ASP.NET

Sanjana Singh
Dronacharya College of Engineering
Khentawas, Haryana

What is ASP.net

ASP.NET is an open-source[2] server-side Web application framework designed for Web development to produce dynamic Web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.

It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages.

ASP.NET is in the process of being re-implemented as a modern and modular web framework, together with other frameworks likeEntity Framework. The new framework will make use of the new open-source .NET Compiler Platform (code-name "Roslyn") and becross platform. ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages (a platform using only Razor pages) will merge into a unified MVC 6.[1] The project is called "ASP.NET vNext".

Asp.Net is a server-side Web application framework designed for Web development to produce dynamic Web pages. It allows programmers to build dynamic web sites, web applications and web services. Asp.Net is built on the Common Language Runtime (CLR), allowing programmers to write Asp.Net code using any supported .NET language.

The .Net Framework family also includes two versions for mobile or embedded device use. A reduced version of the framework, the .NET Compact Framework, is available on Windows CE platforms, including Windows Mobile devices such as smartphones. Additionally, the .NET Micro Framework is targeted at severely resource-constrained devices.

Characteristics of Asp.Net Pages

- Directives
- User controls
- Custom Controls
- Rendering Techniques
- State management
- Application
- Session state
- Performance

Directives

A directive is special instructions on how ASP.NET should process the page. The most common directive is <%@ Page %> which can specify many things, such as which programming language is used for the server-side code.

User controls

User controls are encapsulations of sections of pages which are registered and used as controls in ASP.NET. User controls are created as ASCX markup files. These files usually contain static (X)HTML markup, as well as markup defining server-side Web controls. These are the locations where the developer can place the required static and dynamic content. A user control is compiled when its containing page is requested and is stored in memory for subsequent requests.

Custom controls

Programmers can also build *custom controls* for ASP.NET applications. Unlike user controls, these controls do not have an ASCX markup file, having all their code compiled into a dynamic link library (DLL) file. Such custom controls can be used across multiple Web applications and Visual Studio projects.

Rendering technique

ASP.NET uses a *visited composites* rendering technique. During compilation, the template (.aspx) file is compiled into initialization code which builds a control tree (the composite) representing the original template. Literal text goes into instances of the Literal control class, and server controls are represented by instances of a specific control class. The initialization code is combined with userwritten code (usually by the assembly of multiple partial classes) and results in a class specific for the page. The page doubles as the root of the control tree.

State management
ASP.NET applications are hosted by a Web server and are accessed using the stateless HTTP protocol. As such, if an application uses stateful interaction, it has to implement state management on its own. ASP.NET provides various functions for state management. Conceptually, Microsoft treats “state” as GUI state.

APPLICATION
Application state is held by a collection of shared user-defined variables. These are set and initialized when the Application_OnStartevent fires on the loading of the first instance of the application and are available until the last instance exits. Application state variables are accessed using the Applications collection, which provides a wrapper for the application state. Application state variables are identified by name.

SESSION STATE
Server-side session state is held by a collection of user-defined session variables that are persistent during a user session. These variables, accessed using the Session collection, are unique to each session instance. The variables can be set to be automatically destroyed after a defined time of inactivity even if the session does not end. Client-side user session is maintained by either a cookie or by encoding the session ID in the URL itself.

<u>WHY USE ASP.NET?</u>

1. Improved Performance and Scalability

- Compiled Execution: ASP.NET is much faster than classic ASP, while preserving the "just hit save" update model of ASP. No explicit compile step is required. ASP.NET automatically detects any change, dynamically compiles files if needed, and stores the compiled results to reuse for subsequent requests. Dynamic compilation ensures that your application is always up to date, and compiled execution makes it fast. Most applications migrated from classic ASP to ASP.NET see a 3x to 5x increase in pages served.

- Rich Output Caching: ASP.NET output caching can dramatically improve the performance and scalability of your application. When output caching is enabled on a page, ASP.NET executes the page once and saves the result in memory before sending it to the user. When another user requests the same page, ASP.NET serves the cached result from memory without re-executing the page. Output caching is configurable, and it can be used to cache individual regions or an entire page.

- Web Farm Session State: ASP.NET session state lets you share session data across all machines in a Web farm. Now a user can hit different servers in the Web farm over multiple requests and still have full access to session data.

2. Enhanced Reliability

Memory Leak, Dead Lock, and Crash Protection: ASP.NET automatically detects and recovers from errors such as dead locks and memory leaks to ensure that your application is always available. For example, when a memory leak is detected, ASP.NET automatically starts up a new copy of the ASP.NET worker process and directs all new requests to the new process. After the old process has finished processing pending requests, it is gracefully disposed of and the leaked memory is released.

3. Easy Deployment

- "No Touch" Application Deployment: With ASP.NET you can deploy an entire application by copying it to the server. Configuration settings are stored in an XML file within the application.
- Dynamic Update of Running Application: ASP.NET lets you update compiled components without restarting the Web server. Unlike classic COM components that required the Web server to be manually restarted when an update was deployed, ASP.NET automatically detects the change and starts using the new code.
- Easy Migration Path: ASP.NET runs side by side on IIS with classic ASP applications on Microsoft Windows 2000 and Windows XP, and on

members of the Windows Server 2003 family. You can migrate one application at a time, or even single pages. ASP.NET even lets you continue to use your existing classic COM business components.

4. Developer Productivity
   - Easy Programming Model: ASP.NET makes building real-world Web applications dramatically easier with server controls that let you build great pages with far less code than classic ASP.
   - Flexible Language Options. ASP.NET supports not only Microsoft Visual Basic Scripting Edition (VBScript) and Microsoft JScript but also more than 25 .NET languages, including built-in support for Visual Basic .NET, Microsoft C#, and JScript .NET.
   - Rich Class Framework: The .NET Framework class library offers over 4,500 classes that encapsulate rich functionality such as XML, data access, file upload, regular expressions, image generation, performance monitoring and logging, transactions, message queuing, and SMTP mail.

REFERENCE

- https://en.wikipedia.org/wiki/ASP.NET
- http://blog.spec-india.com/characteristics-of-asp-net-pages
- https://kirandn.wordpress.com/about/characteristics-of-asp-net/
- http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/ad2a22b9-135c-432a-bc9f-c67f074242b7.mspx?mfr=true