Views of Proff. B.NAGARAJU

NIT

Microsoft VS .NET 2005

ASP .NET NOTEBOOK®

Author: SREEKANTH .P

Visit us @:www.dotnetsparkels.wordpress.com

E-mail:sreekanth@dotnetsparkles.wordpress.com

Profile

Professor's Profile:

Proff.B.NAGARAJU is a Sr DotNet Consultant for MNC's and Trainer in Naresh Information Technologies, Hyderabad, AP India. Started with Dbase III+, FoxPro, COBOL, C, ORACLE and today well-versed with top Microsoft Technologies ASP.NET, C#.NET also VB.NET, SQL SERVER. Latest Developments AJAX, Silver Light, LIVE etc. Interested to implement always new and wanted technologies. Professor has own Development Team for some projects. Love to teach more technical skills to the professionals.

Index

INTRODUCTION

ASP.NET

ASP.NET, the next version of ASP, is a programming framework used to create enterprise-class Web Applications. These applications are accessible on a global basis leading to effecient information managment. The advantages ASP.NET offers is more than just the next version of ASP.

Why ASP.NET?

Since 1995, Microsoft has been constantly working to shift it's focus from Windows-based platforms to the Internet. As a result, Microsoft introduced ASP (Active Server Pages) in November 1996. ASP offered the efficiency of ISAPI applications along with a new level of simplicity that made it easy to understand and use. However, ASP script was an interpreted script and consisted unstructured code and was difficult to debug and maintain. As the web consists of many different technologies, software integration for Web development was complicated and required to understand many different technologies. Also, as applications grew bigger in size and became more complex, the number of lines of source code in ASP applications increased dramatically and was hard to maintain. Therefore, an architecture was needed that would allow development of Web applications in a structured and consistent way.

The .NET Framework was introduced with a vision to create globally distributed software with Internet functionality and interoperability. The .NET Framework consists of many class libraries, includes multiple language support and a common execution platform. It's a very flexible foundation on which many different types of top class applications can be developed that do different things. Developing Internet applications with the .NET Framework is very easy. ASP.NET is built into this framework, we can create ASP.NET applications using any of the built-in languages.

Unlike ASP, ASP.NET uses the Common Language Runtime (CLR) provided by the .NET Framework. This CLR manages execution of the code we write. ASP.NET code is a compiled CLR code instead of interpreted code (ASP). CLR also allows objects written in different languages to interact with each other. The CLR makes developement of Web applications simple.

Advantages Using ASP.NET

- ASP.NET drastically reduces the amount of code required to build large applications
- ASP.NET makes development simpler and easier to maintain with an event-driven, server-side programming model
- ASP.NET pages are easy to write and maintain because the source code and HTML are together
- The source code is executed on the server. The pages have lots of power and flexibility by this approach
- The source code is compiled the first time the page is requested. Execution is fast as the Web Server compiles the page the first time it is requested. The server saves the compiled version of the page for use next time the page is requested
- The HTML produced by the ASP.NET page is sent back to the browser. The application source code you write is not sent and is not easily stolen
- ASP.NET makes for easy deployment. There is no need to register components because the configuration information is built-in

- The Web server continuously monitors the pages, components and applications running on it. If it noticies memory leaks, infinite loops, other illegal software or activities, it seamlessly kills those activities and restarts itself
- ASP.NET validates information (validation controls) entered by the user without writing a single line of code
- ASP.NET easily works with ADO .NET using data-binding and page formatting features
- ASP.NET applications run fater and counters large volumes of users without performance problems

Differences between ASP.NET and Client-Side Technologies

Client-side refers to the browser and the machine running the browser. Server-side on the other hand refers to a Web server.

Client-Side Scripting

Javascript and VBScript and generally used for Client-side scripting. Client-side scripting executes in the browser after the page is loaded. Using client-side scripting you can add some cool features to your page. Both, HTML and the script are together in the same file and the script is download as part of the page which anyone can view. A client-side script runs only on a browser that supports scripting and specifically the scripting language that is used. Since the script is in the same file as the HTML and as it executes on the machine you use, the page may take longer time to download.

Server-Side Scripting

ASP.NET is purely server-side technology. ASP.NET code executes on the server before it is sent to the browser. The code that is sent back to the browser is pure HTML and not ASP.NET code. Like client-side scripting, ASP.NET code is similar in a way that it allows you to write your code alongside HTML. Unlike client-side scripting, ASP.NET code is executed on the server and not in the browser. The script that you write alongside your HTML is not sent back to the browser and that prevents others from stealing the code you developed.

VALIDATORS

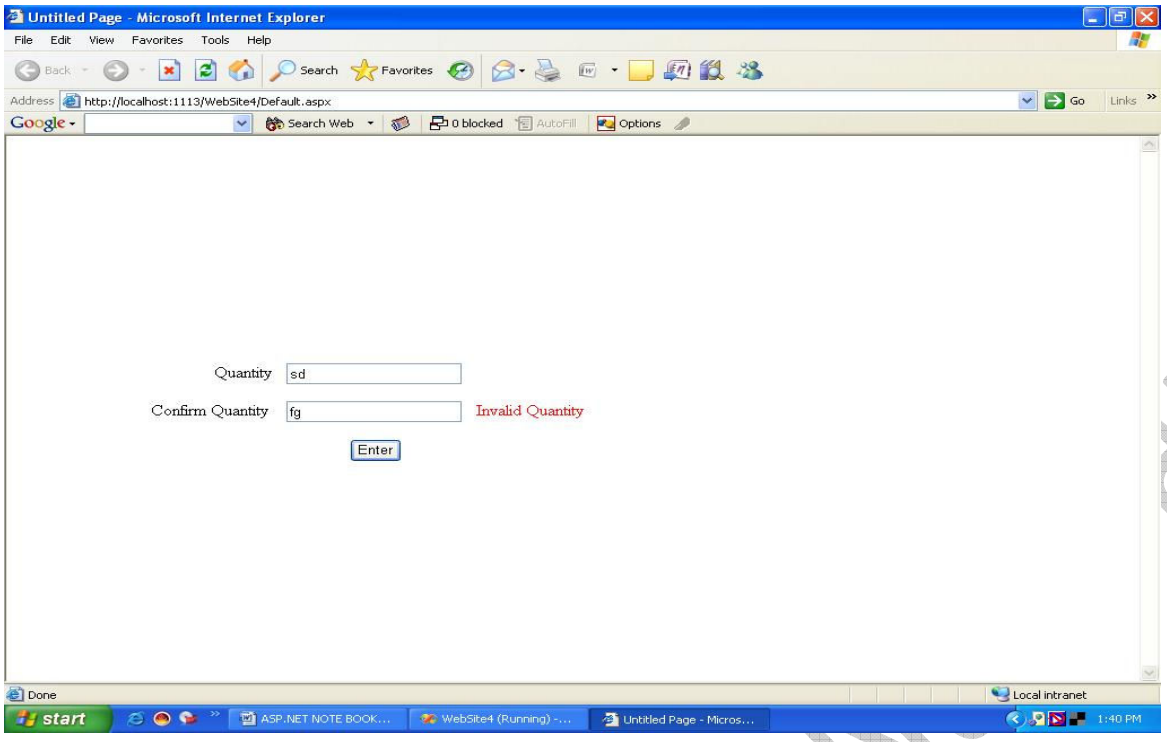
(a) Compare field Validator: Is a Control with compare feature.
Comparing one control with another control.

Properties:

(i) Control to Validate.
(ii) Control to Compare.
(iii) Operator.
(iv) Type.
(v) Error message.

EX:

To compare two Quantities or to compare any data with another data we use these validators.lets take an example of comparing two quantities.



Quantity Text Box1

Confirm Quantity Comp Text Box2

Enter

<asp: Compare Validator id=Compare Validator Control to validate "text Box2" Control to Compare "Text Box1"Operator ="Equal" Error Message="Quantity mismatch" Type=String Runat="server"/>

Interview Point: Server side control :-< asp: Text Box id=Text box1 runat=server/>

"asp:"in the above line is Tag prefix.

"Text Box "is the Tag name.

Syntax: <tag prefix: tag name<attributes>runat=server/>

(b) **Control-Value**

(i) Control to validate.
(ii) Value t Compare.
(iii) Type.
(iv) Operator.
(v) Error message.

(c) **Control-Data type**

(i) Control to validate.
(ii) Operator data type check.
(iii) Type.
(iv) Error Message.

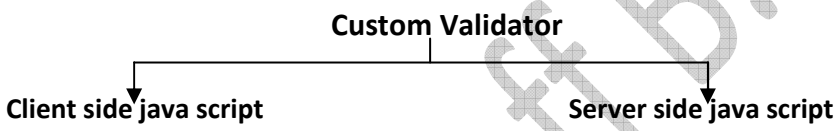
Regular Expression Validator
(Post.Pre name)
Mobile expression: - \d-->digits
\d{10}-->digits of 10
\d{1,10}-->between 1 to 10
\d{5,10}-->ranging 5 numbers only
\d{3} -\d{8} (ex: 040-23754526)
\ (A-Z){8}->8 alphabetic characters are allowed
\ (a-z A-Z){8}-Caps & Small

Note: Custom validator allows users to write the logic for validating data and then takes care of displaying an error and form submission. Using custom validator we can perform validation either in client or at server (if needed both places).client side means using javascript.And server side means using C#.NET.

Note:

ASP.NET controls perform by default validations in the client on submission it also perform validation at server. This is to ensure more accuracy and also to avoid malfunctioning that may occur during submission of data.

→Custom validator allows user to implement client side validation using java script.



Client side JavaScript:

(1) Place the custom validator in the form and set the following properties.

(i) Control to Validate
(ii) Error Message
(iii) Client validate function

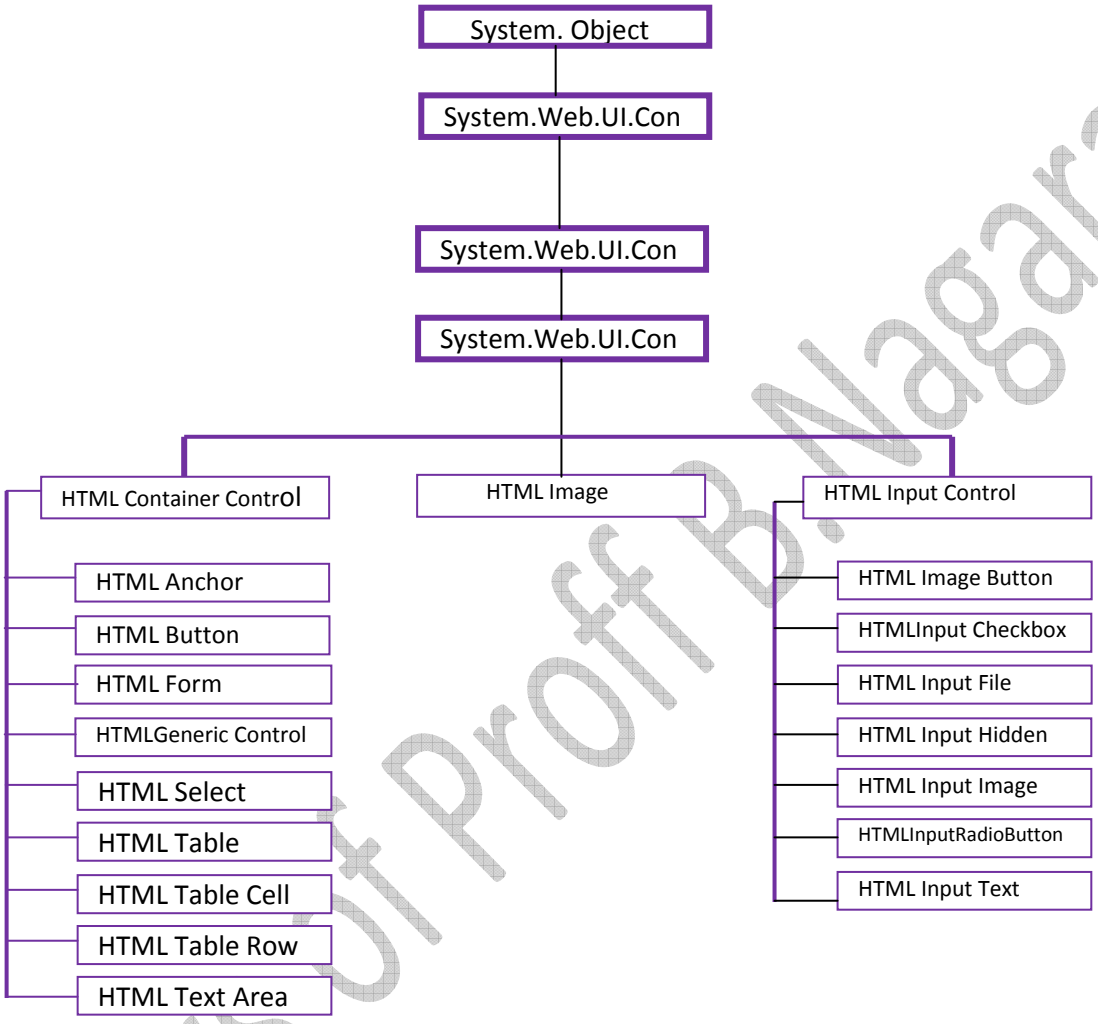
(2) Go to 'source' view or 'html' view and add script tag to it .inside script tag write "JavaScript" function with "two parameters" these two parameters specifies the object and the arguments are passed from control to function or function to control using which we can perform our required validations.

<script language="JavaScript">
Function f1(x, y).
x-is object.
Y-argument.
{
};

Ex: function check data(x, y)
{
If (y.value>10)
Y.Is Valid=True;
else
Y.Is valid=False;

Server side Java Script

HTML Controls Class Hierarchy:



GRID VIEW CONTROLS Date: 25/7/07

***Note**: -In .NET 1.0 we had data grid view control that was replaced in 2.0 with grid view.
-Data grid 1.1 control available in 2.0.
-Grid view is faster in performance.
-it also had a simplified use age.
-Dynamic nature.
-Have new features

***Note**: We cannot drag and drop a data grid but we have to go to the code and create.

➔ Data grid in ASP.NET 2.0 can be created just by writing data grid control creation code in source view (because there is no drag and drop provided by default)
Ex: <asp: Data Grid id="dg1"runat="server"/>

Grid view features

-Most featured rich control.
-Many built-in layouts for designing &dynamic behavior.
-Supports sorting of data.
-Supports Paging of data.
-Supports Editing/Updating/Deleting of data.
-Collection based approach is provided.
-Grid view with auto generates columns true on properties that generate columns dynamically. When data source is specified at run time and it is by default is TRUE.
-we can assign different data objects tables at runtime to provide different results.
- If auto generate columns is set to false then we need to provide columns on our own.
-With auto generate columns true we can really provide some4 automated sorting /paging tasks also.
-Grid view when assigned to data source control it automatically generates the code that is required to show data. It internally sets automate column is set to false.
-Enable sorting, enable paging.
-Grid view with data source controls –providing additionally sorting paging.

Bound Controls

(i) Data Object
(ii) Design
(iii) Bind

Steps to create data Grid

(i) Create a data grid view
(ii) Take a button
(iii) Double click on the design screen to go to the code view.
(iv) Grid view has capability to render itself.

GRID VIEW WITH AUTO GENERATES COLUMNS: (FALSE) (Date: 26/7/07)

- When auto generate columns is set to false we have to explicitly provide the columns information.

- Grid view has a collection called columns<columns>….</columns> that is used to define one or more columns.
- We can define columns in many ways but to categorize we will say in three ways.

METHOD-1:

Using<asp: Bound Field>

METHOD-2:

Using<asp: Template Field>
<Header Template>…. </Header>
<Item Template>…….</Item Template>……..other templates.

METHOD-3:

Using predefined fields of grid views like
<asp: Check Box Field…..> (Not normal check box)
<asp: image Field….> (not normal image).

Note: Grid view present data tabular presentation purely in the form of rows and columns.

USING BOUND FIELD

Process:→ Create button→Keep auto generate columns "False"→Drag and drop Grid view→Go to source view →Use <column><asp: Bound Field…>to define all columns→Inside Grid View write this code

```
                    <asp: Grid view id="gridview1" runat="server">
(//open column//)    <Column>
                    <asp: Bound Field Data Field="Job_id" Header Text="job_id"/>
                    <asp: Bound Field Data Field="job_desc"Header Text="job_desc"/>
                    <asp: Bound Field Data Field="min_lvl" Header Text="Min_lvl"/>
                    <asp: Bound Field Data Field="Max_lvl" Header Text="Max_lvl"/>
                    </Column>
```

//Then give the server connection Code in the" Code view window"//

Ex: Using System. Data.SqlClient.

```
(The data connection code)
 {
  Sql.Connection cn=new Sqlconnection ("user id=sa; database="… "; data source="server");
  cn. Open ( );
 Dataset ds=new Dataset ( );
 Sql Data Adaptor da=new SqlData Adaptor ("select * from "table name");
 Data Fill (ds,"tablename");
 Grid view 1.Data source=ds.table [0];
 Grid view 1.Data Bind ( );
 }
```

**Note:

-Bound Field is better and also only used to display some static data like Data Object Field.

-If we want to display any "Check Box" ,"Link Button", "Button" or others and provide action to it then we would prefer to use template field.

-Template Field has templates that allow user to define own templates or some custom data .it can be used like a bound field also.

Note:

-One Template Field means one column, even though there are 100 columns in Template Field.

Ex:

<asp: Template Field>
<Header Template>select</Header template>
<Item Template>
<asp: Label id="lbl 1"text='<%#Eval ("job-id") %>'runat="server"></asp: Label>
</Item Template>
<Item Template>
<asp: Listitem>interested</asp: ListiItem>
<asp: List item>notintrested</asp: Listitem>
<asp: List item>buy later</asp: List item>
</asp: Drop Down List>
<Item Template>

-**To create a data Grid view to Update, Edit, Cancel the database tables we have to follow these steps**

The Code to write inside the Data grid Source Code is as below:

```
<asp: GridView ID="GridView1" runat="server" Height="376px" Style="z-index:
100; left: 8px;
position: absolute; top: 128px" Width="592px">
<Columns>
<asp: BoundField DataField="job_id" HeaderText="jobID" ReadOnly="true" />
<asp: BoundField DataField="desc" HeaderText="Description"/>
<asp: BoundField DataField="min" HeaderText="MinLvl"/>
<asp: TemplateField>
<ItemTemplate>
<asp: TextBox ID="txtmaxlvl" Text='<%#Eval ("max_lvl") %>' runat="server">
<EditItemTemplate>
<asp: DropDownList ID="ddlmaxlvl" runat="server">
<asp: ListItem>100</asp: ListItem>
<asp: ListItem>200</asp: ListItem>
<asp: ListItem>300</asp: ListItem>
</asp: DropDownList>
</EditItemTemplate>
</asp: TextBox>
</ItemTemplate>
</asp: TemplateField>
<asp: CommandField ShowEditButton="true" ShowCancelButton="true" />
</Columns>
</asp: TemplateField>
</asp: GridView>
```

The Code to write in the View Button Click Code is below:

```
protected void Button1_Click (object sender, EventArgs e)
{
   {
      BindData ();
   {
       void BindData ();
   {
          SqlConnection=new SqlConnection ("User id=sa, database="pubs);
          cn.Open ();
          Dataset ds=new Dataset ();
          SqlData Adaptor Dataset=new SqlDataAdaptor ("Select * from job", cn);
          Data. Fill (ds,"jobs");
          GridView1.DataBindSome=ds.Tables [0];
          GridView1.DataBind ();
     }
     }
     }
   }
}
```

-Then run and click on View Button we have to get the result else once again check the code and try.
-or type the following aspnet_regiis-I in Visual Studio Command prompt then enter we have to get a message like started installing and then again start run the programmed.
-Then select Grid view go to properties then enter select events and then select Row Editing then double click on the that and write the code as follows.

```
 {
    Gridview1.EditIndex=e.New EditIndex
    BindData ();
 }
```

Select Row Cancelling Edit then double-click and write the following code.

```
   {
    Gridview1 EditIndex=-1;
    Bind Data ();
   }
```

Note: ”e” used to indicate the particular content.

Select Row Updating and double click and write this code

```
 {
   Response. Write (Gridview1.Rows [e.RowIndex].Cells [0].Text;
   Textbox tjobdese, tminlvl;
   DropdownList dlmaxlvl;
   Tjobdesc= (TextBox) Gridview1.Rows [e.RowIndex}.Cells [1].Cells [1].controls [0];
   Tminlvl= (TextBox) Gridview1.Rows {e.RowIndex].Cells [2].Controls [0];
   Dlmaxlvl= (DropdownList) Gridview1.Rows [e.RowIndex].Findcontrol (“ddlmaxlvl”);
   Response. Write(s);
   Response. Write (tjobdesc.text);
   Response. Write (tminlvl.text);
   Response. Write (dlmaxlvl.Selecteditem.Tostring ());
  //We can also write code to Update db.after Updation.
    Gridview1.EditIndex=-1;
```

Bind Data ();

Note: To Check the commands.

-to get the data to the dropdown list from another table.

-take Sql Data some and drop then configure data source.

-connection strings then take the tables then check column which we want

-then go to source and then remove drop down list and in DropdownList, Data source ID="Sql Data Source1="Data

TextFill="qty".

<asp: Dropdown List Id="ddlmaxlvl" runat="server" Datasource Id="Sql Data Source 1" Data Text Field="qty">

ERROR HANDLING /RUNTIME DEBUGING

Every web application unlike desktop application needs effective error handling because web is involved with http, web servers, database servers, security servers etc..,ASP .NET being server side technology for web applications provides plenty of support to debug an application the concepts of ASP.NET that supports these are.

1. Exception handling using trycatche.
2. Page level errors using "Page_error event".
3. Web.config custom errors.
4. Application _error event of Global.asax file.
5. Debug class and debug windows (F11, F10).
6. Tracing.

1. Try cache: is called as method or statement level error handling.

Ex: right click in the code view and select snippet then Try.

```
{
  Cache (Exception)
 }

{
Response. Write ("invalid action performed");
}
 Finally { }
```

NOTE: this is a statement level error handling.

-all the code should start with try cache.

2. **Pagelevel Errors using Page error event**

The Page_error fires when the statements in the page or not handled for errors.

Protected Void_Error (Object s, EventArgs e)

// Code gets fire when some statements of this page raise an error and that are not handler.

3. **Web.Config Errors**:

-go to web.config and try this then only in the local system only we get the errors not in the client systems.

To reduce the load on the web page and to get a better performance.

<Custom errors mode="Local only">

Write in "web.config"

Web.config errors

Mode= when "ON" Errors are handled local and remote.

= When "OFF" Errors are handled but remote only.

Ex:

<Custom Errors mode="ON" default redirect="~/Error Page.html"/>

Default Redirected=<Programme>

The page i.e. served when a runtime error occurs and when it is not handled by below levels.

<Custom errors mode="ON" default>

ASP .NET SECURITY CONFIGURATIONS Date: 9/8/07

- Security in ASP .NET refers to authenticating users and providing authorization on resources.
- Authentication means verifying weather user is valid or not, in other words checking for right credentials is authentication.
- In ASP .NET we have authentication methods like "windows passport" forms. Windows and passport authentication methods are not performed by ASP .NET and also are less important as for as web applications are concerned.
- Forms authentication is performed by ASP .NET only .and mostly implemented in web applications.
- Every resource in ASP .NET is accessible only to authenticated users.
    i. Authentication
    ii. Authorization

Both are processors (or) programmer only.

(i) Authentication :

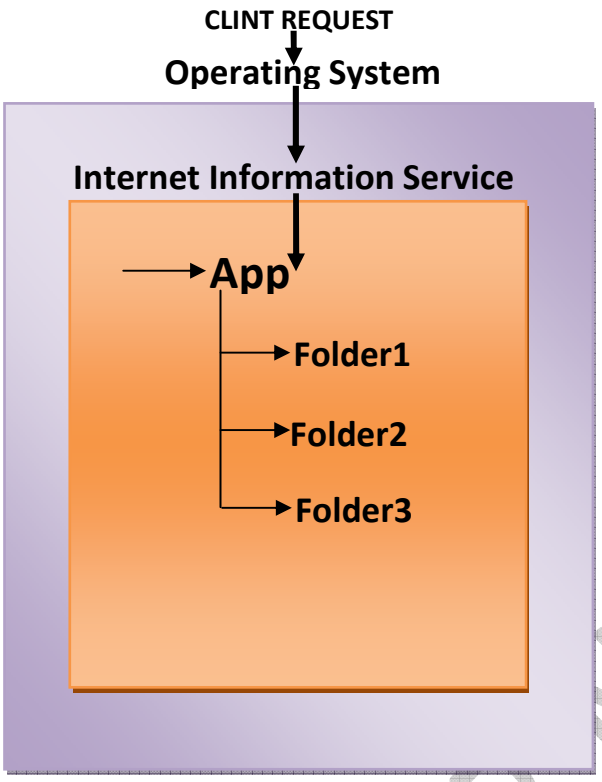
Authentication means getting permission to use the account is called authentication. Users can have the permission to participate in the developing or to use the application but he cannot change or modify the data base.

EX: if we have an account in a bank we are authenticated to use the services of the bank but we don't have the authorization to modify or to change the data base or to get the transaction info of other account holders.

(ii) Authorization:

It is a process where we deny or allowed users for ASP.NET resources.

- Every user normally authenticated first and the authorized in the development.
    i. Windows
    ii. Passport
    iii. Form(ASP .NET)



In the above diagram user should full fill all the security permissions step by step.

Process involved in the Authentication

1. User makes a request for a "secured page".
2. IIS stops the user and redirects him to "OS" or the browser itself displays a dialogue box like "Pleases Enter Username and Password".
3. "OS" displays a "Logon window" asking for user credentials.
4. User enters the required credentials and submits the same to OS and OS checks for Username/Password and if it is "TRUE" creates a "Ticket" and redirects to IIS.
5. IIS allows user now because he holds an" Authentication Ticket".

Note: The created "Ticket" will travel between request and response using that so that every time user is not asked for it.

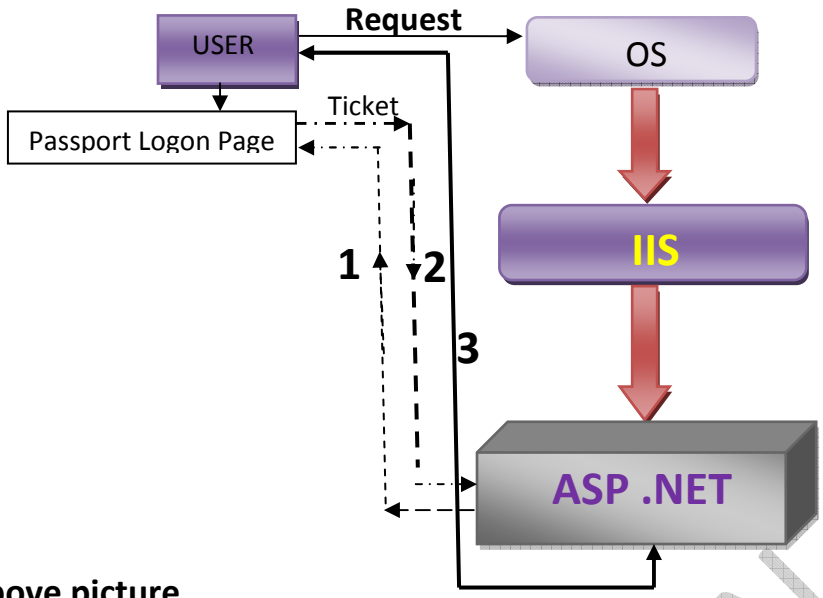
Passport Authentication

Note: NO more "Passport.com" but it changed as "Live.com".this is a third-party web provides authentication services and this is a Microsoft service and can be easily incorporated in ASP .NET.

Steps involved in Passport Authentication:

1. User makes a request for a "secured page".
2. IIS allows user as an anonymous.
3. ASP .NET denies request because user doesn't have "Passport Ticket"and it redirects him.
4. Passport provides a login page where user enters his passport credentials and passport verifies the same and creates a ticket. Also redirects back to the request secured source.
5. ASP .NET allows the user now because the user holds an authentication ticket.

Note: Once a passport ticket is available with the client he can visit all the similar passport enabled sites.



In the above picture

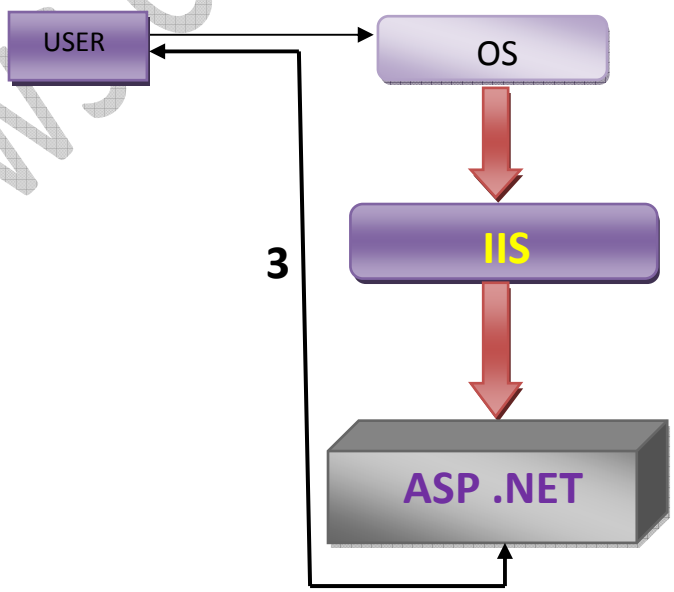
- **The process 1= ASP .NET Redirects the user to a passport logon page.**
- **The process 2= User gets the Passport Ticket by entering necessary credentials.**
- **The process 3=ASP .NET allows user to the required web page.**

Forms Authentication:

1. User request for a secured page.
2. User enters IIS as anonymous.
3. Now ASP.NET (checks whether) has FAM="Form authentication Module" that checks whether user is authenticated or not.
4. If user is found not to be authenticated, it redirects him to login.ASPX page which is in the same site.

NOTE: In form/passport authentication what is the mode of entry? Ans: Anonymous.

<Authentication mode = <windows/passport/form> />



In the above picture

PROCESS 3=ASP.NET Checks and Redirects the user to the logon page which is in the same web page.

N-TIRE MODEL OF WEB BUILDING:

-This is the method that all the real-time developers are using to give more secured page and to simplify the process of modification in a webpage.

-The process of data flow in this is as like shown below
Login.aspx—ClsUsers.cs---DBActions.cs---User Table.

Process:

i). user will be redirected to the login.aspx then if the user enter the username and password then ClsUsers.cs will force the DBActions to check weather the values are true or not then DBActions will check in the database table and give the response to ClsUsers.cs then if the values is true then asp.net will produce a ticket and allows user to go advance to the next page.

Handling cookies in ASP .NET

How to create a cookie, how to get the value stored in a cookie, set the lifetime, path and domain for a cookie, edit a cookie, delete a cookie, remove subkeys from a cookie...

Here's a tutorial that shows you how to use cookies in ASP .NET. I'm not going to explain the role of cookies in web applications or cover any other theoretical aspect of cookies. There are many (similar) ways to handle cookies in ASP .NET. I'm only going to show you one of the ways, my way. Oh, and we're going to use C#, although the code can be adapted to Visual Basic .NET easily.

How to create a cookie.

Here's a new cookie named *cakes*.

```
HttpCookie myCookie = new HttpCookie("cakes");
```

We created the cookie but there are no keys with values in it, so for now it's useless. So let's add some:

```
myCookie.Values.Add("muffin", "chocolate");

myCookie.Values.Add("babka", "cinnamon");
```

We also need to add the cookie to the cookie collection (consider it a cookie jar):

```
Response.Cookies.Add(myCookie);
```

How to get the value stored in a cookie.

Here's how to get the keys and values stored in a cookie:

```
Response.Write(myCookie.Value.ToString());
```

The output to using this with the previous created cookie is this: "muffin=chocolate&babka=cinnamon".

However, most of the time you'll want to get the value stored at a specific key. If we want to find the value stored at our *babka* key, we use this:

```
Response.Write(myCookie["babka"].ToString());
```

Set the lifetime for a cookie.

You can easily set the time when a cookie expires. We'll set the **Expires** property of *myCookie* to the current time + 12 hours:

```
myCookie.Expires = DateTime.Now.AddHours(12);
```

This cookie will expire in twelve hours starting now. You could as well make it expire after a week:

```
myCookie.Expires = DateTime.Now.AddDays(7);
```

Also note that if you don't set a cookie's expiration date & time a transient cookie will be created - a cookie which only exists in the current browser instance. So if you want the cookie to be stored as a file you need to set this property.

Setting the cookie's path.

Sometimes you'll want to set a path for a cookie so that it will be available only for that path in your website (ex.: www.dotnetsparkles.com/Careers). You can set a cookie's path with the **Path** property:

```
myCookie.Path = "/forums";
```

Setting the domain for a cookie.

Perhaps instead of using http://www.dotnetsparkles.com/Careers path style to your forums, you would use a subdomain like http://Careers.dotnetsparkles.com. The **Domain** property should do it:

```
myCookie.Domain = "forums.geekpedia.com";
```

How to edit a cookie.

You don't actually edit a cookie, you simply overwrite it by creating a new cookie with the same key(s).

How to destroy / delete a cookie.

There's no method called *Delete* which deletes the cookie you want. What you can do if you have to get rid of a cookie is to set its expiration date to a date that has already passed, for example a day earlier. This way the browser will destroy it.

```
myCookie.Expires = DateTime.Now.AddDays(-1);
```

How to remove a subkey from a cookie.

This is one of the problems I encountered with cookies. Fortunately I found an answer on MSDN. You can use the **Remove** method:

```
myCookie.Values.Remove("babka");
```

However, you don't usually remove a subkey immediatly after creating it, so first we need to retrieve the cookie, remove the subkey and then add it back to the **Cookies** collection:

```
// Get the cookie from the collection (jar)

myCookie = Request.Cookies["cakes"];

// Remove the key 'babka'

myCookie.Values.Remove("babka");

// Add the cookie back to the collection (jar)

Response.Cookies.Add(myCookie);

// See what's in the cookie now

Response.Write(myCookie.Values.ToString());
```

Of course I suppose you used the code we created earlier (the one with the chocolate muffin and the cinnamon babka), therefore if you test the code now you'll see the result is 'muffin=chocolate' - we got rid of the babka!