ASP.NET
TYBSC CS

Madhuri Sawant

Evolution of web development

➔A web application consists of a set of web pages that are generated in response to user requests

➔A page that contains controls like button , textbox , checkbox , radio button is called a web form ,and an ASP.NET application consists of one web form for each page in the application

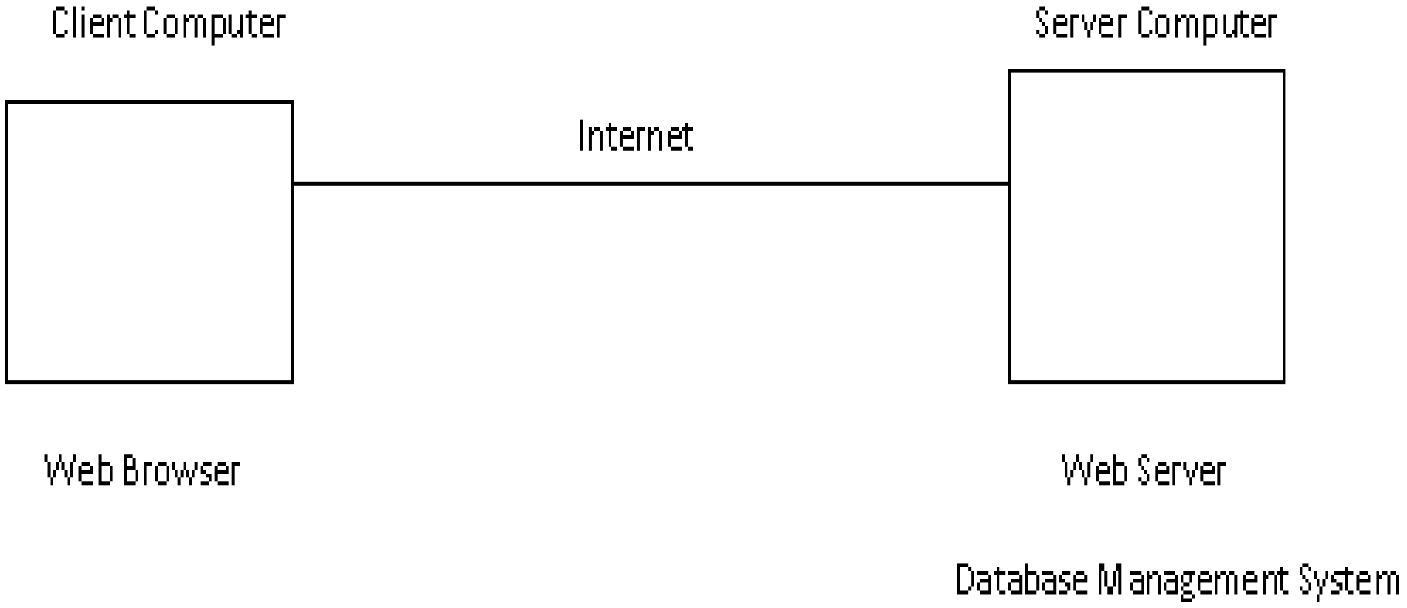
➔Functions of the web application are split between a client computer and a server computer

➔To access a web application, you use a web browser that runs on a client computer.

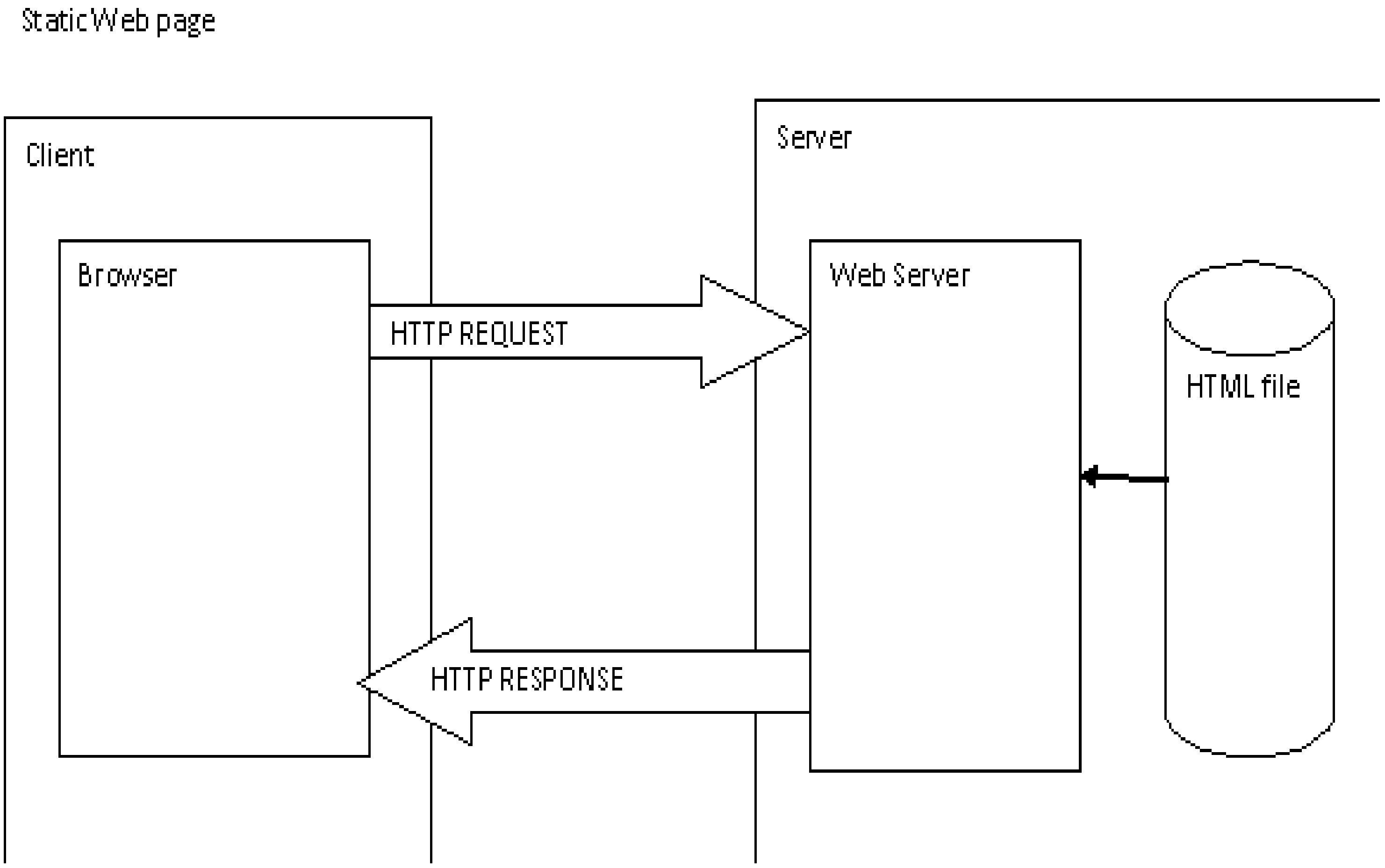
➔The web application itself is stored on the server computer. This computer runs web server software that enables it to send web pages to web browsers

➔The web browser provides the user interface for the application→ implemented as a series of web pages.

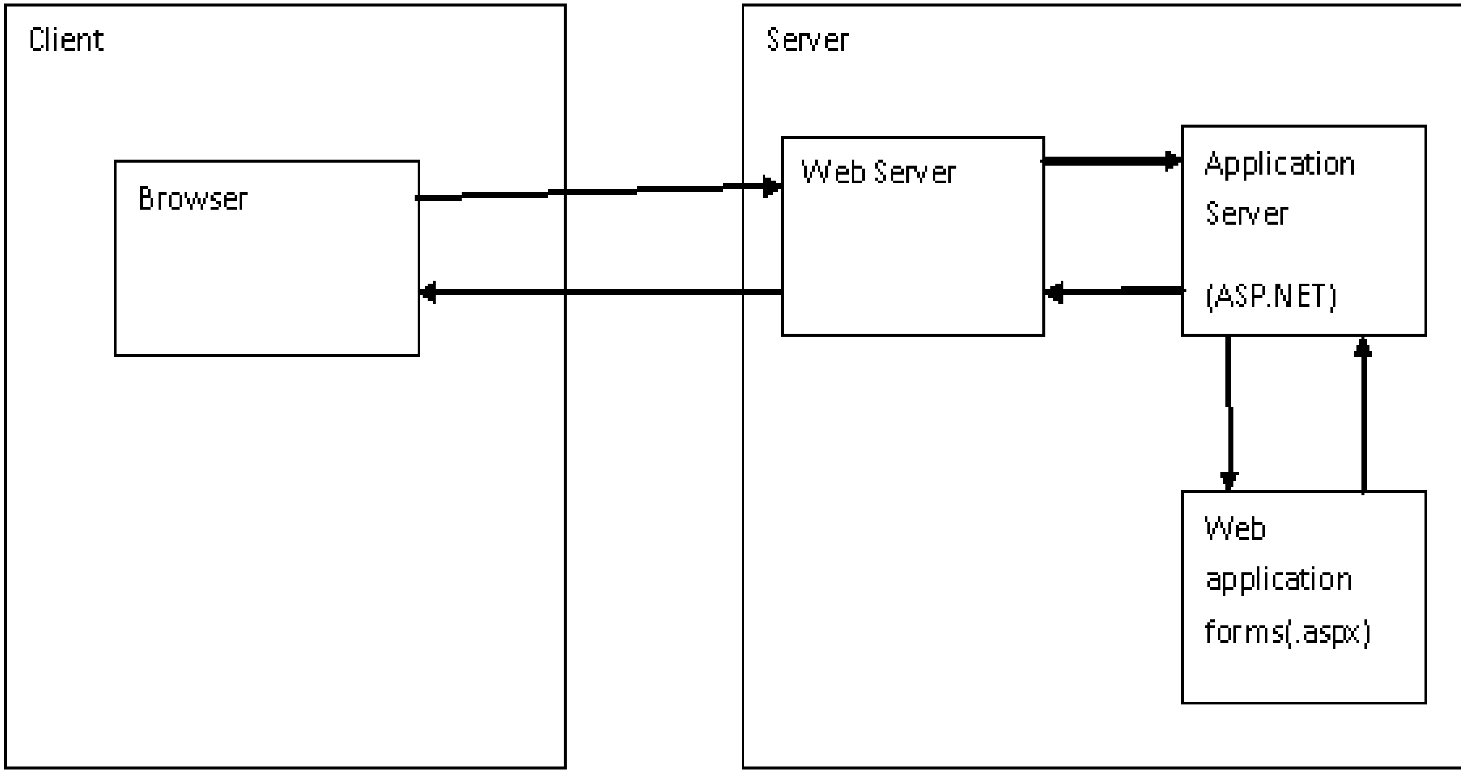
Component of a web application



Static web page



Dynamic web page



Client side and server side programming

- Client side programming → The complete application is downloaded to the client browser and then executed locally to display the result. The drawback of this technology is they are not equally supported by all browsers and operating system
- Server Side programming → The complete application is hosted on the server. The client requests the web page to the server. Server executes the server side application and returns the page result to the client. The result always is in simple HTML format.
- ASP.NET is designed as a server – side technology. All ASP.NET code executes on the server

Creating ASP.net application

1) start Microsoft Visual Studio 2008

2) Use file menu →New web site submenu

3) Select the type of application as ASP.NET web site template for ordinary application

4)Select the .NET version that you want to use from the list in the top-left corner of the dialog box

5)Specify the location of your application using the browse button

6)Select the language as Visual Basic/C# which you want to use for programming

7)Click ok button to create the website

To add web form to your web application

Website Menu → Add New Item submenu

It will open an Add new dialog box from which select web form

Designing Web pages

- When you start the web application , it has three files
- Default.aspx :This is a home page
- Default.aspx.vb : This holds the code for the home page
- Web.config : This is a configuration file
- To design the web page selects the web page from the solution explorer (starts the default.aspx)
- Three different views are available for a single web page
- Design View : It is graphical representation of the page
- Source View : It contains the html code and asp.net control tags
- Split View : This is a combined view using which you can view design as well as source view at once

ASP.NET file types

1)Ends with .aspx →These are ASP.NET Web pages .They contain the user interface and optionally, the underlying application code.

2)Ends with .ascx →These are ASP.NET user controls. User controls allow you to develop a small piece of user interface and reuse it in as many web forms as you want without repetitive code.

3) Ends with .asmx →These are ASP.NET web services –collections of methods that can be called over the internet. Web services work differently than web pages.

4) Web config →This is the XML based configuration file for your ASP.NET application. It includes settings for customizing security,state management , memory management and so on.

5) Global.asax →This is the global application file. You can use this file to define global variables and react to global events.

6)Ends with .vb →These are code-behind files that contain VB code. They allow you to separate the application logic from the user interface of a web page

Anatomy of web form

1) Page Directive →The first set of tags for each web form defines a page directive that provides four attributes

language –> indicates the language to use when compiling inline code blocks (<% …%>) and all the code that appears in the page <script> section .

AutoEventWireup →A Boolean attribute that indicates whether page events are automatically enabled. Set to true by default.

CodeFile →specifies the code behind file.

Inherits → Defines the base class for the page to inherit.

2) DOCTYPE → The second set of tags define a DOCTYPE declaration, which tells the browser what version of HTML ,the HTML document uses.

3) ELEMENTS →The html tags marks the beginning and end of the HTML document.

→ The content of the web page is defined with the DIV tags , which are with in the BODY and FORM tags.

→ The asp tags within the DIV tags define the server control that appear on the page.

→run at attribute with value “server” indicates that the form will be processed on server by ASP.NET

Code behind files

- With code behind you create a separate code file (a .vb file for VB.NET or .cs file for C#) to match every .aspx file
- The .aspx file contains the user interface logic which is the series of HTML code and ASP.NET tags that create controls on the page.
- The .vb or .cs file contains code only.
- At the top of the .aspx file ,you use a special Page directive that identifies the matching code behind file.
- Its easier to separate the code responsible for creating GUI (.aspx) and code responsible for actual functionality.
- Its convenient as well from compilation as well as debugging point of view.

Webform1.aspx-source view

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="WebForm1.aspx.vb"
    Inherits="WebForm1" %>

<!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>Untitled Page</title>
</head>
<body>
  <form id="form1" runat="server">
  <div style="height: 133px; width: 717px">

    <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
    <asp:Button ID="Button1" runat="server" Text="Button" />

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

WebForm1.aspx.vb

```
Partial Class WebForm1
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
     System.EventArgs) Handles Me.Load
        Label1.Text = "Welcome to the world of ASP.NET"
    End Sub

    Protected Sub Button1_Click(ByVal sender As Object, ByVal e
     As System.EventArgs) Handles Button1.Click
        Label1.Text = "U clicked button"
    End Sub
End Class
```

Global.asax file

- Using application events you can write logging code that runs every time a request is received, no matter what page is being requested.
- Global.asax file provides event handlers for application events.
- The Global.asax file allows you to write code that responds to global application events.
- The Global.asax file looks similar to a normal .aspx file, except that is cant contain any HTML or ASP.NET tags.Instead,it contains event handlers.
- To add a Global.asax file to an application in Visual Studio ,

  choose Website →→ Add New Item

  Select the Global Application Class file type

  Then click Ok.

Global.asax file

```
<%@ Application Language="VB" %>
<script runat="server">
   Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
   End Sub

   Sub Application_End(ByVal sender As Object, ByVal e As EventArgs)
   End Sub

   Sub Application_Error(ByVal sender As Object, ByVal e As EventArgs)
   End Sub

   Sub Session_Start(ByVal sender As Object, ByVal e As EventArgs)
   End Sub

   Sub Session_End(ByVal sender As Object, ByVal e As EventArgs)
   End Sub

   Sub Application_EndRequest(ByVal Sender As Object, ByVal e As EventArgs)
      Response.Write("<hr> This page was served at " & DateTime.Now.ToString())

   End Sub
</script>
```

Global.asax file

Event Handling Method	Description
Application_Start()	Occurs when the application starts, which is the first time it receives a request from any user.
Application_End()	Occurs when the application is shutting down
Application_BeginRequest()	Occurs with each request the application receives, just before the page code is executed.
Application_EndRequest()	Occurs with each request the application receives ,just after the page code is executed.
Session_Start()	Occurs whenever a new user request is received and a session is started .
Session_End()	Occurs when a session times out or is programmatically ended.

ASP.NET configuration files

- The ASP.NET configuration system features an extensible infrastructure that enables you to define configuration settings at the time your ASP.NET applications are first deployed so that you can add or revise configuration settings at any time with minimal impact on operational Web applications and servers.
- Every web application includes a web.config file that configures fundamental settings.
- Configuration information for ASP.NET resources is contained in a collection of configuration files, each named Web.config .
- Each configuration file contains a nested hierarchy of XML tags and subtags with attributes that specify the configuration settings.
- Because the tags must be well-formed XML, the tags, subtags, and attributes are case-sensitive.
- Configuration information is stored in XML-based text files.

ASP.NET configuration files

The ASP.NET configuration files have several key advantages

1) They are never locked :- Can update web.config settings at any point, even while your application is running .
2) They are easily accessed and replicated :- can change a web.config file from a remote computer .
3) The settings are easy to edit and understand :- the settings in the web.config file are human readable.

ASP.NET configuration files

```
<?xml version=“1.0” ?>
<configuration>
<configSections>….</configSections>
<appSettings>…</appSettings>
<connectionStrings>…</connectionStrings>
<system.web>…</system.web>
<system.codedom>…</system.codedom>
<system.webServer>…</system.webServer>
</configuration>
```

- The entire content of the file is nested in a root <configuration> element

ASP.NET configuration files

The web.config file is case-sensitive , like all XML documents , and starts every setting with a lowercase letter

1) The **<appSettings>** section allows you to add your own miscellanous pieces of information

2) The **<connectionStrings>** section allows you to define the connection information for accessing a database

3) The **<system.web>** section holds every ASP.NET setting you'll need to configure

Server Controls

- These controls are created and configured as objects

➔ASP.NET actually provides two sets of server – side controls that you can incorporate into your web forms.

HTML Server Controls :These are server based equivalents for standard HTML elements

Web Controls : Similar to HTML server controls but they provide a richer object model with a variety of properties for style and formatting details

closely resemble the controls used for windows development .

also feature some user interface elements that have no direct HTML equivalent ,such as the GridView ,Calendar ,and Validation controls .

HTML Server Control

- HTML server controls are HTML tags understood by the server.
- HTML elements in ASP.NET files are, by default, treated as text.
- To make these elements programmable, add a runat="server" attribute to the HTML element.
- This attribute indicates that the element should be treated as a server control.
- The id attribute is added to identify the server control. The id reference can be used to manipulate the server control at run time.
- All HTML server controls must be within a <form> tag with the runat="server" attribute.

HTML Server Control

HTML server control

In the following example we declare an HtmlAnchor server control in an .aspx file

```
<html>
<body>
<form runat="server">
<a id="link1" runat="server">Visit W3Schools!</a>
</form>
</body>
</html>
```

HTML SERVER CONTROLS

➔HTML server controls provide an object interface for standard HTML elements. They provide three key features:

1) They generate their own interface :set properties in code ,and the underlying HTML tag is created automatically when the page is rendered and send to the client.

2) They retain their state : web is stateless, ordinary web pages need to do a lot of work to store information between request .HTML server controls handle this task automatically.

3) They fire server side events : Eg:buttons fire an event when clicked . Code respond to these events, just like ordinary controls in a windows application. With event based programming, you can easily respond to individual user actions and create more structured code.

HTML Control class

Class Name	HTML Element	Description
HtmlForm	<form>	All ASP.NET server controls must be placed inside an HtmlForm control
HtmlAnchor	<a>	A hyperlink that the user clicks to jump to another page
HtmlImage	<img>	Link that points to an image,to be inserted into the web page at the current location
HtmlTable,HtmlTableRow, and HtmlTableCell	<table>,<tr>,<th> and <td>	A table that displays multiple rows and columns of static text

HTML Control class

HtmlInputButton, HtmlInputSubmit, And HtmlInputReset	<input type=“button”>, <input type=“submit”> and <input type=“reset”>	A button that the user clicks to perform an action,submit the page,or clear all the user supplied values in all the controls
HtmlButton	<button>	A button that the user clicks to perform an action
HtmlInputCheckBox	<input type=“checkbox”>	A checkbox that the user can check or clear
HtmlInputRadioButton	<input type=“radio”>	A radio button that can be selected in a group
HtmlInputText and HtmlInputPassword	<input type=“text”> and<input type=“password”>	A single line text

HTML Control class

HtmlInputImage	<input type=“image”>	Inserts a “clickable” image that submits the page
HtmlInputFile	<input type=“file”>	A browse button and text box that can be used to upload a file to your web server
HtmlInputHidden	<input type=“hidden”>	Contains text information that will be sent to the server but wont be visible in the browser
HtmlSelect	<select>	A drop down or regular list box where the user can select an item
HtmlHead and HtmlTitle	<head> and <title>	Represents the header information for the page
HtmlGeneric Control	Any other HTML element	This control can represent a variety of HTML elements that don’t have dedicated control class.

WEB CONTROLS

- Web server controls are special ASP.NET tags understood by the server.
- Like HTML server controls, Web server controls are also created on the server and they require a runat="server" attribute to work.
- However, Web server controls do not necessarily map to any existing HTML elements and they may represent more complex elements.
- Syntax: <asp:control_name id="some_id" runat="server" />

WEB CONTROLS

Web Server Controls

In the following example we declare a Button server control in an .aspx file.

```
<html>
  <body>
  <form runat="server">
  <asp:Button id="button1" Text="Click me!"
  runat="server" OnClick="submit"/>
  </form>
  </body>
  </html>
```

WEB CONTROLS

1. They provide a rich user interface :-A web control is programmed as an object but doesn’t necessarily correspond to a single element in the final HTML page. The control creates the required HTML tags for you.
2. They provide a consistent Object Model :-A simple text box can appear as one of three elements, including <textarea> , <input type =“text”> and <input type=“password”> .With web controls ,these three elements are consolidates as a single Textbox control
3. They tailor their output automatically :- ASP.NET server controls can detect the type of browser and automatically adjust the HTML code they write to take advantage of features . don't need to know about the client. This feature is known as adaptive rendering.
4. They provide high-level feature :-Web controls allow you to access additional events, properties and methods that don’t correspond directly to typical HTML controls.

WEB CONTROLS

- ASP.NET web controls can detect the target browser’s capabilities and render themselves accordingly
- Newer set of controls are available under web controls which can be used in the same manner as any HTML control eg Calendar controls and Grid View control
- All the processing would be done at the server side
- ASP.NET web controls have an object model different from the traditional HTML .Provide a set of properties and methods that can change the outlook and behavior of the controls.
- ASP.NET web controls have higher level of abstraction. An output of an ASP.NET server control can be the result of many HTML tags that combine together to produce the control and its events.Eg Grid View control
- Syntax for creating web server control is

<asp:control_name id=“some_id” runat=“server”   />

Or   <asp:TextBox id=“txt1” runat=“server”   />