Caching

(To avoid loss of time)

* Caching is a process for storing whole data or part of data in local memory temporarily.

Types of cache:

There are three types of cache.

1. Page Output Caching
2. Page Data Caching
3. page fragment Caching.

* Page Output Caching:

In this process, the entire page or data will be cached for some time.

Eg. pgm 1:

```
Imports System.Data.SqlClient
Dim Con as New SqlConnection
Dim Cmd as SqlCommand
Dim dr as SqlDataReader
        Page_load
Con.ConnectionString = "uid= ; pwd= ;"
        B_click
Con.Open()
Cmd = New SqlCommand("Select * from emp", Con)
dr = Cmd.ExecuteReader()
dg1.DataSource = dr
dg1.DataBind()
Con.Close()
```

01 → Tom
~~13~~
15
~~Parent Cancel~~
NCE → 20
26 → Tom
~~Parent cancel~~ → 28

30 → Tom
37 → Tom
40 → ACC
44 → Tom
46 → Tom
47 → Tom
51
52

Scanned by CamScanner

a1.Html

```
<%@ OutputCache Duration = "300" VaryByParam = "None" %>
```

Eg prgm 2:

```
Imports System.Data.SqlClient
Dim Con as NewSqlConnection
Dim Cmd as SqlCommand
Dim dr as SqlDataReader
Dim x as Integer
Dim y as String
        Page_Load
Con.ConnectionString = "....." ;
        B_click
Con.open()
  x = Request.Params ("dno")
if x is nothing then
   x = 10
end if
  y = "select * from emp where dno = " & X -
Cmd = New SqlCommand (y, Con)
   dr = Cmd.ExecuteReader()
      dg1.dataSource = dr
        dg1.DataBind()
          Con.close()
```

Scanned by CamScanner

* Page Data Cache:

The process of caching only data, is called Data cache. Cached data are stored in local memory or cache memory.

User are permitted to cache one item, more than one data item, dataset, & Arrays.

To cache the data item, the following coding will be used.

```
Cache.Item("x") = "welcome"
```

Here x = name of cached data Item welcome is the value of cached data item. The same process mentioned above will be done in another way also.

```
Cache("x") = "welcome"
```

To retrive the value stored in the cache data item, we use the following coding

① => Response.write (cache("x"))

② =>
```
Dim y as String
y = Cache("x")
```

The following coding is used to remove cache item from local memory.

```
Cache.remove("x")
```

In above statement, we used pre-defined method 'remove()' for deleting cached data Item from local memory.

Scanned by CamScanner

Using this method, user can delete only one item from the local memory. To delete all cache data items from memory at a time, we used pre-defined method 'RemoveAll()'

The statement Cache.RemoveAll() will delete all cache item from the cache memory or local memory.

* Page fragment Caching:

In this method, we cache only part of the webpage to accomplish this task, we need user controls. User controls have the file extension .ascx. User controls have designing window, html window & programming window by default.

Eg. pgm

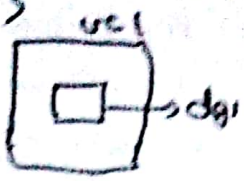
usc1.ascx

```
Imports System.Data.SqlClient
  Dim Con as New SqlConnection
  Dim Cmd as SqlCommand
  Dim dr as SqlDataReader
        Page_load
Con.ConnectionString = "uid= ; pwd= ;"
        B_click
Con.open()
Cmd = New SqlCommand("Select * from Kavithi",Con)
  dr = Cmd.ExecuteReader()
    dgl.DataSource = dr
      dgl.DataBind()
      Con.close()
```

Scanned by CamScanner

UC1.html → Caching the entire control for 5 min

<%@ Output Cache Duration="300"
VaryByParam="None" />

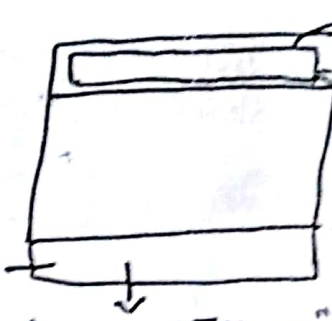


a1.aspx

a1.html

<%@ Register TagPrefix="x" TagName="y"
SRC="http://localhost/uc1.ascx" %>

Add Rotator



KeywordFilter="Top"
Advertisement File="x1.xml"

KeywordFilter="Bottom"
Advertisement File="x2.xml"

Notes:

* User are not permitted to cache part of the webpage directly.
* Cache the part of the webpage can be implemented indirectly only.
* The Control 'AdRotator' is used to display an advertisement in our webpage.
* The property 'KeywordFilter' of this control is used to fix the location in which this control will be displayed.
* The property 'AdvertisementFile' of this control is used to include or specify the name of the advertisement file.

Scanned by CamScanner

Test-8

1. Discuss proxy class
2. Discuss Web Services & data set
3. Explain Page Output Cache with example
4. Discuss Data Cache
5. Write about page fragment Cache with an example

3/9

Handling Errors in Asp.net

Concept:

Execution of any application may have generate errors due to mistakes made by programmer. There are four types of error in Asp.net. They are given below:

1. Configuration Error
2. parser Error
3. Compilation Error
4. Runtime Error

★ **Configuration Error:-**

Mistakes in the Web.Config file will generate Configuration Errors. By default, all .net applications have one web.config file. Refer Eg.2

Eg:

VB Script

```
<Script Runat = "Server" />
Sub page-load
    Dim l as new label
        l.Text = " Hello"
End Sub
```

predefined class

used to generate labels

No </Script>
So there is an error.

~~Notes~~:

*

Eg 2:

```
< Configuration >
    < System.web >
        <add key: "x1" value = "user id = sa, pw = 1230wE..." />
    < /System.web > — without this is an error
< / Configuration >
```

Eg 3:

```
< Script Runat = "Server" />
    Sub page-load
        Dim l as New Label
        l.value = " Hello "
    End Sub
< /Script >
```

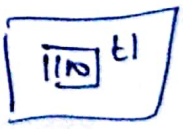
No - there is no property value for label So-error.

Eg 4:

```
< Script Runat = " Server" />
    Sub page-load
        Dim t1 as TextBox
            t1.text = 100
    End Sub
< /Script >
```

To generate it needs initialization

So - error



* Parser Error:-

Mistakes done in Syntax generate parser error. Refer Eg. 1

In that example, the tag Script is not closed.

So, it generate parser Error

Scanned by CamScanner

* Compilation Error:

These Errors will be generated due to making ~~mai~~ mistakes in program. Refer Eg. 3:

In that program, the property 'value' is not valid property of label. Thats why program generate Error.

* Runtime Error:

It will be generated due to making mistake in the logic of the program. Refer Eg. 4:

In that program, the object of Textbox is not initiated. Thats why program generate Errors.

Notes:

* User are permitted to write Events in HTML Window of any .Net Application.

* All Events in HTML window should be written within Script tag.

* Inside the <Script>, user can permitted to write events in VB Script, Java Script, & other Scripting Language also.

* In Web.Config file, all tags will be closed properly.

* whatever logic or value define in web.Config file is visible or is accessible to all web pages of particular Application.

Enabling Error Messages:

By default .net Application don't Show the error messages to the user. To Show the error message to the user, the attribute 'debug' is used. Users are permitted to enable error messages in a particular webpage & also user is permitted to enable error messages in particular applications

Eg 1:

```
<%@ pagedebug="True" %>
```

By default, debug attribute have false value at a pagelevel assigning value 'True' to debug. Attribute help the users to view all error messages.

Eg 2:

```
web.Config
<configuration>
<System-web>
<Compilation debug="True" />
</System-web>
</Configuration>
```

The above program enables error messages to all web pages. it means the debug is enable to all webpages of particular .net applications.

Pagelevel Error Handling:-

To handle the error at page level. User Can use the try, Catch Statement.

Scanned by CamScanner

Syntax:

```
Try
    exe. statement ] B1
Catch
    exe. statement ] B2
finally
    exe. statement ] B3
End Try
```

B1 & B3 are used to write executable statements.
B2 is used to display user defined error messages.

Logic:

If try block have any error, control transfer to Catch block if there is no error in try block, control don't go to Catch block irrespective of errors in the program. Finally block will be executed always.

Eg:

```
Imports System.Data.SqlClient
Dim Con as New SqlConnection
Dim Cmd as SqlCommand
Dim dr as SqlDataReader
                Page_load
Con. ~~open~~ ConnectionString = " ";
Try
    Con.open()
    Cmd = New SqlCommand("Select * from emp", Con)
    dr = Cmd.ExecuteReader()
    dg1.DataSource = dr
    dg1.DataBind()
Catch
    Response.write("welcome to Catch Block")
finally
    Con.close()
End try.
```

Scanned by CamScanner

Exception class:

Exception is a pre defined class. This class is used to display errors messages generated by .net applications. This class contains 4 properties. They are given below

1. Message
2. Object
3. StackTrace
4. TargetSite

* Message:

This property returns a String which represents error message exactly.

* Object:

This property returns a String which represents the object of the class in which errors are occured.

* StackTrace:

This property returns the name of method which is executed successfully before errors are occurred.

* TargetSite:

This property, returns the name of method in which errors are occurred exactly.

~~Ex: Pgm~~

```
Imports System.Data.SqlClient
Dim Con as New SqlConnection
Dim Cmd as SqlCommand
Dim dr as SqlDataReader
```

Scanned by CamScanner

Page-load

```
Con.ConnectionString = " uid= ; pwd= ;"
Try
    Con.open()
    Cmd=New SqlCommand ("Select * from emp", Con)
    dr = Cmd.ExecuteReader
      dg1.DataSource=dr
      dg1.DataBind()
Catch x as Exception
      Response.write (x.message)
      Response.write (x.Sourse)
      Response.write (x.StackTrace)
      Response.write (x.TargetSite)
Finally
      Con.close
end Try.
```

Note:

* In Try...Catch block, Exception class doesn't wanted to initiate by "New" operator.

* ~~Applica~~

Application Level Error Handling:

User are permitted to handle errors at page level & Application Level. To handle the errors at page level, we use try...catch Statement.

To handle the errors at application level, we use the method Redirect() & defaultRedirect()

Eg:

a1.aspx

web.config

In the web.config file of this application, we done the following updations for catching the errors at application level.

Scanned by CamScanner

```
<Configuration>
<System.web>
  <CustomErrors mode="On" defaultRedirect="a.aspx">
<Error StatusCode="404" redirect="a.aspx" />
</System.web>
</Configuration>
```

Notes:

* The property 'Status Code' refers no. of the error message. The property 'Redirect' is used to transfer control from source file to destination file.

* The property 'Default Redirect' also is used to transfer control from source file to destination file.

* By default, all error messages have error code or number.

* To ~~define~~ display user defined errors, use 'mode' property of Custom Errors should have value 'ON'.

* When use 'mode' property of Custom Errors have the value 'OFF', it permits .Net Application to generate it is no error messages.

* Whenever error code or error number generated by .Net, is not matching or not founding list of user defined Error messages, 'default Redirect' property will be trigger automatically.

* If Error Code or number generated by .Net is ~~not~~ ~~not~~ found match with list of user defined Error messages, 'redirect' property will be invoked by .Net.

Scanned by CamScanner

Tracing the Application

By default, all .net applications generate ~~trace~~ Tracing Information. These information are used to know how our .net applications are executed step by step during the runtime.

It contains the following important components.

1. Request Details:

This component contains the information about user id, the time, user entered the application, the time user leave the application, etc;

2. Trace Information:

This section contains the information about step by step execution of .net application.

3. Control Tree:

This section contains the information which describes hierarchy of controls placed on user interface.

4. Session State:

It is used to know about data items stored in user session.

5. Application State:

It is used to know about data item stored in application level.

Enabling Trace at page ~~Load~~ level:

To enable trace at page level, user should do the following updation at page entity or tab

Ex. @ page Trace = "true" ... />

By default, the property 'Trace' of page entity have the 'True' value.

Scanned by CamScanner

To enable Trace at application level, we should do the following modification in web.config file of particular application.

```
<Config>
  <System.web>
    <compilation Trace="True"/>
  </System.web>
</Config>
```

Note:

By default, Tracing Information will be displayed in Black Color.

Adding User Information in Trace:

Users are permitted to display user defined messages in Trace. It also permitted to display result of any calculation, or value stored in any textboxes. To display user messages in trace, we used two pre-defined methods such as,

Trace.warn()

Trace.write()

Trace.warn()

This is a pre-defined method. This method is used to display user messages inside the tracing information. This method display user messages in Red Color.

Trace.write()

This is also a pre-defined method. This method is used to display user messages inside the

tracing information. This method display error mes
in Black color.

Using both methods, user also can display error messages. User is permitted to use both methods inside the catch block. By default, user is permitted to use both methods in try block.

Trace and webpage:

By default, all .net applications have this webpage. This webpage contains the list of request made to website by user. By default, this webpage contains only last 10 request of user. User is also permitted to change the no. of request made by user.

Ex:1

B1-click

Response.write("welcome")
Trace.warn ("welcome")

B2-click

Trace.write ("Hello")

Notes:

* In the above program, response.write method will display the message 'welcome' above (or) below the tracing information.

* In the above program Trace.warn method display the message 'welcome' inside the trace information section in Red color. Likewise Trace.write method also will display the message "hello" inside the tracing information Black color.

Scanned by CamScanner

Ex: 2:

B_Click

```
Try
    Dim t1 as TextBox
        t1.Text = "Hello"
Catch
    Trace.warn("Error 1")
    Trace.write("Error 2")
    Response.write("Error 3")
end try
```

Note:-

From the above program, we receive the following observations.

* The object t1 is declared, but it is not initiated.
* The memory allocation process will be held during runtime only.
* The text property of textbox is used to assign value to a textbox. It also used to fetch value enter in a textbox.
* Try....Catch may (or) may not have finally block. The user defined error message 'Error 3' in the above program will be display above (or) below the tracing infor area.
* The error messages such as Error 1, 2 in the above pgm will be displayed inside the tracing info. The message 'Error 3' will be displayed in Red color where as the message 'Error 2' will be displayed 'Black color'.

Scanned by CamScanner

Ex: 3

[http://localhost/a.aspx/Trace.axd when executing the above address, will list out the no. of requisation make to website named a.aspx by user.]

Notes:

* Any Clients are permitted to access websites Stored in the Server.
* [The Execution of above address help us to know the user details who send requisation to our application.]

Monitering .Net Application:-

[User are also permitted to moniter any .net applications.] This process is used to measure the performance of our application.

Diff b/w Asp & Asp.net (migrating)

There are n. no. of diff exist b/w Asp & Asp.net. The important difference are given below:

* File Extension → Asp Application have .asp file extension whereas Asp.net application have .aspx file extension.
* Option Explicit:- It have value 'off' in asp whereas option explicit have value 'on' in asp.net. The possible value for option explicit are 'on' & 'off' only. If this option have value 'off' user are permitted to skip the declaration of any variable.

Scanned by CamScanner

Suppose this option have value on, user must declare variables before using in the program.

Ex:

In Asp

```
Str = "welcome"
Response.write (Str)
```

Asp.net:

```
Dim Str as String
  Str = "welcome"
Response.write (Str)
```

It is possible to bring this feature in asp.net by doing the following modification at pagelevel.

```
<%@ page explicit="off" .... />
```

when our webpage have value 'off' to attribute explicit of page entity help the user to avoid for declaring variables in this pgm.

* Variant:

Asp support feature variant whereas Asp.net not support feature variant. The possible value of variant are 'True' or 'false'. By default, the option variant have 'True' value in asp ~~values~~ whereas this option have 'False' value in 'asp.net'. This option also play like option Explicit if variant have value True, no need to declare variables. It is necessary to declare the variables when option variant have the value 'false'.

Scanned by CamScanner

Ex:

Asp	Asp.net
Dim str Str = "welcome" Response.write(str)	Dim Str as String Str = "welcome" Response.write(Str)

The following coding help the user to implement this feature in asp.net.

<%@ Page Strict="True" .... />

* Paranthesis:

In asp.net user must pass (or) must have paranthesis. In asp, parenthesis are optional.

Ex:

Response.write "Hello" → ASP

Response.write("Hello") → ASP.Net

* Function:

In Asp, function should be written within <% .... %>. In Asp.net function should be written within <Script> ..... </Script>

Ex:

Asp

```
<%
    function f1 (ByRef x1, ByRef x2)
      Dim x3 as Integer
        x3 = x1 + x2
      Response.write (x3)
      end function
%>
```

Scanned by CamScanner

Asp.net

```
<Script runat="Server"/>
  function f1(Byval x1, Byval x2)
    Dim x3 as Integer
      x3 = x1+x2
    Response.write(x3)
    End function
<(Script)
```

In asp arguments are referred by keyword 'Byre
In Asp.net arguments are referred by the keyword
~~'Byval)~~ 'Byval'.

* Disconnectivity Environment:

Asp does not Support D.E whereas Asp.net Support D.E.

* Language Support:

webpages of Asp will support more than one Scripting languages wherea webpages of asp.net will Support only one Scripting languag at a time.

Asp → Supports many langu
Asp.Net → Supports only one "

Test-9

1. Errors in .Net
2. tracing .Net application
3. webServices
4. Sorting Operations
5. Pgm to Create DataTable program

Scanned by CamScanner

Performance Counter

Performance Counter are pre-defined Variables. Those Variables are used to measure the performance of .net applications. To View the performance Counter, user open the administrative Tools. Values to performance Counter will be assigned by .net application automatically.

Types of performance Counter:

There are two types in performance Counter. They are:

1. System performance Counter
2. Application performance Counter

* System performance Counter:

Those Counters are used to measure the performance of every executed.net application. It means those Counter Contains information about every executive .net applications in a System.

Eg: Here we discuss few Important Counters of System performance.

1. Application Restarts:
   → used to know the number of applications executed since internet information Service (IIS) was last Started.

2. Application Running:
   → used to know how many applications are executed Successfully since Services is Started.

3. Requests Disconnected:
   → used to know the number of request that were disconnected due to a Communication failure.

Scanned by CamScanner

4. Request Queued :-
→ used to know the no. of request that are waiting in a queue.

5. Request Wait Time:
→ used to know the amount of time is needed to ~~execute~~ stay in a Queue before execution

* Application Performance Counter :-
They are used to measure the performance of particular application. It means, those Counters Contains information about that particular application. The important Counters are given below:

1. Errors during Compilation:
→ used to know that the no. of errors occured during Compilisation.

2. Errors during Execution:
→ used to know that the no. of errors occured during Execution.

3. Debug Requests:
→ used to know that no. of request make to application while debugging take place.

4. Session Total:
→ used to know that the no. of Sessions as Created.

5. Request Total :-
→ It Contains Some value which

Scanned by CamScanner

represent the no. of request made to that application.

6. Request / Sec :-

→ used to know that how many request will be ~~Sec~~ executed per second.

Debugging

Concept:

Debugging process is use to measure line by line execution of any application. To perform debugging process, we need debugger. To implement debugging in our prgm, we use the breakpoint. Breakpoints appear in red Solid dot. User allow to make breakpoint On left Side Margin. User is also allowed to make more than one break point ~~is also~~ in our application. When execution reach the breakpoint, it entered to Breakmode, at this point, execution will be stopped & also at this point, we are allowed to view values of any Variables. To restart the execution, we use the double-click.

Watches

To implement line by line execution. It can also use Quick Watch Dialog Box. This Box allows the user to add Variables declared in the program. User Can View the values of those variables during the execution of program. In this manner, user Can't make break point.

Scanned by CamScanner

Overview of .net App

[All .net app will be stored in local Server] asp.net app have the file extension .aspx. [The local Server will be referred as IIS]. It is allowed to user to execute more than one app at a time. [By default .net app have only webpage. User is permitted to include no. of webpages to any .net application].

* [Configuration Settings & codings of one app don't affect the another app, even both app are stored in Same Server]

* Closing process (or) Shuting down process of one app doesn't shutdown the another app.

* [Codes written in one app will don't affect the another app which is also stored in Same local Server.

[By default, all .net app have /bin directory, whenever Controls placed in this directory can be accessed by any webpages of particular .net app. In /bin directory, user can also place the DLL file (Dynamic Link Library). All .net app have Global.asax by default. By default, all .net have web.Config file)

Application State

To declare variables & object, we need app state variables (or) object declared in App State get the public access specifier, it means global declaration of variables & object. User are allow to declare more than one variable (or) object in Application State.

Scanned by CamScanner

Web.Config file

[Web.Config files are human readable 'xml' file. are permitted to notify content of the file. There are 2 Configuration files such as:

* web.Config file
* machine.Config file.

Web.Config file is used for setting configuration to a particular application) where as machine.Config file as used for setting configuration to all applications stored in local server], important sections.

Important Section:

web.Config file have an no. of Config file. The important sections are discussed below:

* **Trace:** This section contains configuration settings for tracing the application.
* **Compilation:** This section contains configuration settings for compiling our .net applications.
* **Custom Errors:** This section is used to specify know error info. will be displayed possible values are 'On' (or) 'off'. The value 'On' refers error will be hidden from users, where as the value 'off' refers errors will be shown to the user.
* **Trust:** This section is used for setting security policies to our application.
* **Security policy:** It contains list of available security policies.
* **Session state:** It contains configuration info for session created where user using the application
* **web Services:** It contains the Config. info for web services used in our applications.

Ex:

```
<Configuration>
<System.web>
<Compilation debug="true" />
</System.web>
</Configuration>
```

In above example, we enabled debugging to all web pages in our application. It means error lines will be highlighted during runtime.

Scanned by CamScanner

```
<configuration> <system.web>
<customErrors mode="off"/> <compilation debug="true"/>
</system.web> </configuration>
```

Ex 3:

```
<configuration> <appSettings>
<add key="xyz" value="Server; uerid:; pwd:;"/>
</appSettings> </configuration>
```

In the above application, ConnectionString is stored in a variable called "xyz". Now user can use this connection String across all web pages in our application. To retrieve the value stored in variable 'xyz'. The following methods will be used.

Ex:

```
Dim x as String
x = ConfigurationSetting.appSettings("xyz")
```

user can also declare any Constant variable inside the tag <appSetting>... </appSetting>

Eg:

```
<Configuration>
<appSettings>
<add key="k" value="3.14"/>
</appSettings>
</configuration>
```

Executed program

```
Imports System.Data.SqlClient
Dim
}
Page_Load => Con
Button1 => ("Select * from Kartikeit", Con)
Button2 => ("Insert into Kartikeit values('5', 'Shyam', '5000', 20)", Con)
Button3 => ("Delete from Kartikeit where dno='10'", Con)
Button4 => ("Update Kartikeit set sal=sal+100", Con)
```

Global.asax

By default all .net applications have this file. In this file user is permitted to declare application wide objects. Normally this file is used to handle predefined Events. There are a number of pre-defined Events exists in .net. The important Events are given below:

1. Application_BeginRequest – This event is raised by every request to our Server. This event will be invoked each time user send a request to the Server.

2. Application_EndRequest – This event will be invoked at end of every Request to our Server. It means whenever user close the application, this event will be triggered.

3. Application_Error ; This event will be raised by unhandled errors occurred in our application.

4. Application_Start : This event will be invoked immediately after application is deployed in Internet. This event will be invoked only one time.

5. Application_End : This event also will be invoked immediately before end of all applications. This event will be invoked only one time to particular application.

Security in Asp.Net:

There are many ways to Secure our .Net apps. To secure a .Net application – .Net Environment have three important ways. They are given below:

1. Forms Authentication
2. Windows Authentication
3. passport Authentication

Forms Authentication:

It is also known as Forms based Authentication. It using this method, user can Secure applications efficiently. In this method, forms are used to ~~store~~ verify authorised users. In this method, username & passwords will be stored in database table (or) xml file. This method enable to store 'n' number of username & password. It also permit the user to apply Constraints on usernames. Here Constraints refer primary key Constraint, Not null Constraint

Ex2:

To enable form based Authentication, user must consider the following Coding in web.Config file of respective application.

Ex: all.aspx

web.config

```
<configuration>
  <System.web>
    <authentication mode="Forms"/>
  </system.web>
</configuration>
```

Note: * By default, forms Authentication is enabled to all applications.

There are 2 important elements of web.config file plays an important role to implement user, the element allow is used to permit user to access our Application. Element <deny> is used to Secure our application from unauthorized users. The Symbol '*' Specifies all users both authorized & unauthorized users. A Symbol '?' Specifies Only unauthorized users.

Eg:-

```
<Configuration>
  <System.web>
    <authorization>
      <allow users="*"/>
    </authorization>
  </System.web>
</Configuration>
```

Eg: 2

```
<Configuration>
  <System.web>
    <authorization>
      <deny users="?"/>
    </authorization>
  </System.web>
</Configuration>
```

Scanned by CamScanner

In example 1, the program allows both authorized & unauthorized users to access our application.

In example 2, we denied to access the ~~autho~~ application who don't have permission. The Symbol * refers both authorized & unauthorized users. The Symbol '?' refers only unauthorized users.

2. Windows Authentication:

Concept: It also refers windows based authentication. In this method, Security to our application will be done at operating System level / windows level. At windows level, Authorised & unauthorized users will be defined. Apart from that the role of the authorized user also will be defined at windows level.

At Application level, in this method, user enjoy the freedom.

Eg:

```
<configuration>
<System.web>
<Authentication mode="windows"/>
</System.web>
</configuration>
```

3. Passport Authentication:

Concept: By default, we need one username & pwd to access any website in Internet. To access more than one website to satisfy our expectation, we need to have more than one username, one password.

Scanned by CamScanner

As a User it is very difficult to remind more than one username & password. To remove this barrier exists in Internet world, microsoft provide passport Authentication Service. To use this Service, websites have agreement with microsoft Corporation. Those websites having agreement with microsoft accomodate the user to access information using only one username & password. In this process, whenever user enter the address of any website which has the dealing with microsoft, user is redirected to passport Authentication Server page of microsoft, instead of redirecting to home page of particular website. After entering User name, password in passport Authentication Service, user is redirected to homepage of website which he want to visit to access information. Those are not very close to the .Net processes.

Eg: Those are very close to the internet process.

```
< Configuration >
  < System.web >
    < Authentication mode = " passport " />
  < /System.web >
< /Configuration >
```

Note:

* In this process, user & password maintenance will be taking Care by microsoft Service provider.

Scanned by CamScanner

Unit-III

Web Services

Concept:

It is a place where user can develop web methods. A web service has the file extensions .asmx. User can create many web methods in a web service.

Web Method

Concept:

It is the collection of executable statements which are used to perform or complete a small task. The methods are implemented only on websites.

Syntax:

```
<Webmethod()> Function  <function name>(Argumet list) as <return data types>
                        Function's body
                End Function
```

<function name>	-	It should be user defined name which satisfies the rules for creating variables.
<argument list>	-	User specifies his arguments here.
<return data type>	-	user should specify the valid data type.

Example:

```
<webmethod()>Public Function squ (dim x as integer) As Integer
                        Return(x*x)
                End Function
```

Attributes of Web Services:

There are 3 main attributes in web services. They are given below.

Cache Duration	-	Used to cache the web services for some time.
Description	-	Used to describe the whole process of web service.
MessageName	-	Used to create alias name for web services.

Attributes of Web Methods:

There are 2 main attributes in web methods.

Description	-	Used to describe the web method.
Name	-	Used to determine the name of web method.

Scanned by CamScanner

Invoking Web Services

There are 3 methods to invoke the web methods of web services. They are given below.

Through Browser:

In this method, user must specify the location of web services in the address bar of the browser to invoke the web services. User can also invoke the particular web method in the web service. User can also pass values to the parameters of the web method through browser.

Example:

You assume that you have one web service with the name webservice1.asmx.

To invoke the webservices1.asmx, user has to write the following address in address bar of the browser.

http://localhost/webservices1.asmx

To invoke the method1 () in webservices1.asmx, user has to write the following address in address bar of the browser.

http://localhost/webservices1.asmx/method1

To pass values to parameters of web method in webservices1.asmx, user has to write the following address in address bar of the browser.

http://localhost/webservices1.asmx/method1? x = "5"

Through HTML

User can also invoke the web methods through Html also. There are two methods to invoke the web methods through Html.

get () Method:

Example:

One.HTML

```
<HTML>
    <BODY>
        <a href="http://localhost/webservices1.asmx">WebService</a>

        <a href="http://localhost/webservices1.asmx\mtd1,">Web Method1</a>

        <a href="http://localhost/webservices1.asmx\mtd1 ? X = "a"">Web
        Method with Parameters</a>
```

Scanned by CamScanner

```
        </BODY>
    </HTML>
```

Post Method:

Example:

One.HTML

```
<HTML>
    <BODY>
        <FORM Method = "Post" Action = "http://localhost/webservice.asmx"/>
            <INPUT Type = "Text" Value="" />
            <INPUT Type = "Submit" Value="Click" />
        </FORM>
    </BODY>
</HTML>
```

Invoking Web Services from desktop Applications:

In this method, we invoke the web methods of the web services from our Asp.Net application. User first executes the web services using Web Services Description Language (WSDL) tool which is used to create the VB class for the particular web services. Then the user execute the VB class file using VB compiler to create "dll" files of that particular VB class. Then the user should copy that "dll" file into the "bin" directory of the Asp.Net application where user needs to use the resources of the web services. "bin" directory is the default directory of any Asp.Net application. If there is no "bin" directory in Asp.Net application, user can create "bin" directory for the particular application.

Syntax for WSDL:

(ASP.Net DOS Prompt)

>wsdl.exe/l:vb<location of web service>?wsdl

Example:

>wsdl.exe/l:vb http://localhost/webservice1.asmx ? Wsdl

Syntax for compiling VB class using VB compiler:

>vbc /t:library /r:System.dll, System.web.Services.dll,System.Xml.dll ws1.vb>

Web Services and DataSet:

You assume that user has the web service webservice1.asmx where user returns the dataset as return value of the web method.

Scanned by CamScanner

Webservice1.asmx:

```
<webmethod()> public function get() As DataSet

        Dim con a New SQLConnection
        Dim cmd as SQLCommand
        Dim da as SQLDataAdapter
        Dim ds as DataSet

        con.ConnectionString = "user name:sa; password:sa;"
        con.open()

        cmd = new SQLCommand("select * from emp",con)
        da = new SQLDataAdapter(cmd)
        da.Fill(ds,"emp")
        return (ds)

    End function
```

In Asp.net application called one.aspx, user invoke the webmethod of webservice1.asmx.

One.aspx:

page_load()

```
Dim x as new ws1
        dg1 DataSource = x.get()
        dg1.DataBind()
```

XML web services and Arrays:

Web services can also returns the arrays. For returning the arrays, web method should have the string () data type as return data type.

Example:

For example, user has a web service called ws1.asmx. In this web service, user has only one web method which returns array.

ws1.asmx

```
Imports system
Imports system.web.services

<web method ()> public function get () as string ()
```

Scanned by CamScanner

```
        Dim arr (3) as string
        Arr (0) = "India"
        Arr (1) = "Pakistan"
        Arr (2) = "America"
        Return arr
End function
```

For accessing this web method in your desktop application, user first creates vb class to this web service. Second, user creates the dll files for vb class which created from this web service. Finally, user copies the dll files in \bin directory of the desktop application. Now, user can able to access the web methods of this web service from the desktop application.

For example, user has an application called one.aspx which contains the repeater control to display values of array. In this application, user has to write the following coding.

One.aspx

```
Page_load

Dim x as new ws1
Dim y as string ()
y=x.get ()
rpt.datasource= y
rpt.databind ()
```

In HTML part of this application, user has to write the following coding to display array's values in repeater control.

```
<html>
<body>
<asp: repeater>
        <itemtemplate>
                <li> <%# container.dataitem %>
        </itemtemplate>
</asp: repeater>
</html>
</body>
```

XML web services and Classes:

Web services can also returns the classes. For returning the classes, web method should have the class name as return data type.

Example:

Scanned by CamScanner

Now, User has one class called emp.vb

Emp.vb

```
Namespace mycomponents

Public class emp
        Dim eno as integer
        Dim ename as string
End class

End namespace
```

For example, user has a web service called ws1.asmx. In this web service, user has only one web method which returns class.

ws1.asmx

```
Imports system
Imports system.web.services
Imports mycomponents

<web method ()> public function get () as emp
        Dim x as new emp
        x.eno=100
        x.ename="kannan"
        Return emp
End function
```

For accessing this web method in your desktop application, user first creates vb class to this web service. Second, user creates the dll files for vb class which created from this web service. Finally, user copies the dll files in \bin directory of the desktop application. Now, user can able to access the web methods of this web service from the desktop application.

For example, user has an application called one.aspx which contains the repeater control to display values of class. In this application, user has to write the following coding.

One.aspx

Page_load

```
Dim x as new ws1
rpt.datasource= x.get ()
```

Scanned by CamScanner

rpt.databind ()

In HTML part of this application, user has to write the following coding to display array's values in repeater control.

```
<html>
<body>
<asp: repeater>
        <itemtemplate>
                <li> <%# container.dataitem.eno %> <br>
                <li> <%# container.dataitem.ename %>
        </itemtemplate>
 </asp: repeater>
 </html>
 </body>
```

SQL Data Reader

* Connection oriented Archi
* Execute Reader object to bind data to datareader.
* retreiving data, only one record brought into memory at a time
* user must explicitly close the datareader.

SQL Data Adapter

* Disconnected oriented Architecture
* It reads the data from database and binds that data to dataset
* It automatically closes the Connection.

Scanned by CamScanner