



Elektrotehnički fakultet Univerziteta u Beogradu
Katedra za računarsku tehniku i informatiku

ASP .NET Active Server Pages

M.Sc. Nemanja Kojić

ASP .NET

- ▶ ASP .NET je veb tehnologija koja omogućava izvršavanje skripti (ugrađenih u veb stranice) na IIS serveru
- ▶ ASP (*Active Server Pages*) – aktivne serverske stranice
- ▶ ASP .NET je program koji se izvršava na IIS serveru
- ▶ IIS (*Internet Information Services*)
 - ▶ Microsoft-ov veb server
 - ▶ dolazi kao besplatna komponenta u paketu sa Windows serverima
 - ▶ IIS je takođe deo Windows 2000 i Windows XP Professional
- ▶ potpuno nova generacija tehnologije za serverski skript (*server-side scripting*)
 - ▶ nije nastao nadogradnjom ASP-a

Microsoft .NET Framework

- ▶ .NET radno okruženje je okruženje za pravljenje (*building*), instaliranje (*deploy*) i za izvršavanje (*running*) veb aplikacija i veb servisa
- ▶ Klasična ASP tehnologija je bila moćna i fleksibilna, ali suviše orijentisana ka programskom kodu (slaba podrška pomoćnih alata – *utilities*)
- ▶ Microsoft .NET Framework je napravljen sa ciljem da se reši ovaj problem (povećanje produktivnosti)
- ▶ **.NET Framework – karakteristike:**
 - ▶ Veća produktivnost programiranja
 - ▶ Smanjena količina programskog koda u izvornim fajlovima
 - ▶ Deklarativni model programiranja ()
 - ▶ Bogatija hijerarhije kontrola zajedno sa obradom događaja
 - ▶ Veća biblioteka klasa
 - ▶ Bolja podrška za rad kroz razne pomoćne alate

Microsoft .NET Framework

- ▶ .NET Framework se sastoji od 3 celine:
 - ▶ Programski jezici
 - ▶ Visual Basic
 - ▶ C#
 - ▶ J#
 - ▶ Serverske i klijentske tehnologije
 - ▶ ASP .NET (Active Server Pages)
 - ▶ Windows Forms (Windows Desktop Solutions)
 - ▶ Compact Framework (PDA/Mobile solutions)
 - ▶ Okruženja za razvoj
 - ▶ Visual Studio .NET
 - ▶ Visual Web Developer

ASP .NET - detalji

Šta je ASP.NET fajl?

- ▶ ASP .NET fajl je isto što i HTML file
- ▶ ASP .NET fajl može da sadrži HTML, XML i skripte
- ▶ Skripte ASP .NET fajlu se izvršavaju na serveru
- ▶ ASP .NET fajl se prepoznaje po ekstenziji ".aspx"

Primer HTML stranice

Primer 1: veb_strana.html

```
<html>
  <body bgcolor="yellow">
    <center>
      <h2>Hello W3Schools!</h2>
    </center>
  </body>
</html>
```

Hello W3Schools!

Primer ASP .NET stranice

Primer 2: veb_strana.aspx

```
<html>  
  <body bgcolor="yellow">  
    <center>  
      <h2>Hello W3Schools!</h2>  
    </center>  
  </body>  
</html>
```

Hello W3Schools!

Kako ASP .NET radi?

- ▶ ASP .NET stranica je ista kao i bilo koja druga HTML stranica
- ▶ Ako pretraživač (*browser*) zahteva HTML (.html) stranicu od servera,
server vrati traženi stranicu
(bez ikakvih modifikacija ili procesiranja te stranice)
- ▶ ASP .NET stranica ima ekstenziju .aspx.
- ▶ Ako pretraživač zahteva određenu ASP .NET stranicu,
 - ▶ IIS prosleđuje zahtev ASP .NET modulu (*engine*) na serveru
 - ▶ ASP .NET engine procesira bilo koji izvršivi kod u okviru date stranice (čita liniju po liniju i izvršava skriptove)
 - ▶ Klijentu se vraća kao rezultat čist HTML

Dinamička ASP .NET veb stranica

Primer 3: dinamičkaVebStrana.aspx

```
<html>
  <body bgcolor="yellow">
    <center>
      <h2>Hello EPOS!</h2>
      <p><%Response.Write(now())%></p>
    </center>
  </body>
</html>
```

Hello W3Schools!

2/25/2010 2:05:17 PM

Hello W3Schools!

2/25/2010 2:10:37 PM

ASP .NET serverske kontrole (*server controls*)

- ▶ ASP.NET je rešio problem "spaghetti-code" koji je bio prisutan u radu sa klasičnim ASP-om
- ▶ Serverske kontrole su tagovi koje server razume (i ume da interpretira)
- ▶ Postoje 3 vrste serverskih kontrola:
 - ▶ HTML serverske kontrole (*HTML Server Controls*) (tradicionalni HTML tagovi)
 - ▶ Web serverske kontrole (*Web Server Controls*) (novi ASP .NET tagovi)
 - ▶ Serverske kontrole za validaciju podataka (*Validation Server Controls*)

HTML serverske kontrole

- ▶ HTML tagovi koje server razume (ume da interpretira)
- ▶ HTML elementi se u ASP .NET fajlovima podrazumevano tretiraju kao tekst.
- ▶ Da bi se ovi elementi učinili programabilnim, dodaje im se atribut **runat="server"**.
- ▶ Ovaj element znači da će element biti tretiran kao serverska kontrola.
- ▶ HTML elementima se dodaje i atribut **id** na osnovu kojeg se identifikuje data kontrola.
- ▶ Vrednosti atributa ID omogućava se referenciranje date kontrole i manipulisanje njom u vreme izvršavanja.
- ▶ **Napomena:** Sve HTML serverske kontrole moraju biti unutar **<form>** taga sa atributom **runat="server"** attribute.
Atribut **runat="server"** označava da se forma procesira na serveru. To takođe znači da se kontrolama unutar date forme može pristupiti iz serverskih skripti.

```
<html>
<body>
  <form runat="server">
    <a id="link1" runat="server">Visit EPOS!</a>
  </form>
</body>
</html>
```

Web serverske kontrole

- ▶ Veb serverske kontrole su specijalni ASP .NET tagovi koje server zna da interpretira
- ▶ Veb serverske kontrole se, kao i HTML kontrole, prave na serveru i zato zahtevaju atribut **runat="server"** .
- ▶ Veb kontrole se u principu ne mapiraju uvek u neke već postojeće HTML kontrole (mogu predstavljati i znatno kompleksnije elemente)
- ▶ Sintaksa za pravljenje veb kontrole:
<asp:control_name id="some_id" runat="server" />

```
<script runat="server">
Sub submit(Source As Object, e As EventArgs)
    button1.Text="You clicked me!"
End Sub
</script>

<html>
  <body>
    <form runat="server">
      <asp:Button id="button1" Text="Click me!"
        runat="server" OnClick="submit"/>
    </form>
  </body>
</html>
```

Serverske kontrole za validaciju

- ▶ Koriste se za validaciju podataka koje korisnik unosi preko forme.
- ▶ Ako podaci ne prođu validaciju ispisuje se poruka o grešci.
- ▶ Postoje različite kontrole za različite tipove validacije.
- ▶ Validacija stranice (podataka unetih na formi) se podrazumevano dešava kada se klikne na Dugme (Button), Sliku (ImageButton) ili link (LinkButton)
- ▶ Validacija se može isključiti postavljanjem vrednosti atributa ***CausesValidation*** na **false**.
- ▶ Sintaksa za kreiranje kontrole za validaciju:
`<asp:control_name id="some_id" runat="server" />`

Serverske kontrole za validaciju-primer

```
<html>
<body>
<form runat="server">
  <p>Enter a number from 1 to 100:
    <asp:TextBox id="tbx1" runat="server" /> <br />
    <asp:Button Text="Submit" runat="server" /></p>

    <p><asp:RangeValidator ControlToValidate="tbx1"
      MinimumValue="1" MaximumValue="100" Type="Integer"
      Text="The value must be from 1 to 100!"
      runat="server" />

</p>
</form>
</body>
</html>
```

ASP .NET veb forme

- ▶ sve serverske kontrole se nalaze u okviru taga <form>
- ▶ svaki tag <form> mora da ima atribut runat="server"
- ▶ **runat="server"** znači da se forma obrađuje na serveru
- ▶ .aspx stranica može da sadrži samo jedan element <form>!
- ▶ neophodni atributi taga <form>
(mogu se automatski dodeliti, ukoliko ih ne specificira programer)
 - ▶ name
 - ▶ method
 - ▶ action
 - ▶ id

```
<form  name="_ctl0"  method="post"  
      action="page.aspx"  id="_ctl0">  
...some code  
</form>
```


Slanje forme na server

(*Submit*)

- ▶ forma (i njen sadržaj) se obično šalju klikom na dugme
- ▶ format serverske kontrole *Button* je sledeći:
`<asp:Button id="id" text="label" OnClick="sub" runat="server" />`

ViewState

- ▶ omogućava čuvanje unetih podataka sa forme
- ▶ ako se na serveru desi greška, podaci koji su već uneti neće biti izgubljeni/izbrisani
- ▶ stanje elemenata forme se prenosu kroz skrivena polja forme
- ▶ održavanja stanja prikaza je podrazumevano ponašanje asp formi
- ▶ može se isključiti stavljanjem sledeće direktive na početak .aspx fajla

```
<%@ Page EnableViewState="false" %>
```

Obrada događaja (Event Handlers)

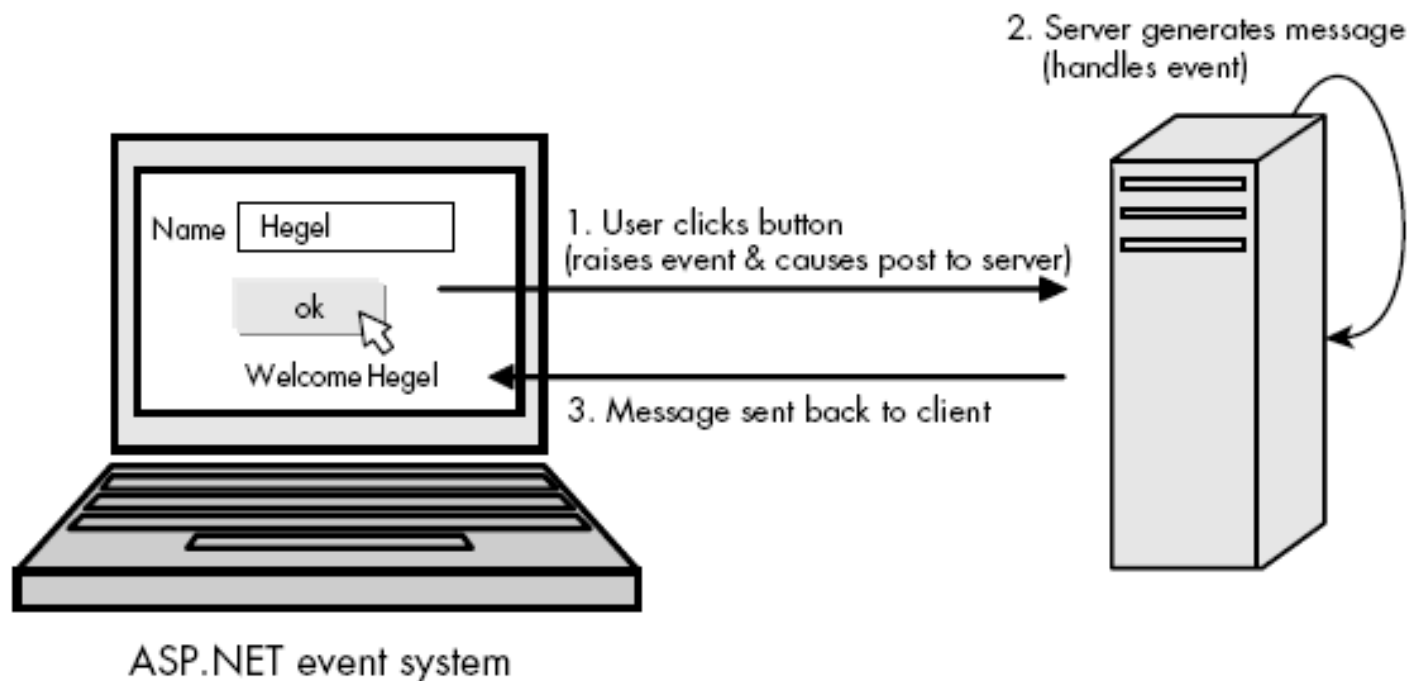


Figure 2.1 Client-based event system versus ASP.NET event system

Tipovi interakcije klijenta i servera

- ▶ Postback
- ▶ Cross-page postback
- ▶ Redirection
- ▶ Server transfer

Postback

- ▶ koncept interakcije klijenta i servera
- ▶ klijent šalje informacije sa trenutne stranice na server i traži da se ista stranica vrati natrag sa servera
- ▶ odgovarajući događaj se dešava se unutar web formi (`<form>`)
- ▶ samo serverske kontrole mogu da izvrše postback slanje informacija na server

Postback interakcija

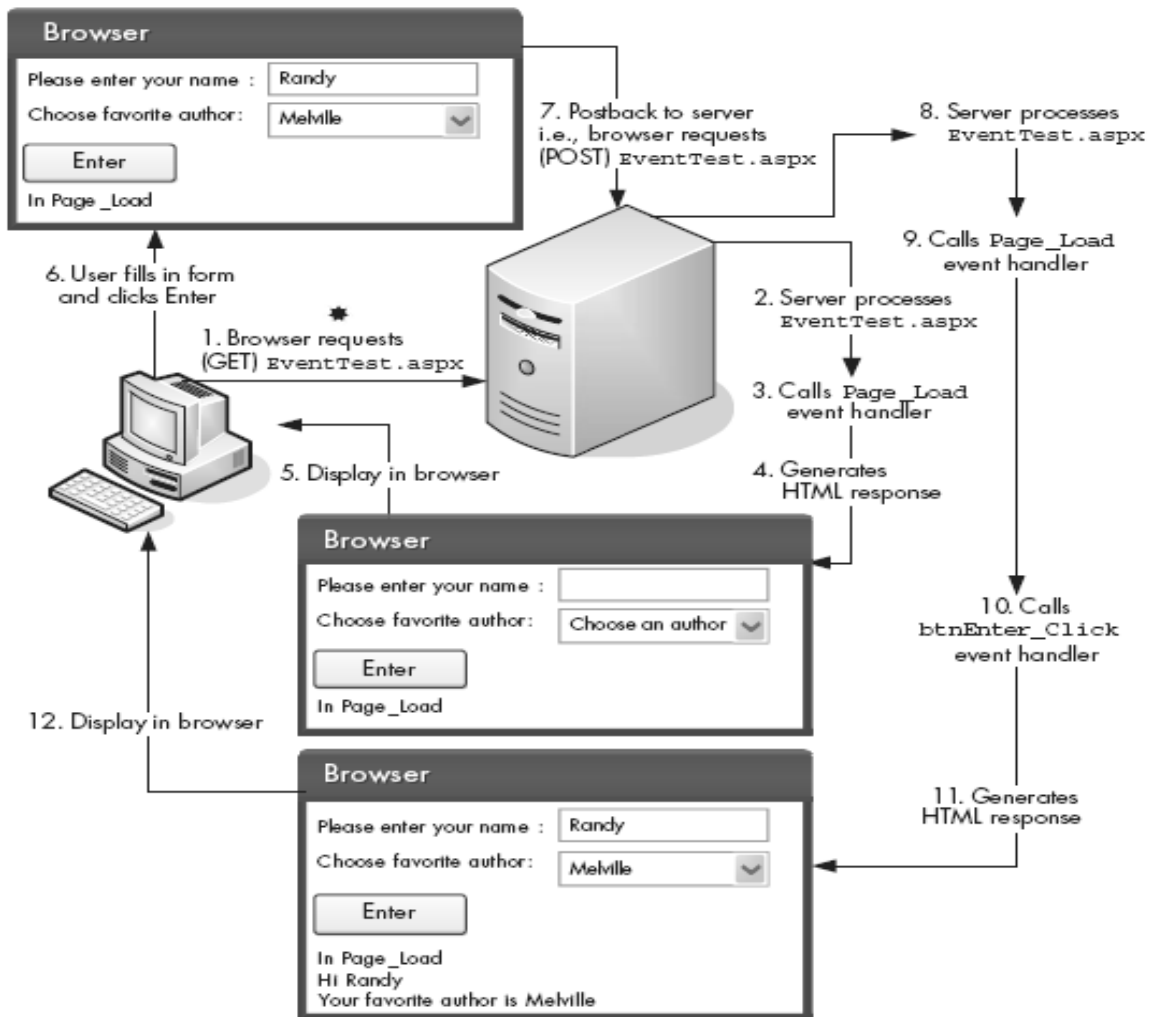


Figure 2.2 Postback flow

Životni ciklus stranice – uprošćeni model (*Page lifecycle*)

- ▶ **Inicijalizacija**
 - ▶ inicijalizacija same stranice i kontrola na njoj
 - ▶ pozivaju se `Page_PreInit` i `Page_Init` za stranicu i kontrole
- ▶ **Učitavanje (loading)**
 - ▶ ako je postback u pitanju, kontrole se popunjavaju vrednostima iz `ViewState-a`
 - ▶ pozivaju se handleri `Page_Load` za stranicu i kontrole na njoj
- ▶ **Obrada postback događaja**
 - ▶ ako je postback, poziva se handler za kontrolu koja je poslala događaj
- ▶ **Renderovanje**
 - ▶ serijalizacija stranice (HTML) kroz response stream
 - ▶ poziva se `PreRender` i `Render` metoda za stranicu i kontrole
- ▶ **Uklanjanje stranice (Unloading)**
 - ▶ oslobađanje resursa
 - ▶ poziva se metoda `Unload` za stranicu i njene kontrole

Zivotni ciklus stranice (2)

	Page methods	Page events	Control methods and events
Start	<div>Construct</div> <div>ProcessRequest</div> <div>InitializeCulture</div> <div>DeterminePostBackMode</div> <div>OnPreInit</div> <div>OnInit</div> <div>TrackViewState</div> <div>OnInitComplete</div>	<div>PreInit</div> <div>Init</div> <div>InitComplete</div>	<div>Init</div> <div>Load control state and view state</div> <div>Load</div> <div>Control-changed events</div>
Load	<div>LoadPageStateFromPersistenceMedium</div> <div>OnPreLoad</div> <div>OnLoad</div>	<div>PreLoad</div> <div>Load</div>	<div>Load</div>
Event handling	<div>RaisePostBackEvent</div>		<div>Control-changed events</div>
Validation	<div>Validate</div>		
Rendering	<div>OnLoadComplete</div> <div>OnPreRender</div> <div>OnPreRenderComplete</div> <div>SaveViewState</div> <div>SavePageStateToPersistenceMedium</div> <div>OnSaveStateComplete</div> <div>RenderControl</div> <div>Render</div> <div>RenderChildren</div>	<div>LoadComplete</div> <div>PreRender</div> <div>PreRenderComplete</div> <div>SaveStateComplete</div>	<div>PreRender</div> <div>Data binding events</div> <div>Save control state and view state</div> <div>Render</div> <div>Unload</div>
Unload	<div>OnUnload</div> <div>Dispose</div>		

- Method name
- Event name
- PostBack only

Page_Load Event

- ▶ Page_Load event se javlja kada se stranica učitava,
- ▶ ASP.NET automatski poziva rutinu Page_Load() i izvršava kod unutar nje.
- ▶ **Primer:**

```
<script runat="server">  
Sub Page_Load  
    lbl1.Text="The date and time is " & now()  
End Sub  
</script>  
  
<html>  
    <body>  
        <form runat="server">  
            <h3><asp:label id="lbl1" runat="server"/></h3>  
        </form>  
    </body>  
</html>
```

Page.IsPostBack atribut (detektovanje postback-a)

- ▶ Page_Load rutina se poziva svaki put kada se stranica učitava.
- ▶ Ako je (**Page.IsPostBack = false**) => stranica se učitava **prvi put**,
- ▶ Ako je (**Page.IsPostBack = true**) => stranica se **šalje na server postback-om**
- ▶ korisno za izvršavanje koje treba da se desi samo kada se stranica traži prvi put

```
<script runat="server">
Sub Page_Load
    if Not Page.IsPostBack then
        lbl1.Text="The date and time is " & now()
    end if
End Sub
Sub submit(s As Object, e As EventArgs)
    lbl2.Text="Hello World!"
End Sub
</script>

<html><body>
    <form runat="server">
        <asp:button text="Submit" onclick="submit" runat="server"/>
    </form>
</body></html>
```

The date and time is 2/25/2010 3:31:31 PM

Hello World!

Submit

AutoPostBack

- ▶ neke kontrole uvek generišu postback zahteve
 - ▶ button
- ▶ neke kontrole nikad ne generišu postback zahteve
 - ▶ label
- ▶ neke kontrole podrazumevano ne generišu postback zahteve, ali mogu da se iskonfigurišu drugačije
 - ▶ dropdown list
 - ▶ AutoPostBack =true

Cross-page posting

- ▶ podaci sa trenutne stranice se šalju sa klijenta POST metodom
- ▶ umesto iste stranice, nazad se zahteva druga .aspx stranica
- ▶ kontrole imaju atribut `PostBackUrl`
 - ▶ `<asp:Button ID="btnEnter" Text="Enter" runat="server" PostBackUrl="OtherPage.aspx" />`
- ▶ omogućava se tok podataka između stranica
 - ▶ atribut `PreviousPage` stranice na koju se odlazi omogućava pristup podacima sa stranice sa koje je poslat zahtev
- ▶ Primer:

```
TextBox txtName =  
(TextBox) PreviousPage.FindControl("name");
```

Osnovni koncepti na serverskoj strani

▶ **Page**

- ▶ klasa čiji objekti predstavljaju stranice koje se obrađuju na serveru
- ▶ klasa Page poseduje odgovarajuće attribute koji nose sve relevantne informacije o stranici

▶ **Request**

- ▶ predstavlja trenutni zahtev koji se obrađuje

▶ **Response**

- ▶ predstavlja objekat pomoću kojeg se šalje rezultat na klijenta

▶ **Server**

- ▶ data klasa sadrži razne pomoćne metode

▶ **Session**

- ▶ objekat koji omogućava čuvanje stanja interakcije jednog klijenta sa serverom

Response.redirect vs. *Server.transfer*

- ▶ **`Response.redirect(String pageUrl)`**
 - ▶ pozivom date metode na serveru, klijentu se šalje URL stranice
 - ▶ klijent na osnovu dobijenog URL-a generiše novi HTTP zahtev da bi dobio sadržaj date stranice
 - ▶ mana: round trip
- ▶ **`Server.transfer(String pageUrl)`**
 - ▶ umesto URL-a, učitava se stranica i šalje klijentu
 - ▶ efikasnije od redirekcije
 - ▶ međutim, sadržaj adresnog (URL) polja pretraživača se ne menja!
 - ▶ korisno za pipeline procese, gde ne moraju da se otkrivaju u URL polju pretraživača adrese za pojedine korake.

Kontrole koje podržavaju DataBinding

- ▶ `asp:RadioButtonList`
- ▶ `asp:CheckBoxList`
- ▶ `asp:DropDownList`
- ▶ `asp:Listbox`
- ▶ DataBinding označava povezivanje sadržaja datih kontrola sa podacima u izvoru podataka

ASP .NET 2.0 - novine

- ▶ Master Pages, Themes, Web Parts
- ▶ Standardne kontrole za navigaciju
- ▶ Standardne kontrole za sigurnost
- ▶ Uloge, personalizacija, internacionalizacija
- ▶ Poboljšane i pojednostavljene kontrole za pristup podacima
- ▶ Puna podrška za XML standarde (XHTML, XML, i WSDL)
- ▶ Poboljšano kompajliranje i instaliranje aplikacija
- ▶ Poboljšano upravljanje sajtom (*site management*)
- ▶ Novi i bolji alati za razvoj ...

Navigacija – ASP .NET 2.0

- ▶ Ugrađene kontrole za navigaciju:
 - ▶ Mape sajta
(*Site Maps*)
 - ▶ Dinamički HTML meniji
(*Dynamic HTML menus*)
 - ▶ Stabla prikaza
(*Tree Views*)

Sigurnost – ASP .NET 2.0

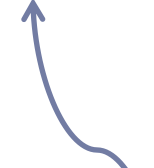
- ▶ Zaštita poverljivih i ličnih informacija.
- ▶ Dodate su sledeće kontrole za sigurnost:
 - ▶ Kontrola Login
 - ▶ Kontrola LoginStatus
(kontrolisanje statusa operacije za prijavljivanje - login)
 - ▶ Kontrola LoginName
(prikazuje trenutno prijavljenog korisnika)
 - ▶ Kontrola LoginView
(omogućava različite prikaze u zavisnosti od statusa prijavljivanja)
 - ▶ “Vizard” za registrovanje (kreiranje) korisničkog naloga CreateUser
 - ▶ Kontrola PasswordRecovery
(omogućava "I forgot my password" funkcionalnost)

Šabloni ASP stranica (*Master Pages*)

- ▶ Definisanje šablona za sve stranice koje dele isti izgled i funkcionalnosti
- ▶ Definiše se zajednički deo, dok se izmenljiva mesta označavaju kao *placeholders*
- ▶ Stranice šablona (master pages)
- ▶ Stranice konkretnih sadržaja (content pages)

Master i content page - primer

```
<%@ Master %>
<html>
<body>
    <h1>Standard Header For All Pages</h1>
    <asp:ContentPlaceHolder id="CPH1" runat="server">
    </asp:ContentPlaceHolder>
</body>
</html>
```



```
<%@ Page MasterPageFile="master1.master" %>
<asp:Content ContentPlaceHolderId="CPH1" runat="server">
    <h2>W3Schools</h2>
    <form runat="server">
        <asp:TextBox id="textbox1" runat="server" />
        <asp:Button id="button1" runat="server" text="Button"/>
    </form>
</asp:Content>
```