



## **ASP web-programiranje**

Četvrta laboratorijska vježba

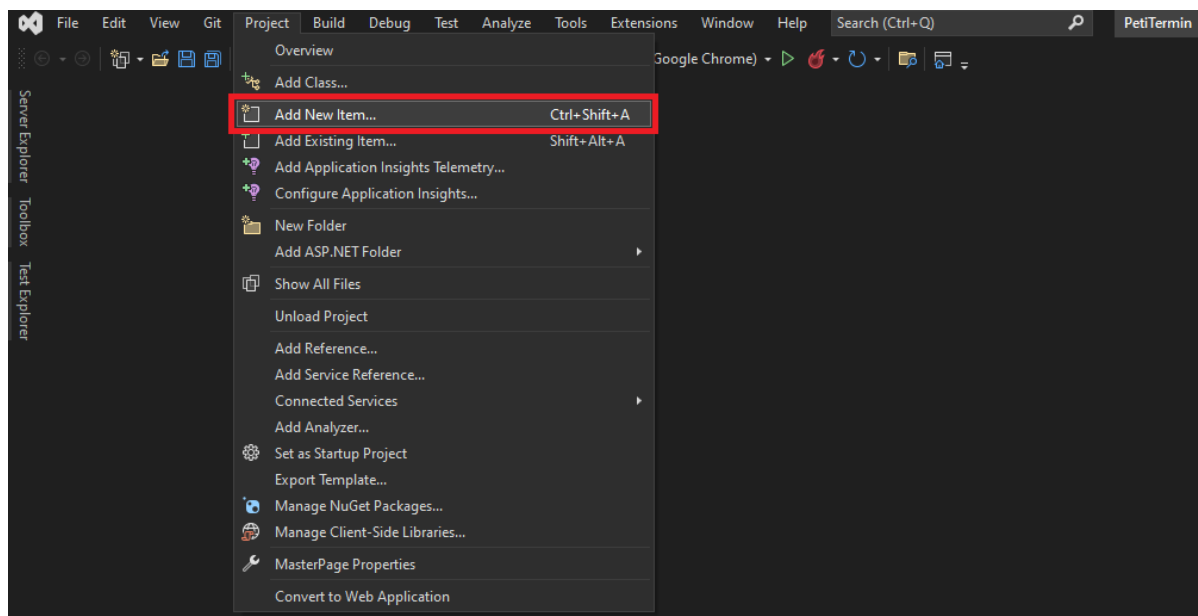
Prilikom izrade višestraničnih ASP.NET aplikacija važno je uskladiti izgled svih stranica, te osigurati međusobnu povezanost i dostupnost svih stranica.

Moguće je uređivati svaku stranicu posebno, ali ukoliko se odluči promijeniti dizajn, odnosno tema ASP.NET aplikacije, moraju se vršiti izmjene dizajna na svakoj stranici posebno. Da bi se to izbjeglo izrađuje se predložak (eng. *template*) koji nasljeđuju sve stranice. U tom slučaju, ukoliko je potrebno napraviti izmjenu u dizajnu, dovoljno je izmijeniti samo predložak.

Zajednički predložak može sadržavati i sadržaj za koji je potrebno da je dostupan na svim stranicama, kao npr. CSS, JavaScript, ASP.NET kontrole, C# izvorni kod, itd.

U ASP.NET-u zajednički predložak se naziva *Master Page* (ekstenzija .master), a stranice koje koriste zajednički predložak *Content Page* (ekstenzija .aspx).

Za dodavanje *Master Page* stranice u projekt koristi se opcija *Project > Add New Item* u programskom alatu Visual Studio (Slika 1).



Slika 1 Visual Studio, dodavanje nove stavke u projekt

U novootvorenom prozoru je potrebno odabrati opciju *Master Page* i odabrati odgovarajuće ime, ali pritom pazeći na ekstenziju.

Prilikom kreiranja *Master Page* stranice automatski se dodaju dvije ASP.NET kontrole (Slika 2).

```
<!DOCTYPE html>

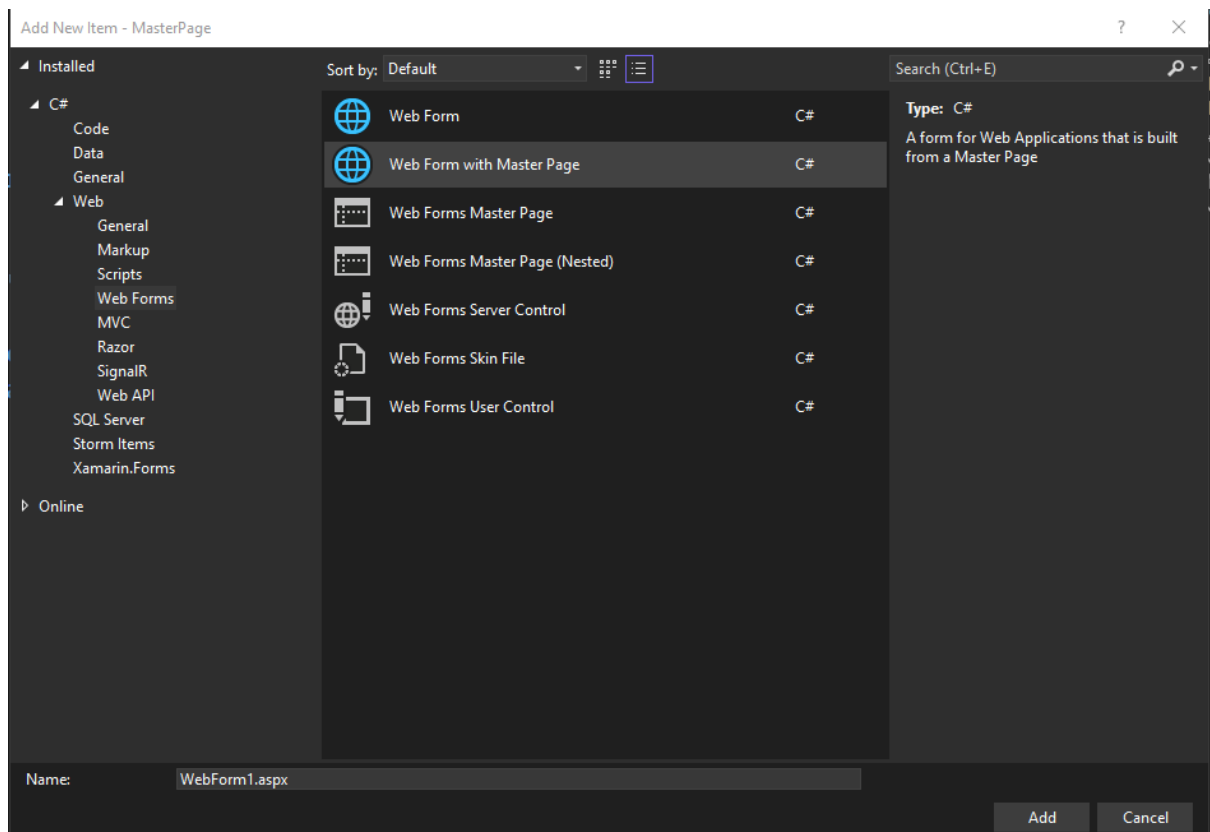
<html>
<head runat="server">
  <title></title>
  <asp:ContentPlaceHolder ID="head" runat="server">
</asp:ContentPlaceHolder>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
</asp:ContentPlaceHolder>
    </div>
  </form>
</body>
</html>
```

Slika 2 Master Page

ASP.NET kontrola *ContentPlaceHolder* označava prostor unutar *Master Page* stranice gdje će se upisivati sadržaj specifičan za pojedinu stranicu. Sve što se postavi izvan kontrola *ContentPlaceHolder* će biti dostupno na svim stranicama koje nasljeđuju, tj. koriste zajednički predložak.

Za dodavanje *Content Page* stranice (stranica koja nasljeđuje, odnosno koristi zajednički predložak) u projekt koristi se opcija *Project > Add New Item* u programskom alatu Visual Studio.

Ovisno o verziji programskog alata *Content Page* se drugačije naziva. U nastavku se nalazi primjera za dodavanje *Content Page* stranice (Slika 3) koji ne sadrže stavku naziva *Content Page*.



Slika 3 Web Form with Master Page

Prazna *Content Page* stranica ima izgled kao na donjoj slici (Slika 4). *Content Page* stranica sadrži nestandardni izgled XHTML stranice. Sadrži samo dvije ASP.NET kontrole *Content*. Kod navedenih dviju kontrola važno je kako se nazivaju, odnosno vrijednost svojstva (atributa) *ContentPlaceHolderID*. Kontrola *Content* imena *head* predstavlja zaglavlje XHTML dokumenta, a druga kontrola *Content* predstavlja tijelo XHTML dokumenta. Ukoliko *Content Page* sadrži više od dvije kontrole *Content*, samo kontrola imena *head* predstavlja zaglavlje, dok su sve preostale kontrole *Content* dio tijela (*body*) XHTML dokumenta.

```
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
</asp:Content>
```

Slika 4 Content Page

Dozvoljeno je dodavanje sadržaja samo unutar ASP.NET kontrola *Content*. Sve što se nalazi izvan kontrola *Content* moguće je promijeniti jedino unutar *Master Page* stranice. Sadržaj izvan kontrola *Content* se nalazi unutar *Master Page* stranice, a ne unutar *Content Page* stranice.

Tema višestranične ASP.NET aplikacije s zajedničkim predloškom je obrađena na predavanju na kojem su dani primjeri i detaljnije objašnjenje.

### **Zadaci:**

- 1) Kreirajte novu ASP.NET aplikaciju u programskom alatu Visual Studio.
- 2) Kreirajte jednu stranicu za prijavljivanje u aplikaciju i registraciju novog korisnika koristeći bazu podataka s prethodne (treće) laboratorijske vježbe. Navedena stranica ne smije koristiti Master Page i Content Page stranice.
- 3) Kreirajte jednu *Master page* stranicu i tri *Content Page* stranice koje nasljeđuju navedenu *Master Page* stranicu.
- 4) Na *Master Page* stranici postavite poveznice prema svim *Content Page* stranicama, te mogućnost odjave iz ASP.NET aplikacije.
- 5) Na prvoj *Content Page* stranici napravite jednostavni kalkulator.
- 6) Na drugoj *Content Page* stranici napravite ispis podataka o Vama.
- 7) Na trećoj *Content Page* stranici napravite ispis podataka iz tablice studenti iz baze podataka.
- 8) Postavite potrebne metode kako bi na *Content Page* stranice mogli pristupiti samo registrirani i prijavljeni korisnici.
- 9) Dodajte u ASP.NET aplikaciju logo VVG-a.
- 10) Dodajte u ASP.NET aplikaciju mogućnost poništavanja odjave koristeći JavaScript. Ispišite pitanje korisniku unutar odgovarajućeg JavaScript prozora. Ukoliko korisnik potvrdi odjavu odjavite ga iz ASP.NET aplikacije (tj. nemojte zaustavljati izvršavanje), a u suprotnom nemojte odjaviti korisnika (prekinite slanje zahtjeva poslužitelju).

*Primjer:*

*onclick="return false;" //zaustavlja slanje zahtjeva poslužitelju*

*onclick="return true;" //nastavlja izvršavanje*

Umjesto vrijednosti true/false treba postaviti poziv prema JavaScript funkciji.

Primjer 2:

*onclick="return confirm('Odjava?');"*

- 11) Dodajte novu stranicu koristeći kreirani zajednički predložak za ASP.NET aplikaciju. Na novododanu stranicu postavite veze prema datotekama koje korisnik može preuzeti (eng. download). Datoteke za vježbu odaberite po vlastitom izboru, ali sve odabrane datoteke postavite u jedan direktorij i postavite referencu prema direktoriju unutar projekta programskog alata Visual Studio.