Unit -5 Web service, Configuration & Deployment

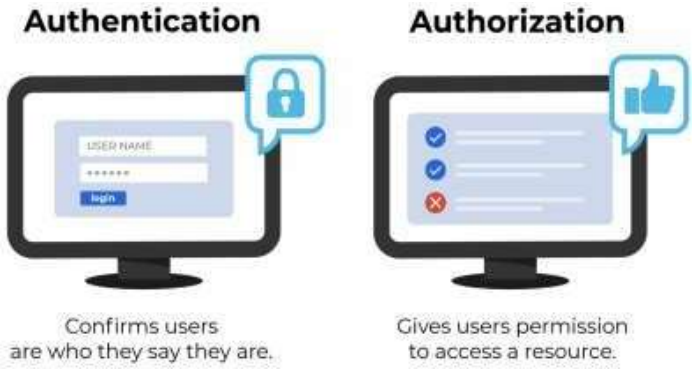
5.1 Authentication & Authorization:

Authentication:

Authentication is the process of identifying someone's identity by assuring that the person is the same as what he is claiming for. It is used by both server and client. The server uses authentication when someone wants to access the information, and the server needs to know who is accessing the information. The client uses it when he wants to know that it is the same server that it claims to be. The authentication by the server is done mostly by using the *username and password.* Other ways of authentication by the server can also be done using *cards, retina scans, voice recognition, and fingerprints.* Authentication does not ensure what tasks under a process one person can do, what files he can view, read, or update. It mostly identifies who the person or system is actually.

Authorization:

Authorization is the process of granting someone to do something. It means it a way to check if the user has permission to use a resource or not. It defines that what data and information one user can access. The authorization usually works with authentication so that the system could know who is accessing the information.



Authentication	Authorization
Authentication is the process of identifying a user to provide access to a system.	Authorization is the process of giving permission to access the resources.
In this, the user or client and server are verified.	In this, it is verified that if the user is allowed through the defined policies and rules.
It is usually performed before the authorization.	It is usually done once the user is successfully authenticated.
It requires the login details of the user, such as user name & password, etc.	It requires the user's privilege or security level.
Example: Entering Login details is necessary for the employees to authenticate themselves to access the organizational emails or software.	**Example:** After employees successfully authenticate themselves, they can access and work on certain functions only as per their roles and profiles.
Authentication credentials can be partially changed by the user as per the requirement.	Authorization permissions cannot be changed by the user. The permissions are given to a user by the owner/manager of the system, and he can only change it.

5.2 Web.Config File :

A configuration file (web.config) is used to manage various settings that define a website. The settings are stored in XML files that are separate from your application code. In this way you can configure settings independently from your code. Generally a website contains a single Web.config file stored inside the application root directory. However there can be many configuration files that manage settings at various levels within an application. ASP.NET Configuration system is used to describe the properties and behaviors of various aspects of ASP.NET applications. Configuration files help you to manage the many settings related to your website. Each file is an XML file (with the extension .config) that contains a set of configuration elements. Configuration information is stored in XML-based text files.

There are number of important settings that can be stored in the configuration file. Some of the most frequently used configurations, stored conveniently inside Web.config file are:

- Database connections
- Caching settings
- Session States
- Error Handling
- Security

```
<configuration>
   <connectionStrings>
      <add name="myCon" connectionString="server=MyServer;database=puran;uid=puranm
ehra;pwd=mydata1223" />
  </connectionStrings>
</configuration/>
```

Different types of Configuration files

- Machine.config - Server or machine-wide configuration file
- Web.config - Application configuration files which deal with a single application

Machine.config File :

Configuration files are applied to an executing site based on a hierarchy. There is a global configuration file for all sites in a given machine which is called Machine.config. This file is typically found in the C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CONFIG directory.The Machine.config file contains settings for all sites running on the machine provided another .config further up the chain does not override any of these settings. Although Machine.config provides a global configuration option, you can use .config files inside individual website directories to provide more granular control. Between these two poles you can set a number of other .config files with varying degree of applicable scope.

Application Configuration file (Web.config) :

Each and Every ASP.NET application has its own copy of configuration settings stored in a file called Web.config. If the web application spans multiple folders, each sub folder has its own Web.config file that inherits or overrides the parent's file settings.

5.3 Global.asax file :

The Global. sax file resides in the root directory of an asp.net web application.Events and states such as session state and Application state are specified in the Global.asax file. Global.asax file is used to maintain session and application specific event within an application.We can have only one Global.asax file in a asp.net application.

Point to remember about global.asax:

1. They do not contain any HTML or ASP.NET tags.
2. Contain methods with specific predefined names.
3. They defined methods for a single class, application class.
4. They are optional, but a web application has no more than one global.asax file.

Example :

```
<%@ Application Language="C#" %>
<script runat="server">
   void Application_Start(object sender, EventArgs e)
   {
      // Code that runs on application startup
   }
   void Application_End(object sender, EventArgs e)
   {
      //  Code that runs on application shutdown
   }
   void Application_Error(object sender, EventArgs e)
   {
      // Code that runs when an unhandled error occurs
```

```
}
void Session_Start(object sender, EventArgs e)
{
   // Code that runs when a new session is started
}
void Session_End(object sender, EventArgs e)
{
   // Code that runs when a session ends.
   // Note: The Session_End event is raised only when the sessionstate mode
   // is set to InProc in the Web.config file. If session mode is set to StateServer
   // or SQLServer, the event is not raised.
}
```

5.4 Tracing:

ASP.NET tracing enables you to view diagnostic information about a single request for an ASP.NET page. ASP.NET tracing enables you to follow a page's execution path, display diagnostic information at run time, and debug your application. ASP.NET tracing can be integrated with system-level tracing to provide multiple levels of tracing output in distributed and multi-tier applications.

There are two ways:

1. Page Level Tracing
2. Application Level Tracing

Page Level Tracing

We can control whether tracing is enabled or disabled for individual pages. If tracing is enabled, when the page is requested, ASP.NET appends to the page a series of tables containing execution details about the page request. Tracing is disabled by default in an ASP.NET application.

Syntax:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWir
eup="true"  CodeFile="Default.aspx.cs"Inherits="chat_Default"  Trace="true"%>
```

Application Level Tracing

Instead of enabling tracing for individual pages, you can enable it for your entire application. In that case, every page in your application displays trace information. Application tracing is useful when you are developing an application because you can easily enable it and disable it without editing individual pages. When your application is complete, you can turn off tracing for all pages at once.

Syntax :

```
<system.web>
   <trace enabled="true" pageOutput="true" requestLimit="40" localOnly="false"/>
 </system.web>
```

</configuration>

5.5 Custom Error handling in asp.net:

When a run-time or design-time error occurs in an application, ASP.NET shows a default Yellowish-Red error page that gives a brief description of the error along with some status code. If this page is viewed by the developer, then he can analyse the actual reason by its status code, but if a user or anyone other than the developer is viewing this error page, they won't get the actual idea of why this error is coming. So to provide a friendly error message page we need to set custom error in web.config.
Some of the basic common errors with status codes are:

Error Code	Description
400	Bad Request The request cannot be fulfilled due to bad syntax
403	Forbidden The request was a legal request, but the server is refusing to respond to it
404	Not Found Page Not Found
405	Method not allowed A request was made of a page using a request method not supported by that page
408	Timeout server time out

Example:

```
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.5"/>
    <httpRuntime targetFramework="4.5"/>
  <customErrors mode="On">
    <error statusCode="404" redirect="error.aspx"/>

  </customErrors>
  </system.web>
</configuration>
```

5.6 Web Services:

Web Service is known as the software program. These services use the XML to exchange the information with the other software with the help of the common internet Protocols. In the simple term, we use the Web Service to interact with the objects over the internet.

Here are some points about the Web Service.

1. Web Service is not dependent on any particular language.
2. Web Service is a protocol Independent.
3. Web Service is platform-independent.
4. These services are dependent only on the given input.
5. Web Service is also Scalable.
6. Web Service is based on the XML.

Technology Used in Web Service

XML(Extensible Markup Language): Web Service specifies only the data. So, the application which understands the XML regarding the programming language or the platform can format the XML in different ways.

SOAP(Simple Object Access Protocol): The communication between the Services and the application is set up by the SOAP.

WSDL(Web Services Description Language): WSDL gives us a uniform method that is helpful to specify the Web Services to the other programs.

UDDI(Universal Description, Discovery, and Integration): With the help of UDDI, we can search the Web Service registries.

At the time of the deployment of these technologies, this allows the developers to do the packaging of the applications in the form of the Service and publishing of the Service on the network.

Advantages of Web Services

1. Web Services always use the open, text-based standard. Web service uses all those components even they are written in various languages for different platforms.
2. Web Service promotes the modular approach of the programming so that the multiple organizations can communicate with the same web service.
3. Web Services are easy to implement but expensive because they use the existing infrastructure, and we can repackage most of the applications as the Web Service.
4. Web Service reduces the cost of enterprise application integration and B2B communications.
5. Web Services are the interoperable Organization that contains the 100 vendors and promote them interoperability.

Limitations of Web Services

1. One of the limitations of the Web Services is that the SOAP, WSDL, UDDI requires the development.
2. Supports to the interoperability is also the limitation of the Web Service.
3. The limitation of web service is also royalty fees.
4. If we want to use web services for the high-performance situation, in that case, web service will be slow.
5. The use of web service increases the traffic on the network.

Example:

- Web Portal
- Whether Reporting
- Stock Quote
- News Headlines