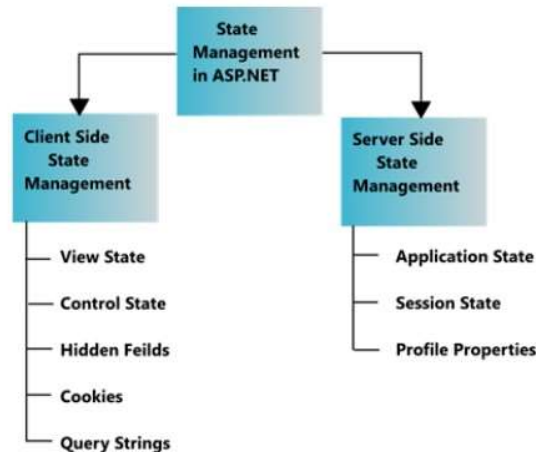


### STATE MANAGEMENT

- વેબ એ સ્ટેટલેસ છે. તેનો મતલબ એવો થાય છે કે જ્યારે જ્યારે વેબ પેજ સર્વર પર પોસ્ટ થાય છે ત્યારે દરેક ટાઈમ, વેબ પેજનો નવો ઈન્સ્ટન્સ રી-ક્રીએટ થાય છે.
- આપણે જાણીએ છીએ કે HTTP એ સ્ટેટલેસ પ્રોટોકોલ છે, તે પેજ પર ક્લાઈન્ટની ઈન્ફોર્મેશન સ્ટોર કરી શકતું નથી.
- જેમ કે, કોઈ ટેક્સ્ટ બોક્ષમાં આપણે ટેક્સ્ટ એન્ટર કરેલા છે અને હવે યુઝર સબમિટ બટન પર ક્લિક કરે તો પેજ પોસ્ટ બેક થયા બાદ તે ટેક્સ્ટ ડિસ્પ્લે થતા નથી કેમ કે તે પેજ રાઉન્ડ ટ્રીપ કર્યા બાદ રી-ક્રીએટ થાય છે.
- વેબ પેજ પરના બધા કંટ્રોલ્સ ક્રીએટ થાય છે અને સર્વર પરની રાઉન્ડ ટ્રીપ બાદ તેના બધા ઈન્સ્ટન્સ ડીલીટ થઈ જાય છે, તેથી બધા કંટ્રોલ્સની વેલ્યુ સાચવવા માટે સ્ટેટ મેનેજમેન્ટ ટેકનીકનો ઉપયોગ થાય છે.



- ઉપર આકૃતિમાં દર્શાવ્યા પ્રમાણે, જ્યારે જ્યારે પેજની રીક્વેસ્ટ કરવામાં આવે છે ત્યારે ત્યારે પેજ રી-ક્રીએટ થાય છે અને પછી ક્લાઈન્ટને મોકલવામાં આવે છે. તેથી તે એક મોટો ઈસ્યુ છે કે પેજનું સ્ટેટ અને વેબ પેજ પરની ઈન્ફોર્મેશન કેવી રીતે મેનેજ કરવી. તેથી સ્ટેટ મેનેજમેન્ટ ટેકનીકનો ઉપયોગ થાય છે.
- જ્યારે આપણે કોઈ વેબસાઈટને વિઝીટ કરીએ છીએ ત્યારે બ્રાઉસર સર્વર સાથે HTTP પ્રોટોકોલની મદદથી જરૂરિયાત પ્રમાણે કમ્યુનિકેશન કરે છે.
- પરંતુ રિસ્પોન્સ પછી, જ્યારે આપણે તે જ વેબસાઈટ બ્રાઉસરને ક્લોઝ કર્યા બાદ ફરીથી વિઝીટ કરી છીએ ત્યારે શું થશે? તે આપણે વિઝીટ કરેલી વેબસાઈટનું સ્ટેટ જાળવી રાખતું નથી કેમ કે આપણે બ્રાઉસર ક્લોઝ કરીને વેબસાઈટ વિઝીટ કરેલ છે એટલે. તેથી જ તેને સ્ટેટલેસ કહેવામાં આવે છે.
- જો આપણે યુઝરની ઈન્ફોર્મેશન જેવી કે એક પેજ પરથી બીજા પેજ પર જાય છે તે અથવા એક જ પેજની મલ્ટીપલ વિઝીટ ટ્રેક કરવી હોય તો આપણે ASP.NET ની સ્ટેટ મેનેજમેન્ટ ટેકનીકનો ઉપયોગ કરવો પડે છે.
- ASP.NET ની સ્ટેટ મેનેજમેન્ટ ટેકનીકની મદદથી ડેવલોપર્સ એક અથવા એક કરતા વધારે પેજની ઈન્ફોર્મેશન મલ્ટીપલ રીક્વેસ્ટ માટે સ્ટોર કરાવી શકે છે.



### VIEW STATE\*

- View State એ ક્લાઈન્ટ સાઈડ સ્ટેટ મેનેજમેન્ટ ટેકનિક છે.
- Asp.Net માં, View State પ્રોપર્ટીની મદદથી વેબ પેજની પેજ લેવલ પરનાં કંટ્રોલના સ્ટેટની ઇન્ફોર્મેશન સ્ટોર કરાવી શકાય છે. જ્યાં સુધી યુઝર પેજ સાથે વર્ક કરે છે ત્યાં સુધી સીન્ગલ યુઝર માટે View State પ્રોપર્ટીની મદદથી સ્ટેટ ઇન્ફોર્મેશન સ્ટોર કરાવી શકાય છે.
- View State પ્રોપર્ટીની મદદથી જ્યારે યુઝર પહેલી વખત વેબ પેજને વિઝીટ કરતા ટાઈમ જે ઇન્ફોર્મેશન સબમિટ કરેલી હોય તે તે જ ઇન્ફોર્મેશન જ્યારે યુઝર બીજી વખત વેબ પેજને વિઝીટ કરે તે ટાઈમ મળી રહે છે.
- દરેક પેજ માટે View State પ્રોપર્ટી ઈનીસીયલાઈઝ અને સ્ટોર કરાવી શકીએ છીએ.
- View State પ્રોપર્ટીના વેલ્યુ એટ્રીબ્યુટને જે સ્ટ્રીંગ અસાઈન કરવામાં આવે છે અને તે જ વેબ પેજના ભાગ તરીકે પાછી મોકલવામાં આવે છે.

#### ViewState.aspx

```
<table><tr><td><asp:Label ID="lblUserName" runat="server" Text="User
Name"></asp:Label></td>
<td><asp:TextBox ID="txtUserName" runat="server"></asp:TextBox></td></tr>
<tr><td><asp:Label ID="lblPassword" runat="server"
Text="Password"></asp:Label></td>
<td><asp:TextBox ID="txtPassword"
runat="server"></asp:TextBox></td></tr>
<tr><td><asp:Button ID="btnSubmit" runat="server" Text="Submit" /></td>
<td><asp:Button ID="btnRestore" runat="server" Text="Restore" /></td></tr>
</table>
```

#### ViewState.aspx.vb

```
Public username As String
Public password As String
```

```
Protected Sub btnSubmit_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnSubmit.Click
```

```
ViewState("username") = txtUserName.Text
ViewState("password") = txtPassword.Text
txtUserName.Text = ""
txtPassword.Text = ""
```

```
End Sub
```

```
Protected Sub btnRestore_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnRestore.Click
```

```
txtUserName.Text = ViewState("username").ToString
txtPassword.Text = ViewState("password").ToString
```

```
End Sub
```



### VIEW STATE ના ગેર ફાયદા:

1. **Security Risk:** પેજ આઉટપુટના સોર્સમાં View State ની ઇન્ફોર્મેશન જોઈ શકાય છે. હિડન ફિલ્ડના કન્ટેન્ટને થોડા પ્રોગ્રામીંગની મદદથી એનક્રિપ્ટ અને ડીક્રીપ્ટ કરી શકાય છે.
2. **Performance:** વધારે ડેટા હોય ત્યારે તેનું પરફોર્મન્સ સારું હોતું નથી કેમ કે View State એ પેજમાં જ બધી ઇન્ફોર્મેશન સ્ટોર કરે છે તેથી વધારે વેલ્યુ સ્ટોર કરવાથી પેજ સ્લો થઈ જાય છે.
3. **Device limitation:** મોબાઈલ પાસે તેટલી મેમરીની કેપેસિટી ના પણ હોય કે તે View State નો વધારે ડેટા હેન્ડલ ના પણ કરી શકે.
4. તે એક જ પેજ માટેની વેલ્યુ સ્ટોર કરે છે.

### QUERYSTRING

- Query String એ ક્લાઈન્ટ સાઈડ સ્ટેટ મેનેજમેન્ટ ટેકનિક છે.
- Query String એટલે કોમ્પ્યુટર અથવા વેબ બ્રાઉસર માટે ફેરેક્ટરનો સમૂહ.
- Query String ની મદદથી આપણે વેલ્યુ એક વેબ પેજ પરથી બીજા વેબ પેજ પર મોકલી શકીએ છીએ.
- ASP.NET માં વેબ ફોર્મ્સમાં કન્ટેન્ટ એક પેજમાંથી બીજા પેજ પર ટ્રાન્સફર કરી શકીએ છીએ, Query String ખુબ જ સરળ છે અને Query String માં ફેરેક્ટરને અલગ કરવા માટે જનરલી '?' નો ઉપયોગ થાય છે.
- જો મોટી અમાઉન્ટમાં ડેટા ટ્રાન્સફર કરવો હોય (100 kb કરતા વધારે) તો **Request.QueryString** નો ઉપયોગ કરી શકાતો નથી.
- URL માં એક કરતા વધારે query strings ને પાસ કરી શકાય છે અને તને અલગ કરવા માટે ampersand ('&') નો ઉપયોગ થાય છે.
- જો આપણે "DARSHAN & COMPUTER" કોઈ આ રીતે વેલ્યુ પાસ કરવી હોય તો કોડમાં થોડો ચેન્જ કરવો પડે છે.

### QueryString.aspx

```
<div><b>QueryString Example</b></div><br />
<table><tr><td><b>UserId:</b></td><td><asp:TextBox ID="txtUserId"
runat="server"/></td></tr>
<tr><td><b>UserName:</b></td><td><asp:TextBox ID="txtUserName"
runat="server"/></td></tr>
<tr><td></td><td><asp:Button ID="btnSend" Text="Send Values"
runat="server"/></td></tr></table>
```

### QueryString.aspx.vb

```
Protected Sub btnSend_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnSend.Click
    Dim susername As String = txtUserName.Text.Replace("&", "%26")
    Dim suserid As String = txtUserId.Text.Replace("&", "%26")
    Response.Redirect("Default.aspx?UserId=" + susername + "&UserName="
+ suserid)
End Sub
```

### Default.aspx

```
<div><b>QueryString parameter Values in Default.aspx Page</b></div><br />
<div><b>UserId:</b><asp:Label ID="lblUserId" runat="server"/></div><br />
<div><b>UserName:</b><asp:Label ID="lblUserName" runat="server"/></div>
```

### Default.aspx.vb

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    If Not Page.IsPostBack Then
        lblUserId.Text = Request.QueryString("UserId")
        lblUserName.Text = Request.QueryString("UserName")
    End If
End Sub
```

### CONTROL STATE:

- Control State એ ક્વાઈન્ટ સાઈડ સ્ટેટ મેનેજમેન્ટ ટેકનિક છે.
- જ્યારે ડેવલોપર કોઈ કસ્ટમ કંટ્રોલ બનાવે છે અને તેની પાસે કંઈક ઈન્ફોર્મેશનન સ્ટોર કરવી હોય ત્યારે, ત્યારે આપણે view state નો ઉપયોગ કરી શકીએ છીએ. પરંતુ જો યુઝર દ્વારા જ view state ને ડીસએબલ કરી નાખેલ હોય તો કંટ્રોલ સરખું વર્ક કરી શકતું નથી.
- આપણી જરૂરિયાત પ્રમાણે કંટ્રોલની Control State પ્રોપર્ટીનો ઉપયોગ કરવો પડે છે. Control state એ view state કરતાં અલગ છે.
- જો ડેવલોપર કોઈ કસ્ટમ કંટ્રોલ બનાવે અને તેમાં view state બરોબર વર્ક કરે તેના માટે control state નો ઉપયોગ કરી શકાય છે જેનાથી કોઈ બીજા ડેવલોપર view state ડીસએબલ ના કરી શકે.

### Default.aspx

```
<table><tr><td>
<asp:Label ID="lblname" runat="server" Text="Name"></asp:Label>
</td><td><asp:TextBox ID="txtname"
runat="server"></asp:TextBox></td></tr>
<tr><td><asp:Button ID="btnsubmit" Text="Submit" runat="server" />
</td></tr></table>
```

### Default.aspx.vb

```
Protected Sub btnsubmit_Click(sender As Object, e As EventArgs) Handles
btnsubmit.Click
    Server.Transfer("WebForm.aspx")
End Sub
```

### WebForm.aspx

```
<div><asp:Label ID="lblmsg" runat="server"></asp:Label></div>
```

### WebForm.aspx.vb

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    Dim msg As TextBox = CType(PreviousPage.FindControl("txtname"),
TextBox)
    lblmsg.Text = "Enterd Name is: " & msg.Text
End Sub
```

### HIDDEN FIELDS:

- Hidden Field એ ક્લાઈન્ટ સાઈડ સ્ટેટ મેનેજમેન્ટ ટેકનિક છે.
- Hidden Field ની મદદથી આપણે કોઈ ઇન્ફોર્મેશન ડિસ્પ્લે કર્યા વગર તેને સ્ટોર કરાવી શકીએ છીએ.
- આ કંટ્રોલની મદદથી ડેવલોપર વેલ્યુ ડિસ્પ્લે ના થાય તે રીતે સ્ટોર કરાવી શકીએ છીએ. સ્ટેટ વેલ્યુ સ્ટોર કરવા માટે Hidden Field કંટ્રોલનો ઉપયોગ કરી શકાય છે.
- Hidden Field ની વેલ્યુ ક્લાઈન્ટના બ્રાઉસરમાં જ રેન્ડર થાય છે તેથી સેન્સીટીવ વેલ્યુ કે જેને સિક્યોરીટીની જરૂર હોય તેવી વેલ્યુ સ્ટોર ના કરાવવી જોઈએ.

#### • Default.aspx

```
<table><tr><asp:HiddenField ID="hdn" runat="server" />
<td><asp:TextBox ID="txtname" runat="server"></asp:TextBox></td></tr>
<tr><td><asp:Button ID="btnsubmit" runat="server" Text="Submit" />
</td><td><asp:Label ID="lblname" runat="server" Text=""></asp:Label>
</td></tr></table>
```

#### • Default.aspx.vb

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    hdn.Value = txtname.Text
End Sub
Protected Sub btnsubmit_Click(sender As Object, e As EventArgs) Handles
btnsubmit.Click
    lblname.Text = hdn.Value
End Sub
```

### COOKIES

- Cookie એ ક્લાઈન્ટ સાઈડ સ્ટેટ મેનેજમેન્ટ ટેકનિક છે. તેની મદદથી પેજને પોસ્ટ-બેક કર્યા બાદ કંટ્રોલની વેલ્યુ સ્ટોર કરાવી શકાય છે.
- Cookie મતલબ વેબ સર્વર દ્વારા મોકલવામાં આવેલ નાની ઇન્ફોર્મેશન કે જે ક્લાઈન્ટના મશીન પર સ્ટોર થાય છે.
- તે ક્લાઈન્ટના મશીન પર યુઝર નેમ, પાસવર્ડ, સીટી અથવા ફોન નં. વગેરે વેલ્યુ સ્ટોર કરાવી શકાય છે.
- **Cookie ના પ્રકારો:**
  1. Persistent Cookie: તેવી cookie કે જે ગમે ત્યારે કોઈ પણ સમયે ઉપયોગ કરી શકીએ છીએ.
  2. Non-Persistent Cookie: તેવી cookie કે જે અમુક સમય પછી ઉપયોગ કરી શકતા નથી.



- Asp.net માં Cookie ક્રીએટ સરળતાથી કરી શકાય છે, તેની મુખ્ય ૨ રીતો છે: Response object અને HttpCookie

### 1. Response object Type:

```
Response.Cookies("UserName").Value = txtUserName.Text
```

### 2. HttpCookie Type:

```
Dim nameCookie As New HttpCookie("UserName")
nameCookie.Values("UserName") = txtUserName.Text
```

### CookiesDemo.aspx

```
<asp:Label ID="lblCookie" runat="server"></asp:Label>
<asp:TextBox ID="txtUserName" runat="server"></asp:TextBox>
<asp:Button ID="btnCreateCookie" runat="server" Text="Create Cookie" /><br />
<asp:Button ID="btnRetrieve" runat="server" Text="Retrieve Cookie" />
```

### CookiesDemo.aspx.vb

```
Protected Sub btnCreateCookie_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnCreateCookie.Click
    Response.Cookies("UserName").Value = txtUserName.Text
    Response.Cookies("UserName").Expires = DateTime.Now.AddSeconds(30)
    lblCookie.Text = "Cookies Created"
    txtUserName.Text = ""
End Sub

Protected Sub btnRetrieve_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnRetrieve.Click
    If (Request.Cookies("UserName") Is Nothing) Then
        lblCookie.Text = "There is no cookie"
    Else
        lblCookie.Text = "Your cookie contains:" &
Request.Cookies("UserName").Value
    End If
End Sub
```

### • IMPORTANT PROPERTIES OF COOKIES:

Name	Description
Domain	Cookie નું domain સેટ કરવા માટે આ પ્રોપર્ટીનો ઉપયોગ કરી શકીએ છીએ.
Expires	Cookie નો Expiration time સેટ કરવા માટે આ પ્રોપર્ટીનો ઉપયોગ કરી શકીએ છીએ.
Value	Cookie ની વેલ્યુ સ્ટોર કરે છે.

### HOW CAN WE CONTROL COOKIES SCOPE?

- કોઈ વેબસાઈટ માટેની બધી cookie ક્લાઈન્ટ પર એક સાથે સ્ટોર થાય છે અને તે પછી વેબસાઈટ માટેની બધી cookies કોઈ રીક્વેસ્ટની સાથે એક સાથે મોકલવામાં આવે છે. Cookie પર સ્કોપ આપવાની ૨ રીતો છે જે નીચે પ્રમાણે છે:

#### 1. સર્વર પરના કોઈ ફોલ્ડર માટે

- પ્રેક્ટીકલી જોવા જઈએ તો વેબસાઈટ પરની કોઈ એપ્લિકેશન માટે cookie ની લીમીટ સેટ કરી શકાય છે.

```
Dim appCookie As New HttpCookie("AppCookie")
appCookie.Value = "written " & Now.ToString
appCookie.Expires = Now.AddDays(1)
appCookie.Path = "/Application1"
Response.Cookies.Add(appCookie)
```

- આની મદદથી આપણે એ કહી શકીએ કે cookie, Application1 ફોલ્ડરમાં આવેલ વેબ પેજ માટે જ વર્ક કરે છે.
- દા:ત: કોઈ એક વેબસાઈટ [www.dashan.ac.in](http://www.dashan.ac.in) છે, આપણે આગળ દર્શાવેલ ઉદાહરણમાં બનાવેલ cookie <http://www.dashan.ac.in/Application1/> આ પથ પર જ અવેલેબલ હશે અને આ ફોલ્ડરમાં આવેલ વેબપેજ <http://www.dashan.ac.in/Application2/> or just <http://www.dashan.ac.in/> આ પાથ માટે અવેલેબલ હશે નહિ.

#### 2. ડોમેઈન પર સ્કોપ

- Cookies ડોમેઈન સાથે સંકળાયેલ હોય છે.
- દા:ત: કોઈ એક વેબસાઈટ [www.dashan.ac.in](http://www.dashan.ac.in) છે, જ્યારે યુઝર આ વેબસાઈટના કોઈ વેબ પેજ માટે રીક્વેસ્ટ કરે છે ત્યારે યુઝર દ્વારા બનાવેલ Cookies સર્વર પર મોકલવામાં આવે છે.
- જો તમારી વેબસાઈટમાં સબડોમેઈન હોય જેમ કે - [degree.darshan.ac.in](http://degree.darshan.ac.in) અને [diploma.darshan.ac.in](http://diploma.darshan.ac.in) તો તમે cookies ને કોઈ સ્પેસિફિક ડોમેઈન સાથે સંકળાયેલ રાખી શકો છો.

```
Response.Cookies("domain").Domain = "diploma.darshan.ac.in"
```

### SESSION STATE\*

- Session state એ સર્વર સાઈડ સ્ટેટ મેનેજમેન્ટ ટેકનીક છે.
- Session state નો ઉપયોગ દરેક યુઝરનો એપ્લિકેશન દરમિયાન session મેઈન્ટેઈન કરવા માટે થાય છે. Session ની મદદથી તમે ઇન્ફોર્મેશન એક પેજમાં સ્ટોર કરી અને બીજા પેજમાં ઉપયોગ કરી શકો છો અને તે દરેક પ્રકારના ઓબ્જેક્ટને સપોર્ટ કરે છે.
- આપણે ઘણી વેબસાઈટમાં જોઈએ છીએ કે યુઝર એન્ટર કરેલ યુઝર નેમ બધા જ વેબ પેજ પર હોય છે, તે session માં સ્ટોર કરવામાં આવે છે અને વેબસાઈટના દરેક પેજ પર જોઈ શકીએ છીએ.
- Important Properties of Session:
  - ✓ **Timeout** – કોઈ એપ્લિકેશન માટે Session ઓબ્જેક્ટનો timeout period (મિનીટમાં) સેટ અથવા રીટર્ન કરે છે.
- Important Method of Session:
  - ✓ **Abandon** – કોઈ યુઝરનું Session ડીસ્ટ્રોય કરી આપે છે.

- Modes of Session:**

Sr. No.	Mode Name	Description
1	InProcMode	સેસન સ્ટેટનો આ ડીફોલ્ટ મોડ છે અહીં સેસન ડેટા વેબ સર્વરની મેમરીમાં સ્ટોર કરવામાં આવે છે.
2	OutProcMode (State Server Mode)	અહીં સેસન ડેટા અલગથી ASP.NET Service નામની મેમરીમાં સ્ટોર કરવામાં આવે છે.
3	SQL Server Mode	અહીં સેસન ડેટા SQL સર્વર ડેટા બેઝમાં સ્ટોર કરવામાં આવે છે.
4	Customer Mode	અહીં સેસન ડેટા કોઈ કસ્ટમ સ્ટોરેજમાં સ્ટોર કરી શકાય છે.
5	Off Mode	અહીં સેસન સ્ટેટ ડીસએબલ થઈ જાય છે.

### Login.aspx

```
<table><tr><td>User Name</td><td><asp:TextBoxID="txtUserName"
runat="server"></asp:TextBox></td>
</tr><tr><td>Password</td><td><asp:TextBox ID="txtPassword"
runat="server"></asp:TextBox></td>
</tr><tr><td><asp:Label ID="lblMessage" runat="server"
Text="Label"></asp:Label></td>
<td><asp:Button ID="btnLogin" runat="server" Text="Button" /></td></tr>
</table>
```

### Login.aspx.vb

```
Protected Sub btnLogin_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnLogin.Click
Dim objConn As New SqlConnection
    objConn.ConnectionString = "Data Source=ANS-PC\SQLEXPRESS;Initial
Catalog=Diploma;Integrated Security=True"
    objConn.Open()
    Dim objcmd As New SqlCommand
    objcmd.Connection = objConn
    objcmd.CommandType = CommandType.Text
    objcmd.CommandText = "select * from UserDetail where User_Name='"
+ txtUserName.Text + "' and Password='" + txtPassword.Text + "'"
    Dim objDR As SqlDataReader = objcmd.ExecuteReader()
    If objDR.HasRows = True Then
        Session("UserName") = txtUserName.Text
        Response.Redirect("Home.aspx")
    Else
        lblMessage.Text = "Please Enter Correct User Name and
Password"
    End If
End Sub
```



### Home.aspx

```
<table><tr><td><asp:Label ID="lblShowUserName" runat="server"
Text="Label"></asp:Label></td>
<td><asp:LinkButton ID="lbLogout"
runat="server">Logout</asp:LinkButton></td></tr>
</table>
```

### Home.aspx.vb

```
Protected Sub lbLogout_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles lbLogout.Click
    Session.Abandon()
    Response.Redirect("Login.aspx")
End Sub

Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    If (Session("UserName") Is Nothing) Then
        Response.Redirect("Login.aspx")
    Else
        lblShowUserName.Text = "Welcome " + Session("UserName").ToString
    End If
End Sub
```

### Application State and Global.asax

- Application State એ સર્વર સાર્થક સ્ટેટ મેનેજમેન્ટ ટેકનીક છે. Application State એ સર્વરની મેમરીમાં સ્ટોર કરવામાં આવે છે અને તે ડેટાબેઝમાંથી વેલ્યુ રીટ્રાઈવ અને સ્ટોર કરવા કરતા વધારે ફાસ્ટ છે.
- Session state એ કોઈ એક સ્પેસિફિક યુઝર માટે છે જ્યારે Application State એ દરેક યુઝર અને સેસન માટે છે. Application State નો expiration period હોતો નથી.
- Important Methods of Application:
  - ✓ **Lock** – લોક મેથડની મદદથી બીજા યુઝર વેરીએબલની વેલ્યુ ચેન્જ કરી શકતા નથી.
  - ✓ **Unlock** – અનલોક મેથડની મદદથી બીજા યુઝર વેરીએબલની વેલ્યુ ચેન્જ કરી શકાય છે.
- Important Methods of Global.asax:

Name	Description
Application_Start()	જ્યારે વેબ સર્વર સ્ટાર્ટ થાય અને તેમાંથી પહેલા રિસોર્સ માટેની રીક્વેસ્ટ કરવામાં આવે ત્યારે આ મેથડ એકઝીક્યુટ થાય છે
Session_Start()	જ્યારે કોઈ એક પેજ માટે દરેક નવો યુઝર રીક્વેસ્ટ કરે છે ત્યારે આ મેથડ એકઝીક્યુટ થાય છે.
Application_Error()	જ્યારે એરર આવે છે ત્યારે આ મેથડ એકઝીક્યુટ થાય છે
Session_End()	જ્યારે યુઝરનું સેસન પૂરું થાય છે ત્યારે આ મેથડ એકઝીક્યુટ થાય છે
Application_End()	જ્યારે વેબ એપ્લિકેશન તેનું એક્ઝીક્યુશન પૂરું કરે છે ત્યારે આ મેથડ એકઝીક્યુટ થાય છે
Application_Disposed()	જ્યારે વેબ એપ્લિકેશન ડીસ્ટ્રોય થાય છે ત્યારે આ મેથડ એકઝીક્યુટ થાય છે

### Global.asax

```
Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
    Application("SiteVisitedCounter") = 0
End Sub
Sub Session_Start(ByVal sender As Object, ByVal e As EventArgs)
    Application.Lock()
    Application("SiteVisitedCounter") =
    Convert.ToInt32(Application("SiteVisitedCounter")) + 1
    Application.Unlock()
End Sub
```

### CountUser.aspx

```
<asp:Label ID="lblSiteVisited" runat="server"></asp:Label>
```

### CountUser.aspx.vb

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    lblSiteVisited.Text = "No of times site visited=" +
    Application("SiteVisitedCounter").ToString()
End Sub
```

### Web.Config file\*

- વેબસાઈટને મેનેજ કરવા માટેના અલગ અલગ સેટિંગ કન્ફીગ્યુરેશન ફાઈલમાં લખવામાં આવે છે. XML ફાઈલમાં જે સેટિંગ સ્ટોર કરવામાં આવે છે તે એપ્લિકેશન કોડથી અલગ સ્ટોર કરવામાં આવે છે. કોઈ વેબસાઈટના એપ્લિકેશન રુટ ડિરેક્ટરી ફોલ્ડરમાં Web.config નામની ફાઈલ સ્ટોર કરવામાં આવે છે. જનરલી કોઈ વેબસાઈટમાં એક જ Web.config ફાઈલ હોય છે.
- કન્ફીગ્યુરેશન ફાઈલમાં આપણે ઘણી પ્રકારના સેટિંગ સ્ટોર કરી શકીએ છીએ. તેમાંના અમુક કન્ફીગ્યુરેશન કે જે વારંવાર ઉપયોગમાં આવતા હોય કે જે Web.config ફાઈલની અંદર સ્ટોર થાય છે. જેવા કે ડેટાબેઝ કનેક્શન, કેશીંગ સેટિંગ, સેસન સ્ટેટ્સ, એરર હેન્ડલિંગ, સીક્યુરીટી વગેરે....

- **Configuration for Connection String:**

```
<configuration>
<connectionStrings>
<add name ="myCon" connectionString ="server=ANS-PC\SQLEXPRESS;Initial
Catalog=Diploma;Integrated Security=True"
</connectionStrings></configuration/>
```

- **Reading connectionstring values**

```
Dim con As string =
ConfigurationManager.ConnectionStrings("myCon").ConnectionString
```



- **Configuration for Custom Error Handling:**

```
<customErrors defaultRedirect ="Error.aspx">
    <error statusCode ="404" redirect ="NotFound.aspx"/>
</customErrors>
```

- **Session State and View State Settings:**

```
<Pages EnableViewState="false" /><sessionState mode="InProc" />
```

### Cross Posting:

- જો આપણે સેસન, કુકી અથવા કોઈ પણ પ્રકારના ઓબ્જેક્ટની મદદ વગર એક પેજ પરથી ડેટા બીજા પેજ પર સેન્ડ કરવાનો હોય તો ત્યારે આપણે ક્રોસ પોસ્ટીંગ ટેકનીક નો ઉપયોગ કરી શકીએ છીએ.
- ક્રોસ પેજ પોસ્ટીંગ કન્સેપ્ટની મદદથી એક પેજ પરના કંટ્રોલ ને બીજા પેજ પર એક્સેસ કરી શકાય છે.

### ADVANTAGES & DISADVANTAGES OF CLIENT SIDE AND SERVER SIDE:

- **CLIENT SIDE:**

- ✓ **ADVANTAGES:**

1. સર્વર સાઈડના રીસોર્સની જરૂર પડતી નથી
2. સિમ્પલ ઇમ્પલીમેન્ટેશન
3. એન્ડેસ સિક્યુરિટી ફ્રીયર
4. વાઈડ સ્પ્રેડ સપોર્ટ
5. Cookie માં expiration rules કન્ટ્રીગ્યુર કરી શકાય

- ✓ **DISADVANTAGES:**

1. મોટી વેલ્યુ સ્ટોર કરી શકતા નથી તેને કારણે યુઝરને પેજ ડિસ્પ્લે થવાની પ્રોસેસમાં ટાઈમ વધારે લાગે છે.
2. મોબઈલ ViewState સપોર્ટ ના પણ કરે
3. સેન્સીટીવ ડેટા માટે સિક્યુરિટી પ્રોવાઈડ કરતું નથી
4. જો યુઝર દ્વારા બ્રાઉસરમાં કુકી ડીસએબલ કરેલી હશે તો તે ટેકનીકનો ઉપયોગ કરી શકાય નહિ
5. Query string ના ઉપયોગથી જે કઈ પણ ડેટાની આપ લે થાય છે તે યુઝર ના બ્રાઉસરમાં ડાયરેક્ટ જોઈ શકાય છે, જેને કારણે તેને સેન્સીટીવ ડેટા માટે ઉપયોગ કરી શકાય નહિ
6. Cookies ક્લાઈન્ટના મશીનમાં સ્ટોર થાય છે, જો હેકર ક્લાઈન્ટનું મશીન હેક કરે તો તે બધી Cookies જોઈ શકે છે.

- **SERVER SIDE:**

- ✓ **ADVANTAGES:**

1. Application Object ની મદદથી કોઈ ઇન્ફોર્મેશનને એપ્લીકેશન લેવલે એક્સેસ કરી શકાય, ગ્લોબલ લેવલે સ્ટોર કરી શકાય અથવા કોઈ વેબસાઈટ અથવા એપ્લીકેશનને એક જ ટાઈમ એક કરતા વધારે યુઝર્સ એક્સેસ કરી શકે છે.
2. સેસન મેનેજમેન્ટ ઇવેન્ટનો એપ્લીકેશનમાં ઉપયોગ કરી શકાય છે.
3. સેસન સ્ટેટ multi-computer અને multi-process બંને પ્રકારના કન્ફીગ્યુરેશનમાં ઉપયોગ કરી શકાય છે.
4. જે બ્રાઉસર HTTP cookies સપોર્ટ કરતા નથી તેમાં સેસન સ્ટેટનો ઉપયોગ કરી શકાય છે.

- ✓ **DISADVANTAGES:**

1. Session-state અને Application state માં સર્વરની મેમરીનો ઉપયોગ થાય છે તેને કારણે સર્વરનું પર્ફોમન્સ ઘટી જાય છે
2. કોઈ એપ્લીકેશનના Application state માં સ્ટોર થયેલ વેરીઅબલ કોઈ એક પ્રોસેસ માટે જ ગ્લોબલ લેવલ એક્સેસ કરી શકો છો અને તેમ જ દરેક એપ્લીકેશન પ્રોસેસની અલગ વેલ્યુ હોય છે. તેના કારણે, કોઈ યુનિક વેલ્યુ અથવા અપડેટ ગ્લોબલ કાઉન્ટર સ્ટોર કરવા માટે Application state નો ઉપયોગ કરવો જોઈએ નહીં.
3. Data maintenance
4. સ્ટેટ મેનેજમેન્ટ ટેકનીકની કોમ્પ્લેક્સિટી