What is 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.

 Usage of configuration file

  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.

 Benefits of XML-based Configuration files

* ASP.NET Configuration system is extensible and application specific information can be stored and retrieved easily. It is human readable.
* You need not restart the web server when the settings are changed in configuration file. ASP.NET automatically detects the changes and applies them to the running ASP.NET application.
* You can use any standard text editor or XML parser to create and edit ASP.NET configuration files.

What Web.config file contains?

  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 file looks like this**

1. <configuration>
2. <connectionStrings>
3. <add name="myCon" connectionString="server=MyServer;database=puran;uid=puranmehra;pwd=mydata1223" />
4. </connectionStrings>
5. </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.

Processing of a Web.config file

  When you initially run your web application, the runtime builds a cache of the configuration settings for your web application by flattening the layer of configuration files as below,

1. The Machine.config file settings are retrieved.
2. The settings from the root Web.config files are added to the caches, overwriting any conflicting settings that were earlier while reading the Machine.config file.
3. If there is a Web.config file at the root of the website, this file is read into the cache, all overwriting any existing settings. The resulting cache contains the setting for this website.
4. If you have subdirectories in your web application, each subdirectory can have a Web.config file that includes settings that are specific to the files and folders that are contained within the subdirectory. To calculate the effective setting for the folders, the website settings are read (step 1-4) and then this Web.config file is read into cache for this folder, overwriting (and thereby overriding) any existing settings.