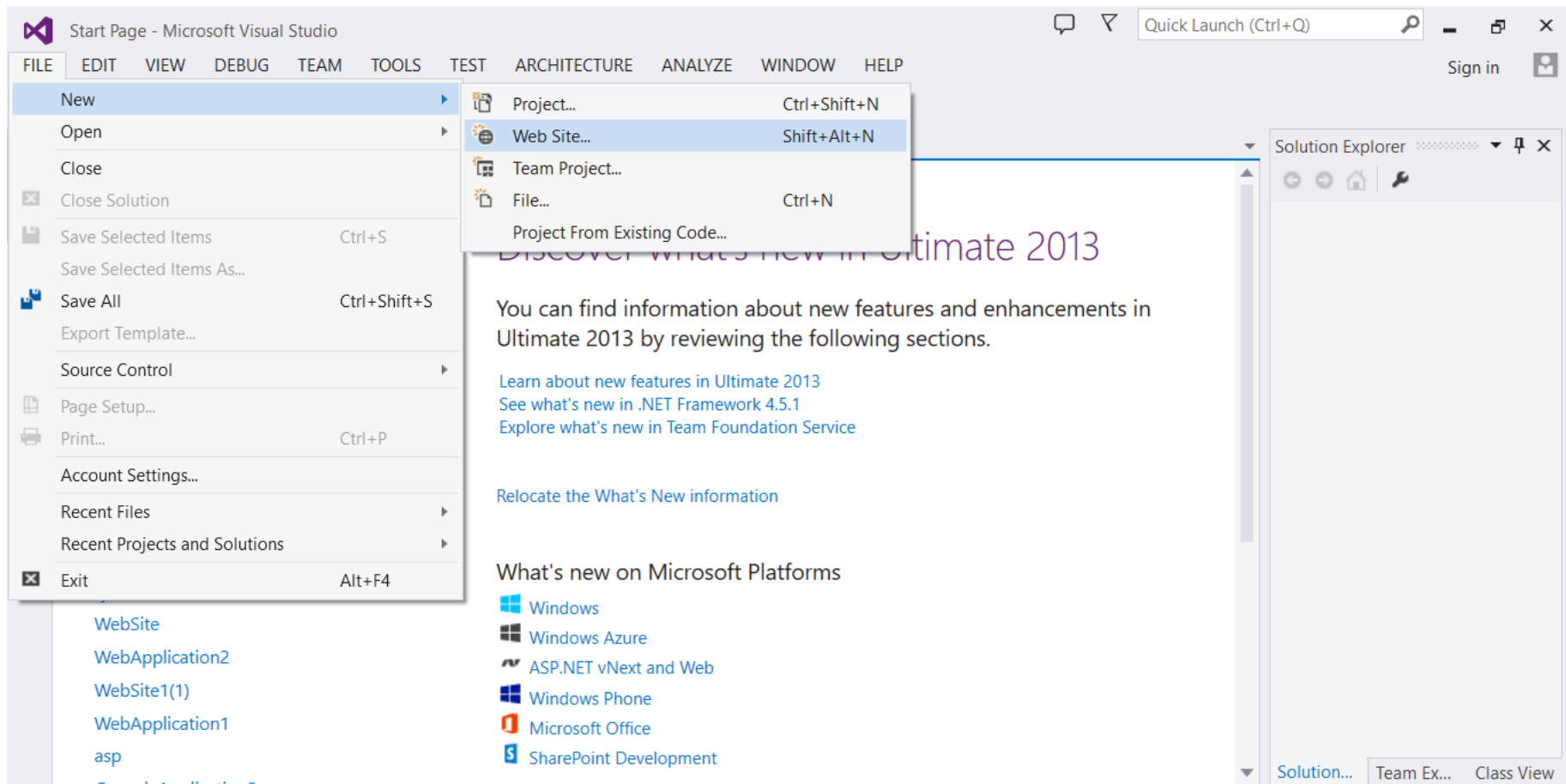
4. Server Side Comments :

- To provide server side comments this block is used.
- This statement is marked with the characters
- **<%--characters--%>.**
- They can also be used for documentation purpose.
- **Server Side Include Directives :**
- To include a file in ASP.NET page the Server Side Include Directives are used. The basic syntax for the directive is :
- **<!--#INCLUDE file="includefile.aspx" -->**
- Here, the File tag indicates the name of the file be included on the Page.

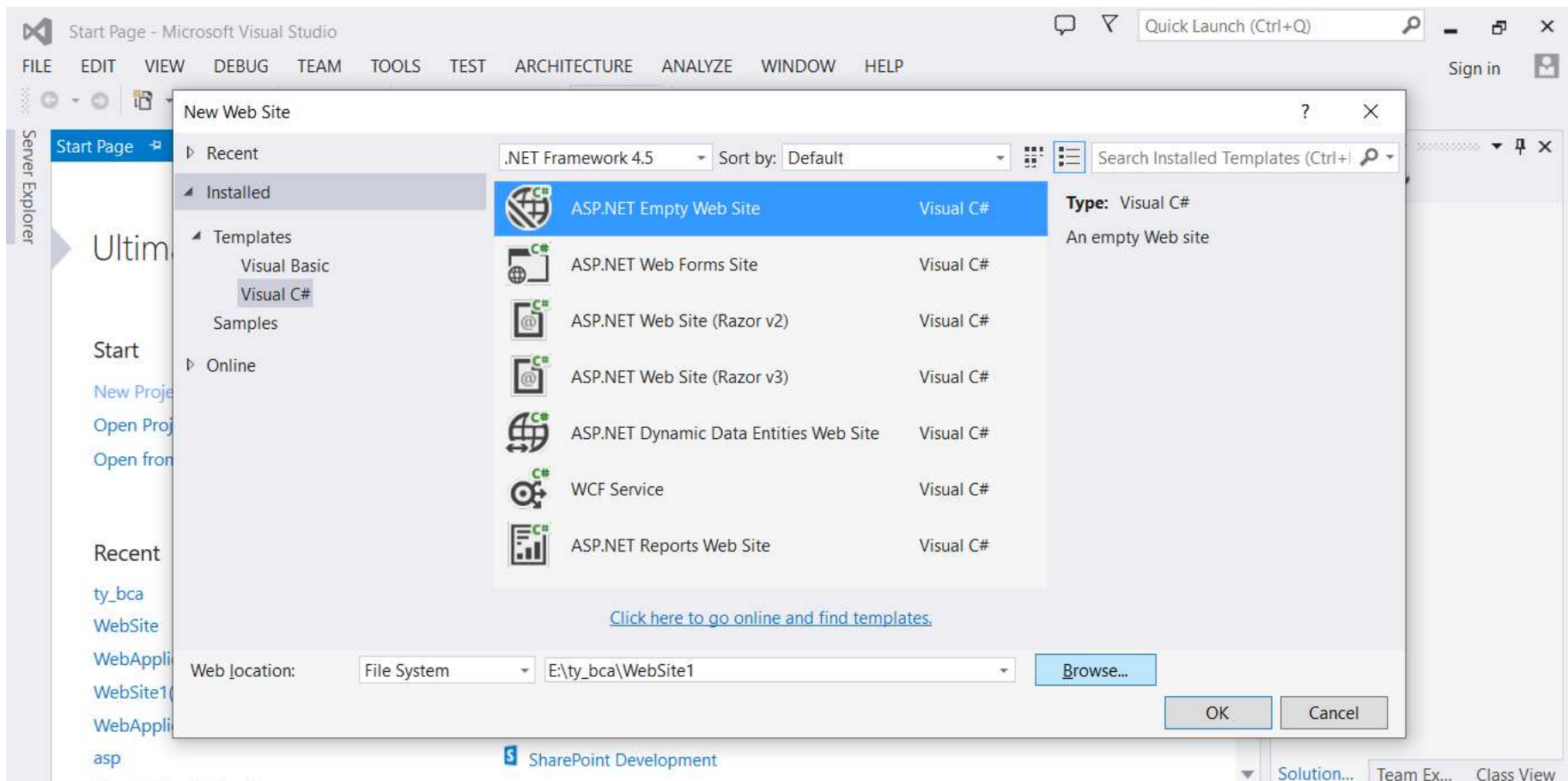
Steps for Creating a Web Application :

- In Visual Studio...
- **Step-1 :**
Goto file menu
 - **Select New**
 - **Website (Shift + Alt + N)**
- **Step-2 :**
Step 1 will display a New Website Dialog Box.
Select **ASP.Net Web** Site option, write your selected web **name** and press **ok**
- **Step-3 :**
It will display a screen with **default.aspx file** source view.
- **Step-4 :**
We can put controls by coding in Source view. OR
Select Design view, now use the ToolBox and place some controls on to default.aspx.
- **Step-5 :**
Now double click on any of control.
It will display a code window with file name, **Default.aspx.cs (Note : For C#)**
now write Application Logic or Code in code window.
- **Step-6 :**
Now execute the application by,
licking on the Run Button from standard tools
Select Debug Menu
 - **Select : Start Debugging**Press F5 to execute the application

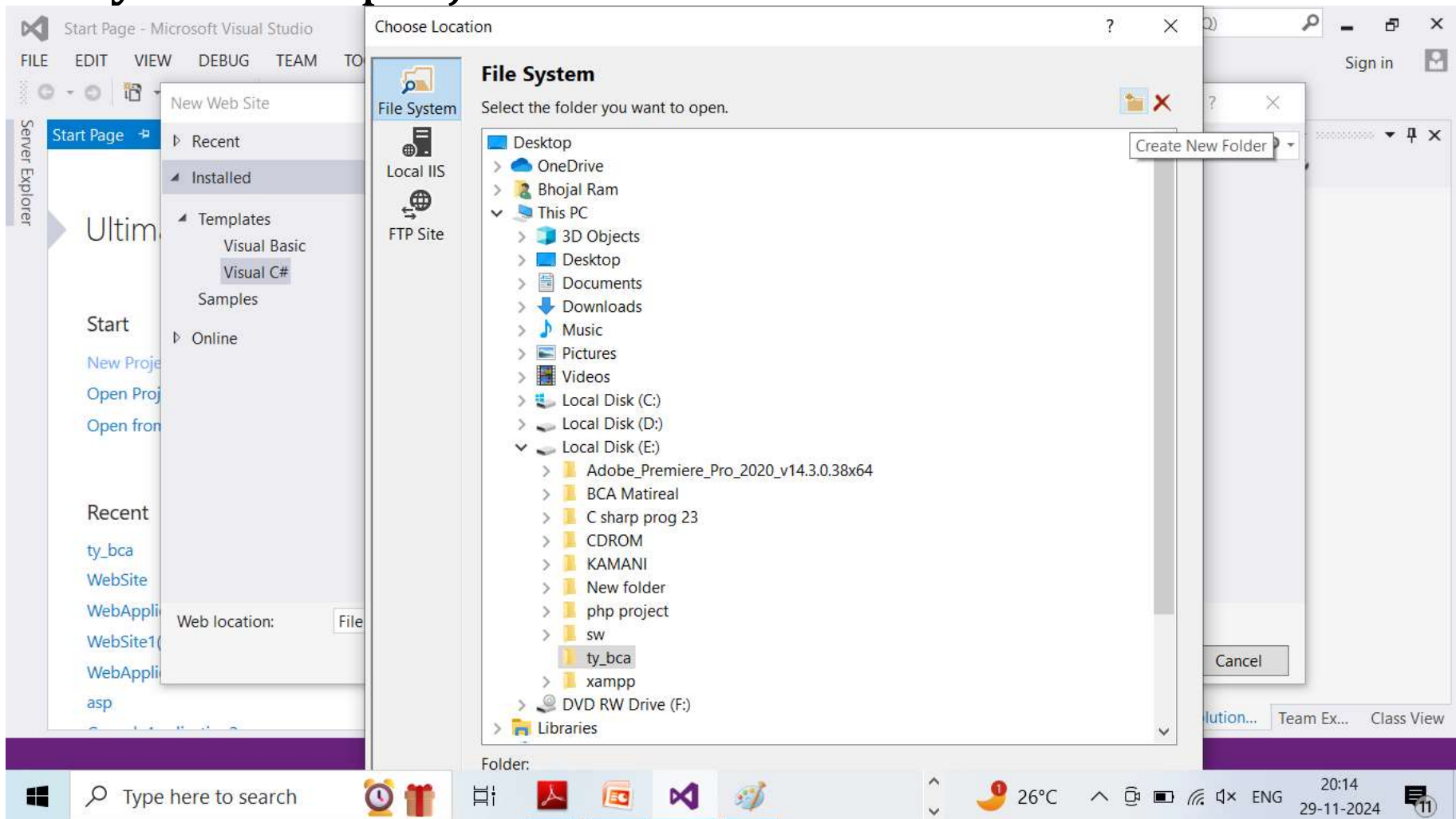
- **Step 1:**
- Open Visual Studio.
- Goto FILE Menu → new → Web Site (Shift+Alt+N)



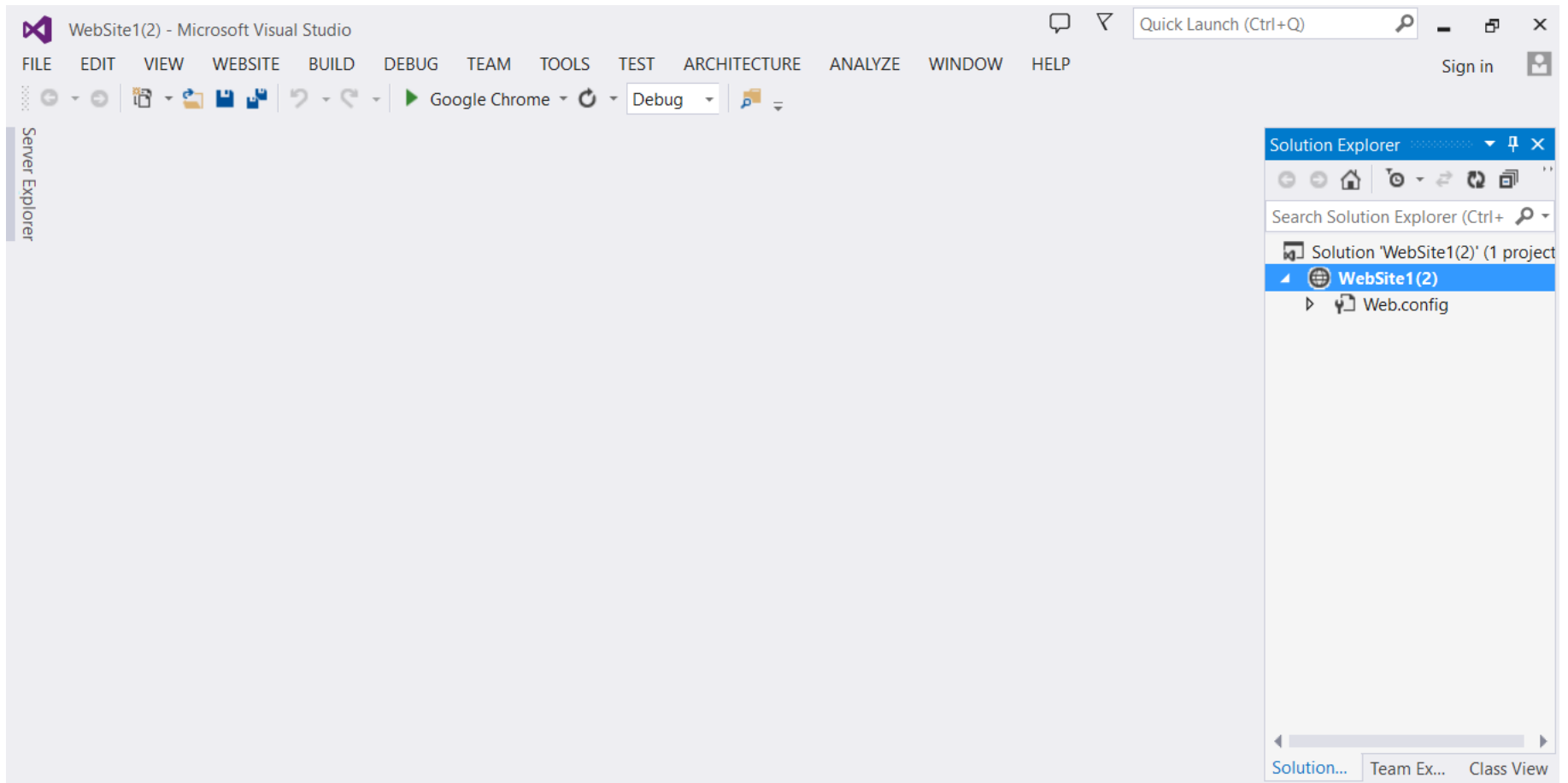
- **Step 2 :**
- Now , Select Visual C# → ASP.NET Empty Web Site
To Select your location click on browse.



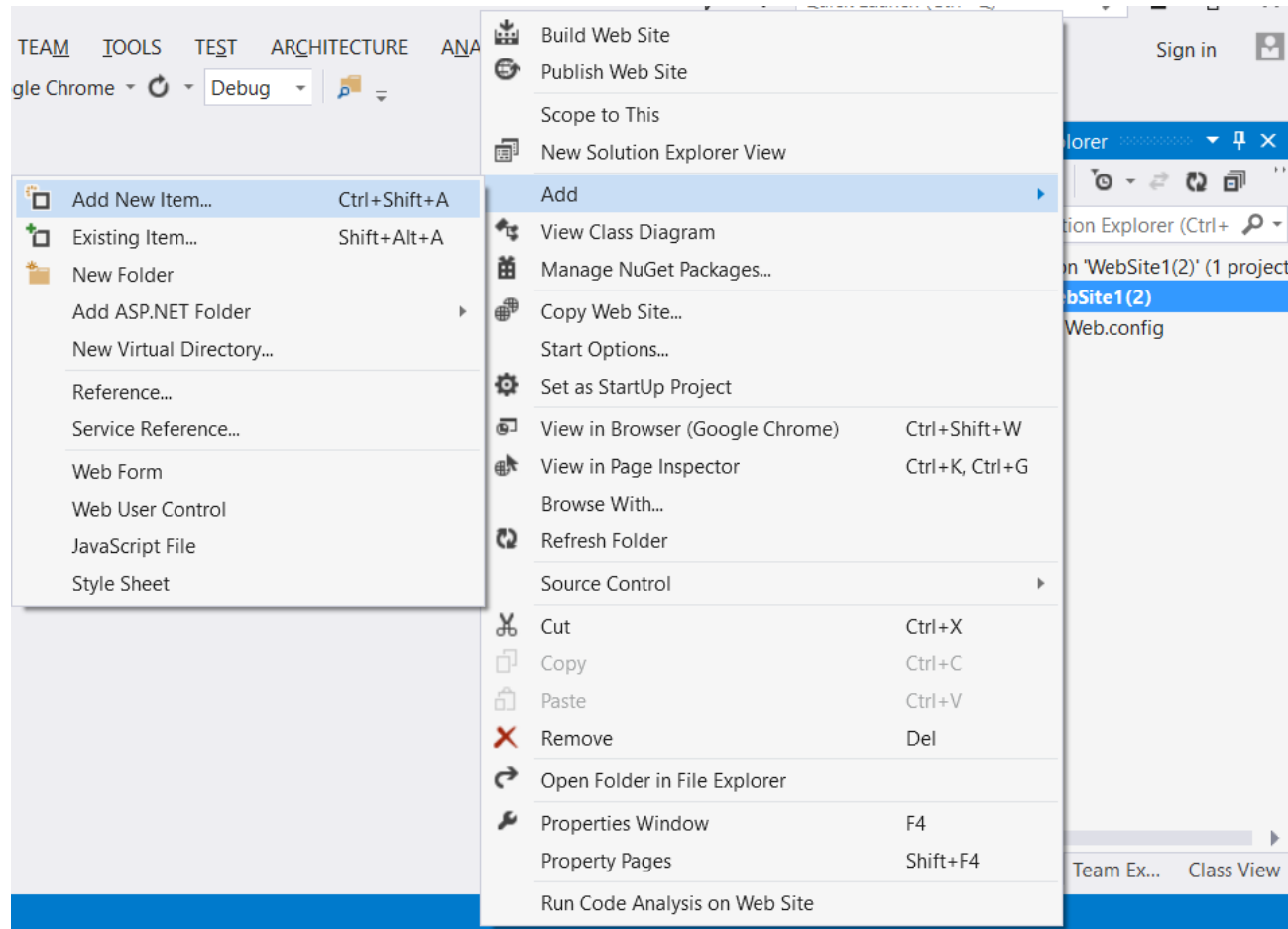
- **Step 3:**
- Select your location and create new folder to create your first project.



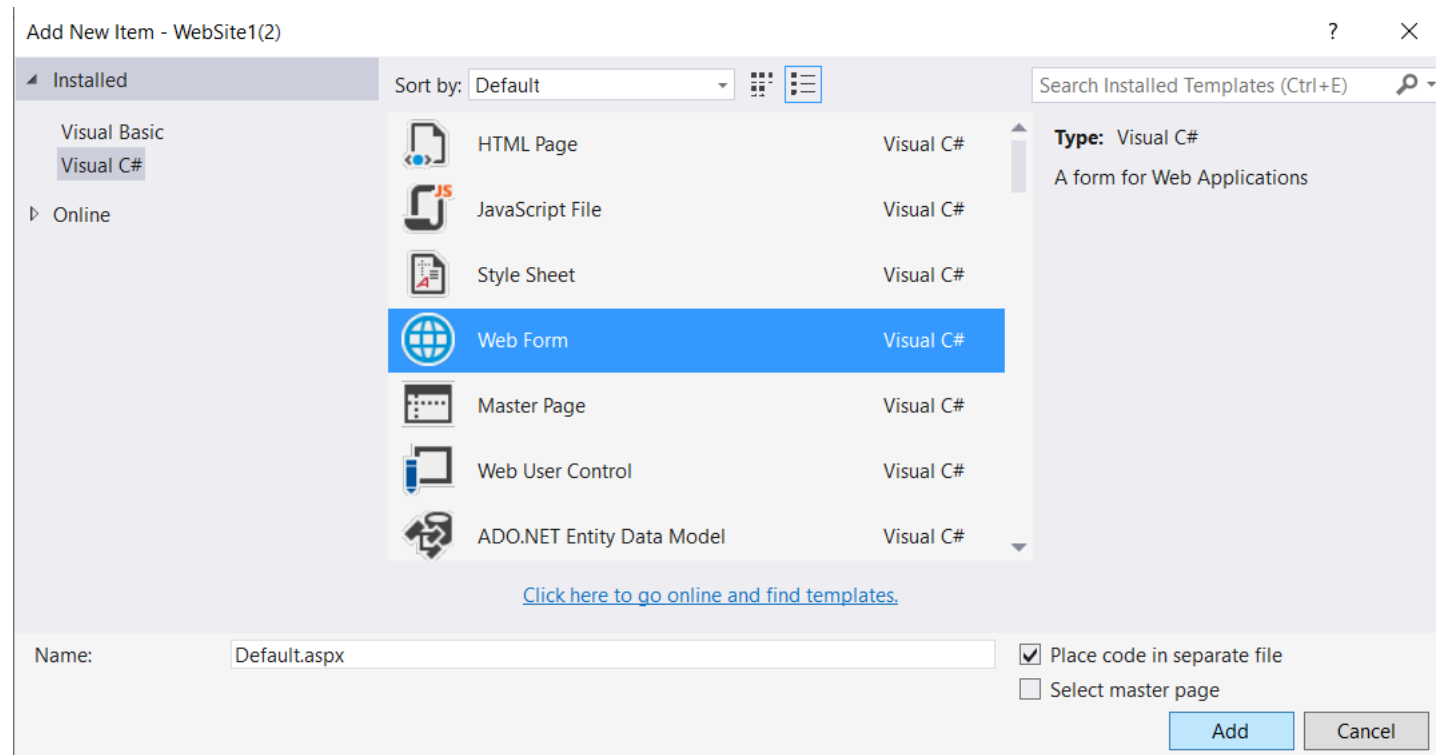
- **Step 4:**
- Now, your site is ready



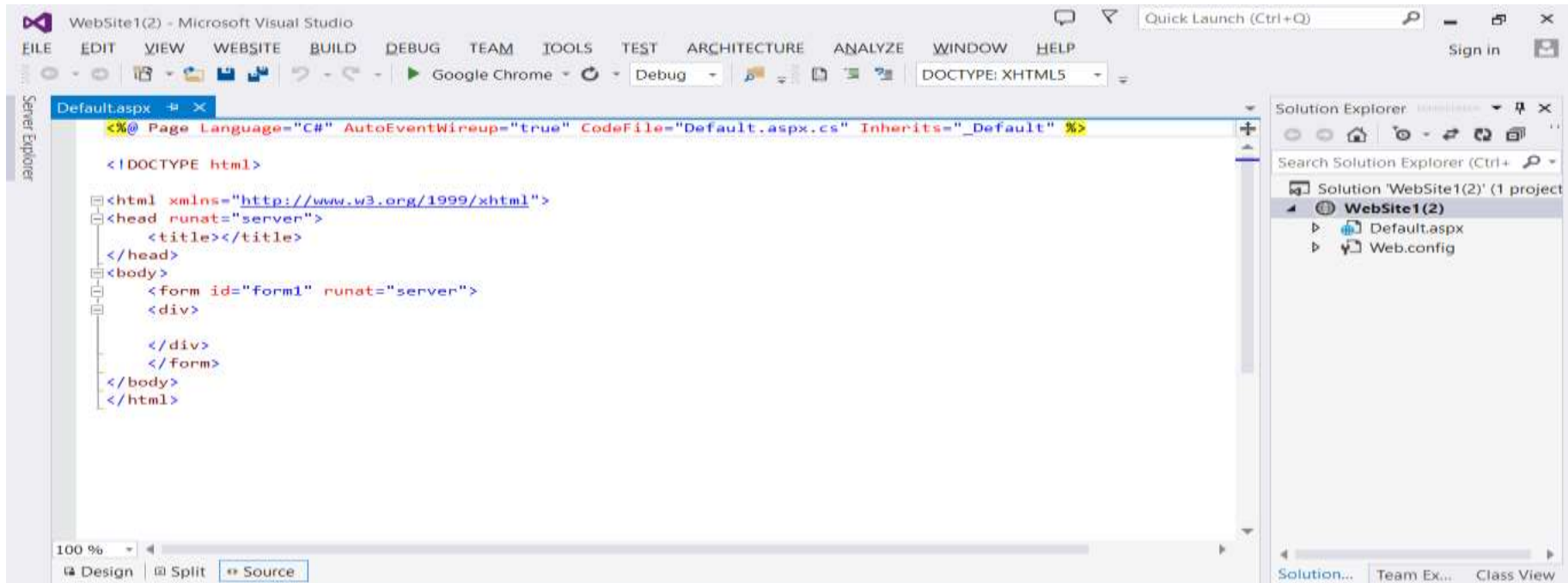
- **Step 5 :**
- Let's add 1st web form :
- Right click on your project name
- Add New Item(Ctrl+Shift+A).



- **Step 6 :**
- Select C#.
- Now click on web form.
- Write name of your web form.
- Click on add.



- **Step 7 :**
- Web form is ready to use.



- **Step 8 :**
- To debug project press F5
- Or run project
- You can also select browser.

What is Web Form :

- A web page that contains <form> tag called 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#.

```
<!DOCTYPE html>
```

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

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

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

```
</head>
```

```
<body>
```

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

```
    <h1>Welcome to TYBCA</h1>
```

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

```
    <asp:Button runat="server" />
```

```
  </form>
```

```
</body>
```

```
</html>
```

FileUpload Control :

- ASP.NET's FileUpload is an input controller used to upload files to a server.
- It appears on the screen with a browse button and opens up a dialogue box to choose a file or multiple files to upload from the local storage to the server.
- This is a server-side control provided by ASP.NET.
- The fileupload control in asp.net used to upload any file like image, document file, zip file, etc... to asp.net website.

<asp:FileUpload ID="flu" runat="server" />

- **Server.MapPath(path)** : MapPath is a method.
- **path:** It stores a string value that defines the relative or virtual path to map to a physical directory.

If the path starts with either a forward slash(/) or backward slash(\) the MapPath Method returns a path as if the path is a full virtual path.

If the path doesn't start with a slash, the MapPath Method returns a path relative to a directory of the .asp file being processed.

Some important Properties :

Properties	Description
FileBytes	Returns an array of the bytes in a file to be uploaded.
FileContent	Returns the stream object pointing to the file to be uploaded.
FileName	Returns the name of the file to be uploaded.
HasFile	Specifies whether the control has a file to upload.
PostedFile	Returns a reference to the uploaded file.

Property	Description
AccessKey	It is used to set keyboard shortcut for the control.
TabIndex	The tab order of the control.
BackColor	It is used to set background color of the control.
BorderColor	It is used to set border color of the control.
BorderWidth	It is used to set width of border of the control.
Font	It is used to set font for the control text.
ForeColor	It is used to set color of the control text.
Text	It is used to set text to be shown for the control.
ToolTip	It displays the text when mouse is over the control.
Visible	To set visibility of control on the form.
Height	It is used to set height of the control.
Width	It is used to set width of the control.
AllowMultiple	It is used to allow upload multiple files by setting true or false.

File.aspx :

```
<!--File upload-->
<form id="form1" runat="server">
<div>
    <asp:FileUpload ID="f_up" runat="server" />
</div>
<div>
    <asp:Button runat="server" ID="btn_save" Text="upload"
OnClick="btn_save_Click" />
</div>
</form>
```

File.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 _Default : System.Web.UI.Page
{
    protected void btn_save_Click(object sender, EventArgs e)
    {
        if(flu.HasFile)
        {
            string cont = f_up.FileContent.ToString( ); //if file has a content
            string nm= f_up.FileName.ToString( );
            Response.Write(nm);
            Response.Write("\n"+cont);
            flu.SaveAs(Server.MapPath("file_folder/" + f_up.FileName));

        }
        else
            Response.Write("No file found"); }
}
```

Validation / Validator Controls :

- Validation is an important process to give valid input data with proper format from user to web form then give meaning full output.
- As we know all web forms must be validated properly.
- In ASP.Net we can validate form manually by doing coding or use readymade validation control from toolbox.
- Validations can be performed on the server side or on the client side (web browser).
- The user input validation take place on the Server Side during a post back session is called Server Side Validation and the user input validation take place on the Client Side (web browser) is called Client Side Validation.
- Client Side Validation does not require a **postback**.
- If the user request requires server resources to validate the user input, you should use Server Side Validation & validated using one of server side scripting languages such as ASP.Net, PHP etc..
- If the user request does not require any server resources to validate the input , you can use Client Side Validation & validation is done on the browser side using script languages such as JavaScript, VBScript or HTML5 attributes..
- For example, if the user enter an invalid email format...

- **TYPES OF VALIDATION**
 - There are two types of validation in ASP.Net
-
1. Client-Side Validation
 2. Server-Side Validation
 1. RequiredField Validator
 2. Range Validator
 3. CompareField Validator
 4. RegularExpression Validator
 5. Custom Validator
 6. ValidationSummary

- **Client-Side Validation :** Client side validation is done on the client browser.
Client side validation is faster than sever side validation.
We use JavaScript, JQuery for client side validation.
- ClientSidevalidationmeansthatcheckingofdataformatorvalidationisdoneatClientSide.
- Client side validation is done by browser itself.
- If all the action of user goes to server side, then the processing would be done on server side and then revert back the response.
- To send request each time and got response each time then it would take too much time to get the page processed.
- To avoid long time processing we can used client side validation.
- The client side validation gives faster response to Client.
- The client side validation is written in a script section.
- In ASP.NET several controls are available to carryout Client Side Validation.

- **Server-Side Validation :** Server side validation is done on the web server.
We use custom logic code for server side validation
- Server side validation means the checking of data format or validation is carried out at Server Side.
- This is more safe then client side validation.
- ASP.NET provides some powerful controls to maintain Server Side Validations.
- The Controls that are used to validate the data against some input format are known as Validation Controls.
- If the data does not pass the validation, it will display an error message to the user and user would not be able to submit the data.

Types of Validators :

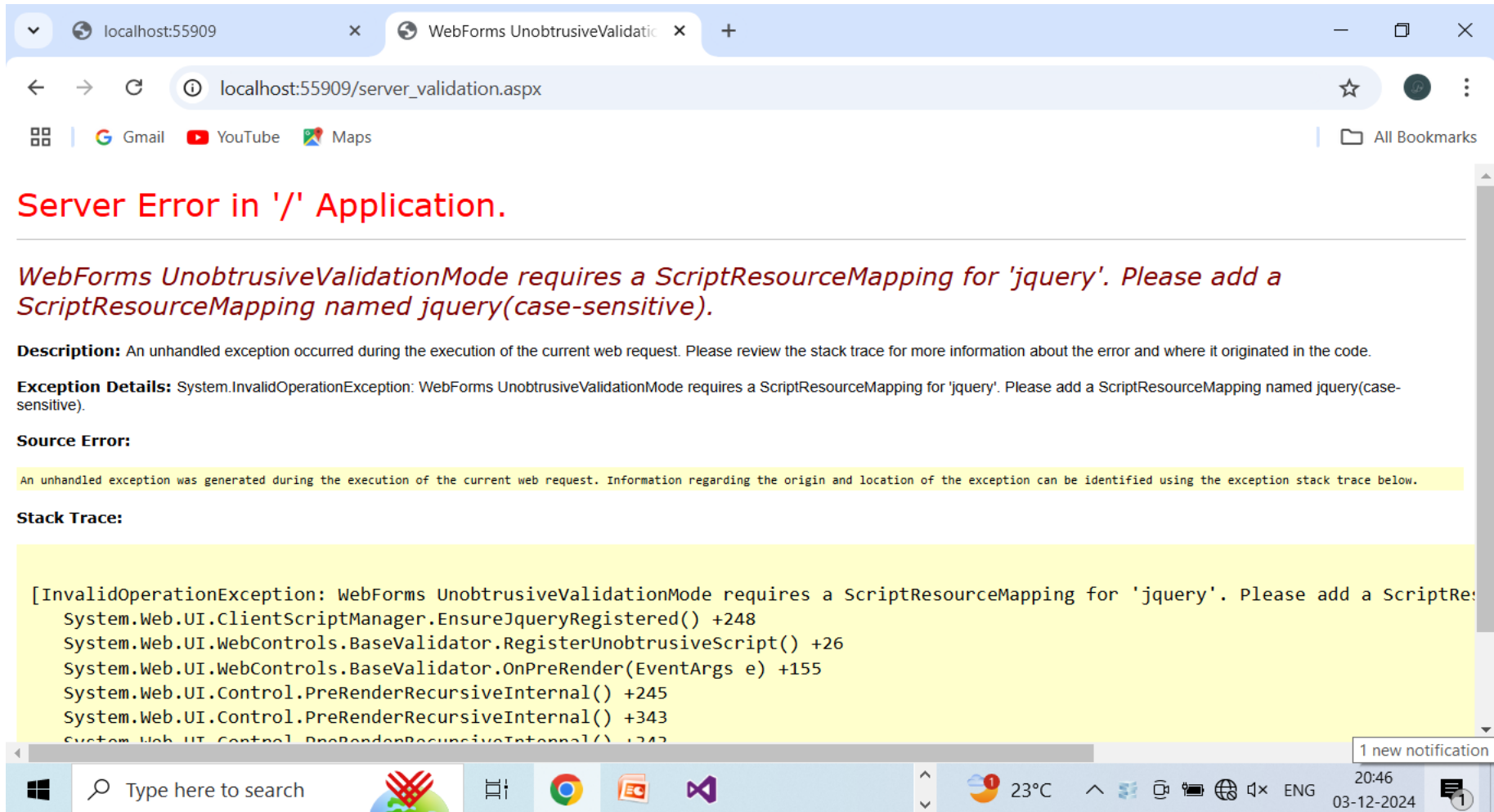
1. RequiredField Validator
2. Range Validator
3. CompareField Validator
4. RegularExpression Validator
5. Custom Validator
6. ValidationSummary

❖ BaseValidator CLASS :

- The validation control classes are inherited from the BaseValidator class hence they inherit its properties and methods.
- it would help to take a look at the properties and the methods of this base class, which are common for all the validation controls:

Members	Description
ControlToValidate	Indicates the input control to validate.
Display	Indicates how the error message is shown.
EnableClientScript	Indicates whether client side validation will take.
Enabled	Enables or disables the validator.
ErrorMessage	Indicates error string.
Text	Error text to be shown if validation fails.
IsValid	Indicates whether the value of the control is valid.
SetFocusOnError	It indicates whether in case of an invalid control, the focus should switch to the related input control.
ValidationGroup	The logical group of multiple validators, where this control belongs.
Validate()	This method revalidates the control and updates the IsValid property.

- For any validator control it will throw following error let's learn how to handle it.



The screenshot shows a web browser window with the address bar displaying `localhost:55909/server_validation.aspx`. The page content displays a red error message: *WebForms UnobtrusiveValidationMode requires a ScriptResourceMapping for 'jquery'. Please add a ScriptResourceMapping named jquery(case-sensitive).*

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.InvalidOperationException: WebForms UnobtrusiveValidationMode requires a ScriptResourceMapping for 'jquery'. Please add a ScriptResourceMapping named jquery(case-sensitive).

Source Error:

An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

Stack Trace:

```
[InvalidOperationException: WebForms UnobtrusiveValidationMode requires a ScriptResourceMapping for 'jquery'. Please add a ScriptResourceMapping for 'jquery'. Please add a ScriptResourceMapping named jquery(case-sensitive).]
System.Web.UI.ClientScriptManager.EnsureJqueryRegistered() +248
System.Web.UI.WebControls.BaseValidator.RegisterUnobtrusiveScript() +26
System.Web.UI.WebControls.BaseValidator.OnPreRender(EventArgs e) +155
System.Web.UI.Control.PreRenderRecursiveInternal() +245
System.Web.UI.Control.PreRenderRecursiveInternal() +343
System.Web.UI.Control.PreRenderRecursiveInternal() +343
```

The Windows taskbar at the bottom shows the Start button, a search bar, and several application icons. The system tray on the right indicates a temperature of 23°C, the time 20:46, the date 03-12-2024, and a notification for 1 new notification.

- To solve above error :
- Open Web.config file and write following code :

- **Web.config :**

```
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.5" />
    <httpRuntime targetFramework="4.5" />
  </system.web>
  <appSettings>
    <add
      key="ValidationSettings:UnobtrusiveValidationMode"
      value="None" />
    </appSettings>
</configuration>
```

RequiredFieldValidator :

- The RequiredFieldValidator control ensures that the required field is not empty.
- It is generally tied to a text box to force input into the text box.
- It makes us to enter a value into a form control before submitting the form.
- If we do not enter a value in the form, it gives error.
- As an Example of using RequiredFieldValidator is username for login must not blank.

- **Syntax :**

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator1"
    runat="server" ControlToValidate="TextBox1"
    ErrorMessage="UserName Must not blank">
</asp:RequiredFieldValidator>
```

Example : File.aspx

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

```
<div>
```

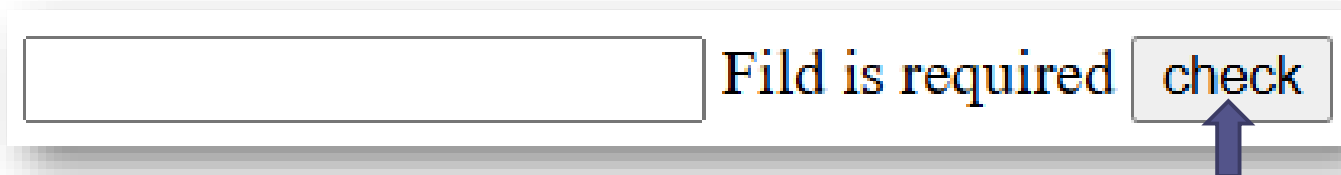
```
<asp:TextBox ID="u_nm" runat="server"></asp:TextBox>
```

```
<asp:RequiredFieldValidator ID="req" runat="server"  
ControlToValidate="u_nm" ErrorMessage="Fild is  
required"></asp:RequiredFieldValidator>
```

```
<asp:Button ID="btn_save" runat="server" Text="check"/>
```

```
</div>
```

```
</form>
```



Fild is required

Property	Description
AccessKey	It is used to set keyboard shortcut for the control.
BackColor	It is used to set background color of the control.
BorderColor	It is used to set border color of the control.
Font	It is used to set font for the control text.
ForeColor	It is used to set color of the control text.
Text	It is used to set text to be shown for the control.
ToolTip	It displays the text when mouse is over the control.
Visible	To set visibility of control on the form.
Height	It is used to set height of the control.
Width	It is used to set width of the control.
ErrorMessage	It is used to set error message that display when validation fails.
ControlToValidate	It takes ID of control to validate.
Type	It is used to define Data type of values for comparison

CompareValidator Control :

- CompareValidator control is used to check value of two controls.
- It is used to check whether both controls have got same value or not.
- If not then it gives an error message else nothing.
- Example of CompareValidator is Password and Confirm Password.
- In this example we compare whether both the passwords entered are same or not.
- CompareValidator not only compare two controls, but it also checks the data type of the control.
- For example if we want that control have only integer value or string value, it is possible using this control.
- Some example of CompareValidator are:
 - Confirm email address must be same
 - Confirm password must be same

Example : File.aspx

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

```
<div>
```

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

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

```
<asp:CompareValidator ID="cmp" runat="server"
```

```
ControlToValidate="con_pwd" ControlToCompare="pwd"
```

```
Operator="Equal" Type="String" ErrorMessage="password is  
not match" ForeColor="Red"></asp:CompareValidator>
```

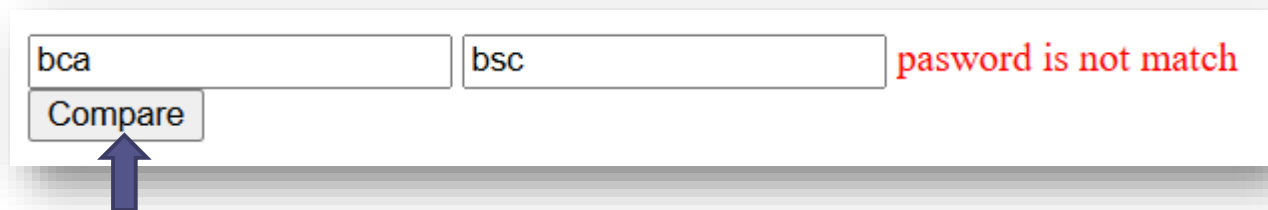
```
</div>
```

```
<div>
```

```
<asp:Button ID="btn" runat="server" Text="Compare" />
```

```
</div>
```

```
</form>
```



The screenshot shows a web form with two text input fields. The first field contains the text 'bca' and the second field contains 'bsc'. Below the first field is a button labeled 'Compare'. To the right of the second field, the text 'password is not match' is displayed in red, indicating a validation error. A blue arrow points upwards towards the 'Compare' button.

Properties	Description
Type	It specifies the data type.
ControlToCompare	It specifies the value of the input control to compare with.
ValueToCompare	It specifies the constant value to compare with.
Operator	It specifies the comparison operator, the available values are: Equal, NotEqual, GreaterThan, GreaterThanEqual, LessThan, LessThanEqual, and DataTypeCheck.

Range validator Control :

- RangeValidator control is used to check value of the control falls between specific range or not.
- The value can be of type numeric, date or string.
- Some example of RangeValidator Control are:
 1. Input age within a specific range.
 2. Input Birth date within a range.

Example : file.aspx

```
<asp:TextBox ID="txtage" runat="server"></asp:TextBox>
<asp:RangeValidator ID="RangeValidator1" runat="server"
  ControlToValidate="txtage" ErrorMessage="&gt; 18 and
  &lt; 30" ForeColor="#993366" MinimumValue="18"
  MaximumValue="30" Type="Integer">
</asp:RangeValidator>
```

Age

> 18 and < 30

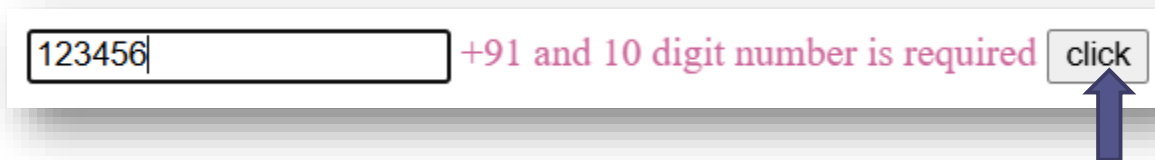
Property	Description
ControlToValidate	It takes ID of control to validate.
ErrorMessage	It is used to display error message when validation failed.
Type	It is used to set datatype of the control value.
MaximumValue	It is used to set upper boundary of the range.
MinimumValue	It is used to set lower boundary of the range.

RegularExpression Control :

- RegularExpressionValidator control is used to check value of given control against a specified expression.
- It is used to check the value of control against string value, numeric value, alphanumeric value etc.
- Some example of RegularExpressionValidator Control are:
 1. To input mobile number in specific format such as +91-9999999999.
 2. To input a pin code, code must be of 6 digits only such as 361002.
 3. To input an email id such as kamani_science.edu@gmail.com
 4. To input a web URL such as <http://google.co.in>
- **ValidationExpression** property is used to set the expression format which user wants.
- For testing or learn more about regular expression
- <https://regex101.com/>

Example : File.aspx

- `<asp:TextBox ID="txtmobile" runat="server"></asp:TextBox>`
- `<asp:RegularExpressionValidator
ID="RegularExpressionValidator2" runat="server"
ControlToValidate="txtmobile" ErrorMessage="+91 and 10
digit number is required" ForeColor="#CC6699"
ValidationExpression="^[+]{1}[0-9]{2}[0-9]{10}$">
</asp:RegularExpressionValidator>`



Property	Description
ErrorMessage	It is used to set error message that display when validation fails.
ControlToValidate	It takes ID of control to validate.
ValidationExpression	It is used to set regular expression to determine validity.

Regular Expression Tokens :

Meta characters	Description
.	Matches any character except \n.
[abcd]	Matches any character in the set.
[^abcd]	Excludes any character in the set.
[2-7a-mA-M]	Matches any character specified in the range.
\w	Matches any alphanumeric character and underscore.
\W	Matches any non-word character.
\s	Matches whitespace characters like, space, tab, new line etc.
\S	Matches any non-whitespace character.
\d	Matches any decimal character.
\D	Matches any non-decimal character.

← Apart from single character match, a class of characters could be specified that can be matched, called the metacharacters.

↙ Quantifiers could be added to specify number of times a character could appear.

Quantifier	Description
*	Zero or more matches.
+	One or more matches.
?	Zero or one matches.
{N}	N matches.
{N,}	N or more matches.
{N,M}	Between N and M matches.

CustomValidator Control :

- Custom validator control is used to validate an input with user defined function.
- Custom validator is used to define User Define Function which would validate the user input.
- We can distribute these function in two type
 1. Client Side Validation Function
 2. Server Side Validation Function
- The client side validation routine should be written in a scripting language, such as JavaScript or VBScript, which the browser can understand.
- The server side validation routine must be called from the control's ServerValidate event handler. The server side validation routine should be written in any .Net language, like C# or VB.Net.

File.aspx

```
<form id="form1" runat="server">
  <div>
    <asp:TextBox ID="num" runat="server"></asp:TextBox>
    <asp:CustomValidator ID="cust" runat="server"
OnServerValidate="cust_ServerValidate"
ControlToValidate="num" ErrorMessage="" Display="Dynamic"
ForeColor="Red"></asp:CustomValidator>
  <asp:Button ID="btn" runat="server" Text="click"
    OnClick="btn_Click" />
  </div>
</form>
```

Note :

double click on customValidator control for write a code on
file.aspx.cs

it will generate **OnServerValidate="cust_ServerValidate"**

File.aspx.cs

```
protected void cust_ServerValidate(object source,
    ServerValidateEventArgs args)
{
    if (Convert.ToInt32(args.Value) % 2 == 0)
    {
        cust.ErrorMessage = "number is even";
        args.IsValid = false;
    }
    else
    {
        cust.ErrorMessage = "number is odd";
        args.IsValid = false;
    }
}
```

<input type="text" value="3"/>	number is odd	<input type="button" value="click"/>
<input type="text" value="2"/>	number is even	<input type="button" value="click"/>

ValidationSummary Control :

- ValidationSummary control is used to show the summary of all the validation controls which are included in the current web page.
- This control is useful when we have a large web form which has more controls visible in a single screen.
- This control is used to get the summary of validation on the same page.
- Each validation control has two property named ErrorMessage and Text.
- ErrorMessage property is used by ValidationSummary control.
- ValidationSummary displays the message that are shown, when a ValidationSummary control is validated.

- **Example : File.aspx**

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


```
<asp:RequiredFieldValidator ID="req" runat="server"  
    ErrorMessage="User name is Required"  
    ControlToValidate="unm"></asp:RequiredFieldValidator>
```

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

```
<asp:RequiredFieldValidator ID="req" runat="server"  
    ErrorMessage="password is Required"  
    ControlToValidate="pwd"></asp:RequiredFieldValidator>
```

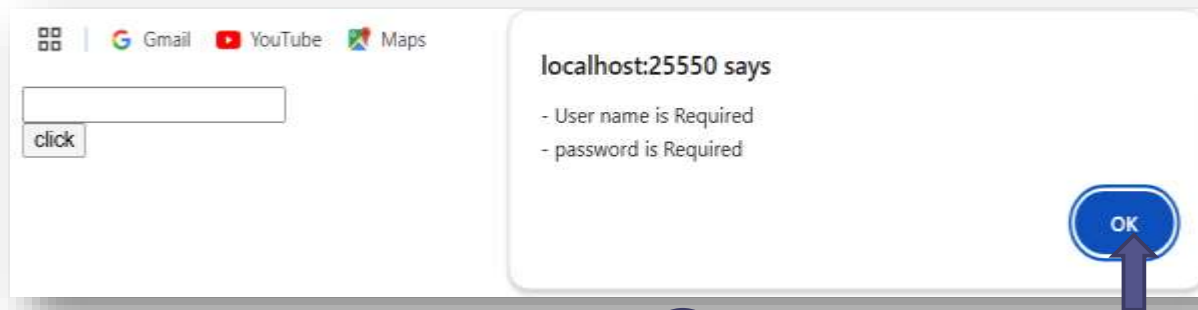
```
<asp:ValidationSummary ID="summary" runat="server"  
    ShowMessageBox="true"/>
```

1



click

2



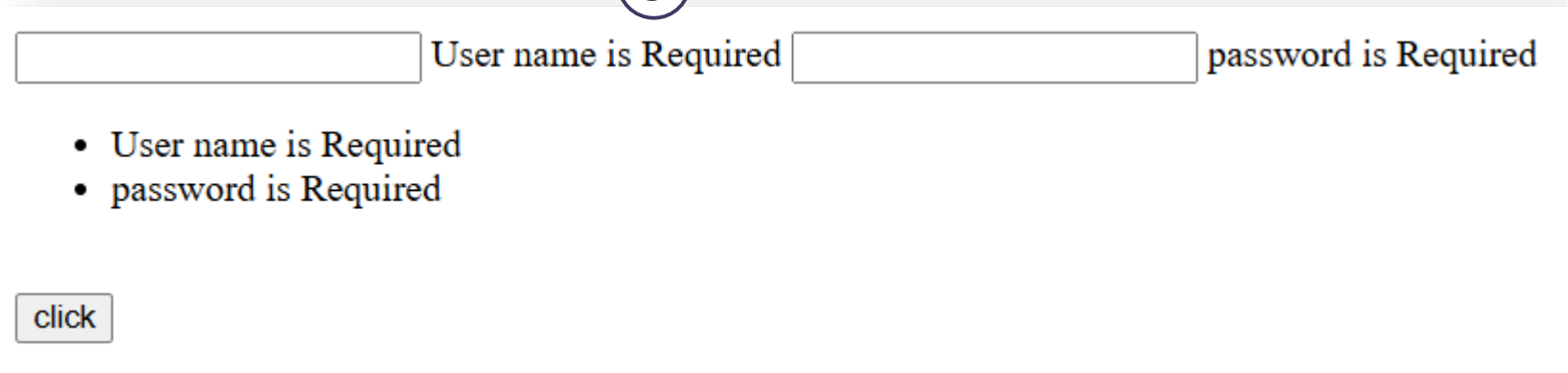
localhost:25550 says

- User name is Required
- password is Required

OK

-----ShowMessageBox="true"

3



User name is Required password is Required

- User name is Required
- password is Required

click

Property	Meaning
DisplayMode	It is used to display the summary of all the error messages in different ways such as List, BulletList or SingleParagraph. By default, the display mode is of BulletList.
HeaderText	It is used to specify the header text for ValidationSummary control.
ShowMessageBox	It is Boolean property which is used to see error list as MessageBox.
ShowSummary	It is Boolean property which is used to see error list as Summary.

Difference between label V/S literal

Label Control	Literal Control
Label control can be styled i.e. its Font, Color, Font Size, etc. can be easily changed.	Literal control cannot be styled as it does not use any HTML tag.
Label control enables to display static text on the web page.	Literal control is used most frequently when adding content dynamically to the page.
Label control does have a "cssclass" Property.	Literal does not have a "cssclass" Property.
By default Text property of Label contains "Label1".	text property of Literal control does not contains any text means It's blank.
Label control can be easily accessed via JavaScript or jQuery.	Literal control in spite of giving ID is rendered without ID hence cannot be accessed via JavaScript or jQuery.
Label control is less lightweight compare to Literal control.	Literal control is more lightweight compare to Label Control.

Assignment Questions :

1. What is ASP.net? Explain with advantages.
2. Explain .net framework architecture.
3. Explain different types of files in ASP.net.
4. Explain ListBox, DropDownList, with example.
5. Explain FileUpload control with example.
6. Explain all types of validation controls in detail.
7. Explain Hyperlink, ImageMap control in detail.