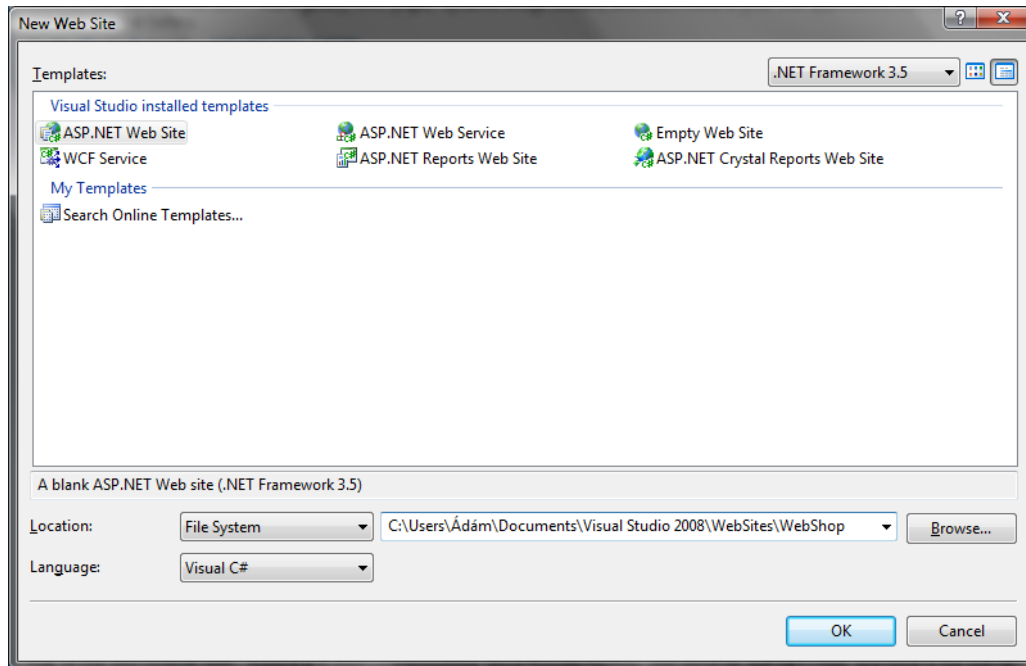


Webshop készítése ASP.NET 3.5-ben – I.

- Portál kialakítása
- Mesteroldal létrehozása
- Témák létrehozása

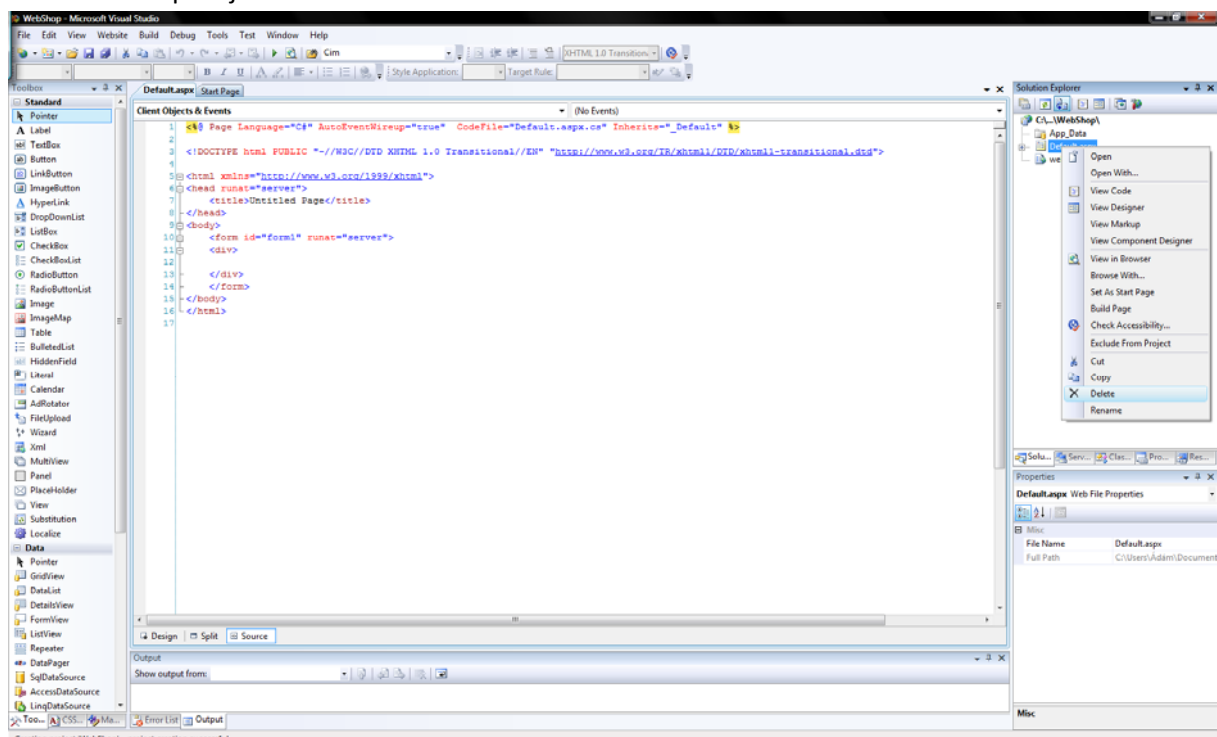
Site létrehozása

1. File / New Web site...
2. A Template-k közül válasszuk az ASP.NET-et, nyelvnek (Language) a Visual C#-ot, a Location-ban adjuk meg a WebSite mentési helyét.

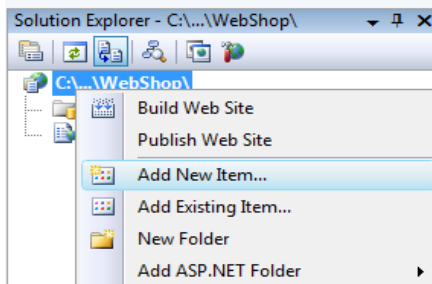


Design – kinézet

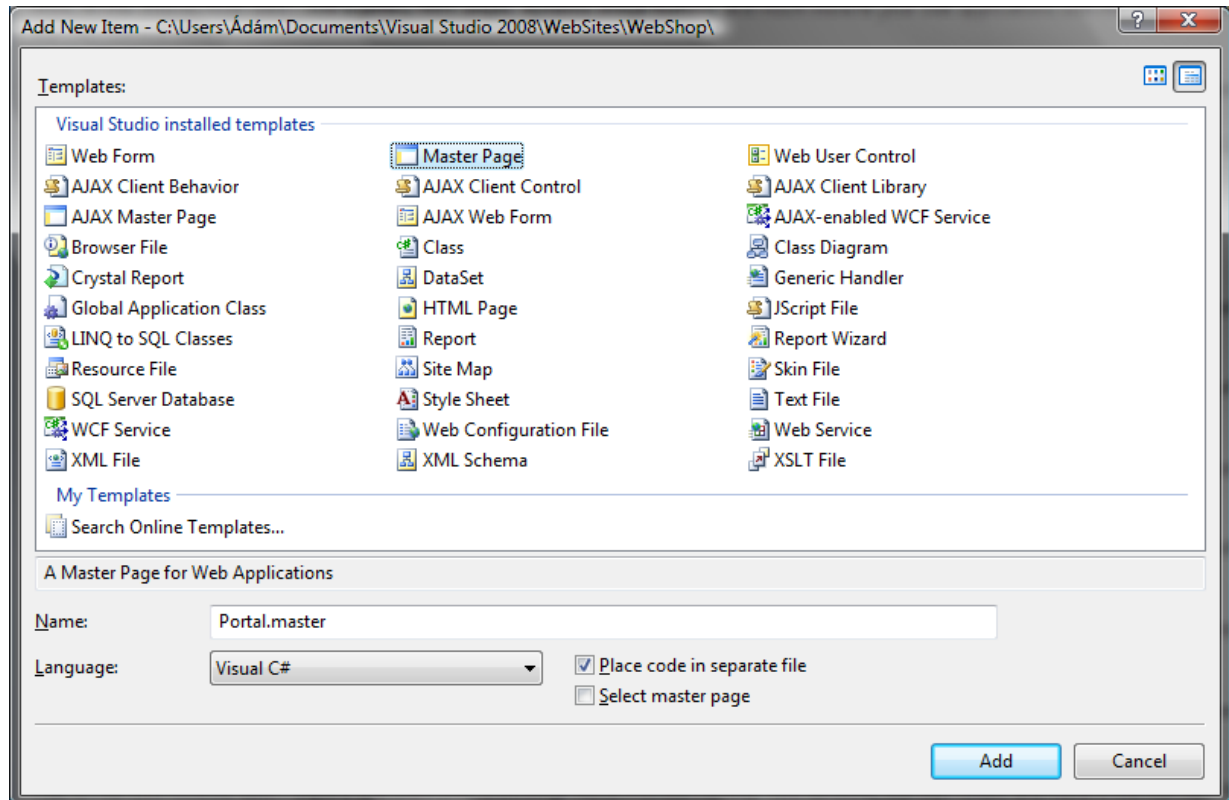
1. Mester oldal (Master Page) létrehozása.
2. Mivel oldalainkat egy sablon alapján készítjük el, a Visual Studio által készített Solution Explorer-ben töröljük ki a Default.aspx fájlt.



3. Master Page hozzáadása a projekthez: a Solution Explorer-ben a projekt nevén nyomjuk jobb egér gombot, majd: **Add New Item...**



a megjelenő ablakban válasszuk a Master Page template-t, majd adjunk egy nevet a mester oldalnak.



4. A mesteroldalunk lesz minden oldalunk alapja, az erre épülő oldalakon csak azokat a tartalmakat tudjuk átírni, ahova elhelyeztünk egy <asp:ContentPlaceHolder vezérlőt.

Mivel szeretnénk egy állandó fejléct is, ezért a body-ban lévő ContentPlaceHolder elé helyezzünk el egy kép (Image) vezérlőt, melynek neve (id) legyen: img_header. A tartalmat pedig helyezzük el egy 3 soros táblázatba valahogy így:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Portal.master.cs" Inherits="Portal" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>WebShop Portal</title>
  <asp:ContentPlaceHolder id="head" runat="server">
  </asp:ContentPlaceHolder>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <table style="width:800px;">
        <tr><td><asp:Image ID="img_title" runat="server"/></td></tr>
        <tr>
          <td>
            <asp:ContentPlaceHolder id="CPH_main" runat="server">
            </asp:ContentPlaceHolder>
          </td>
        </tr>
      </table>
    </div>
  </form>
</body>
```

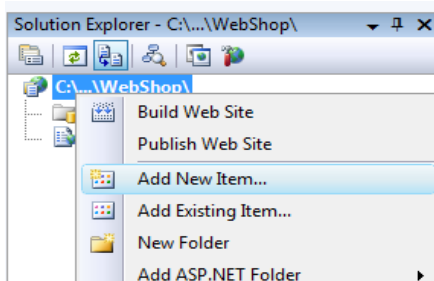
```

        </td>
      </tr>
    <tr><td></td></tr>
  </table>
</div>
</form>
</body>
</html>

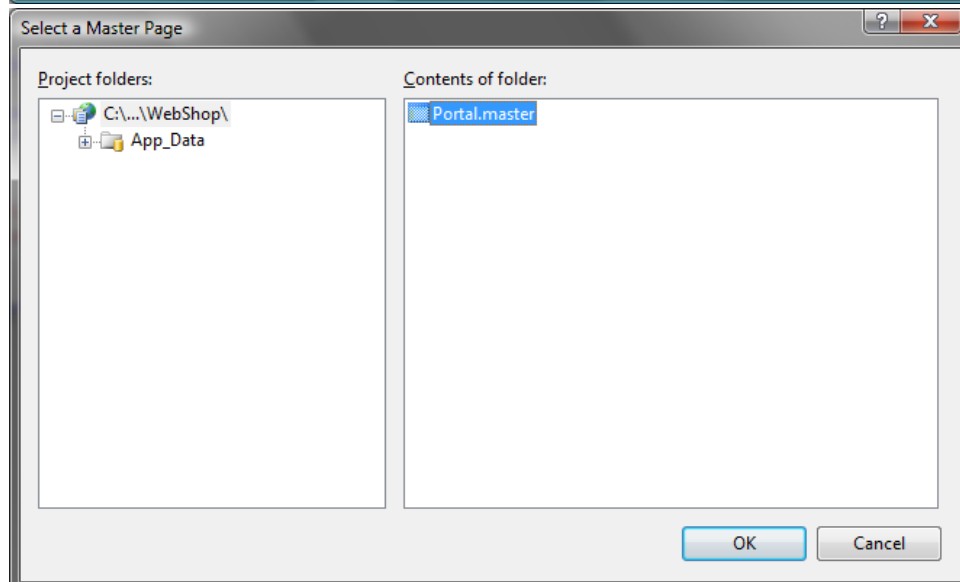
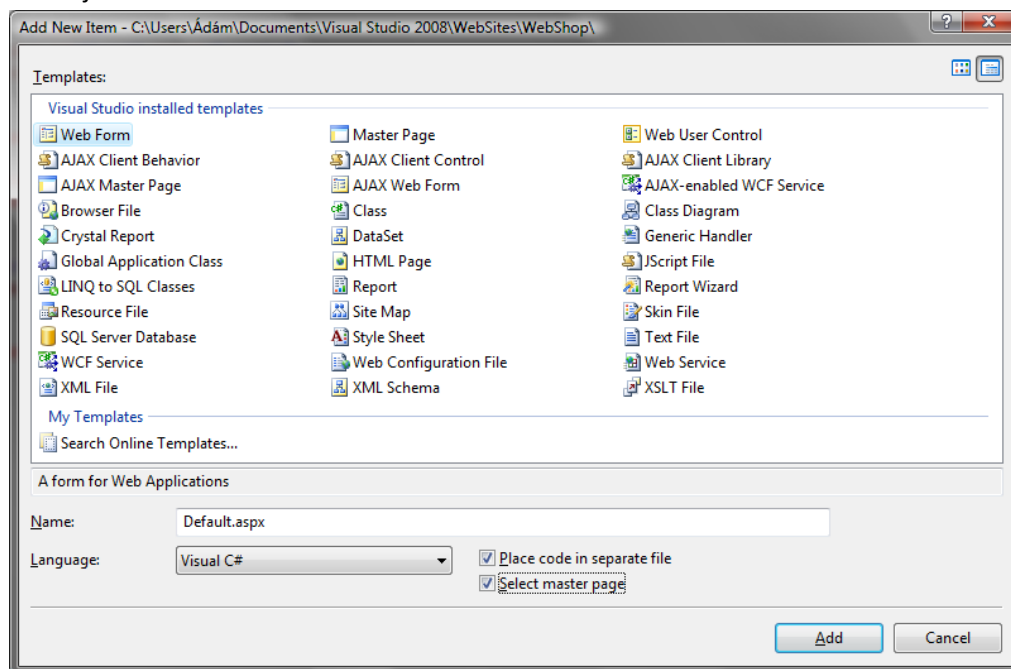
```

5. Elkészült tehát a design szerkezete, a kinézetét később készítjük el, most elkészítjük a master page alapján a szükséges oldalakat. Ehhez a Solution Explorer-ben a projekt nevén nyomjunk egy jobb egér gombot, majd

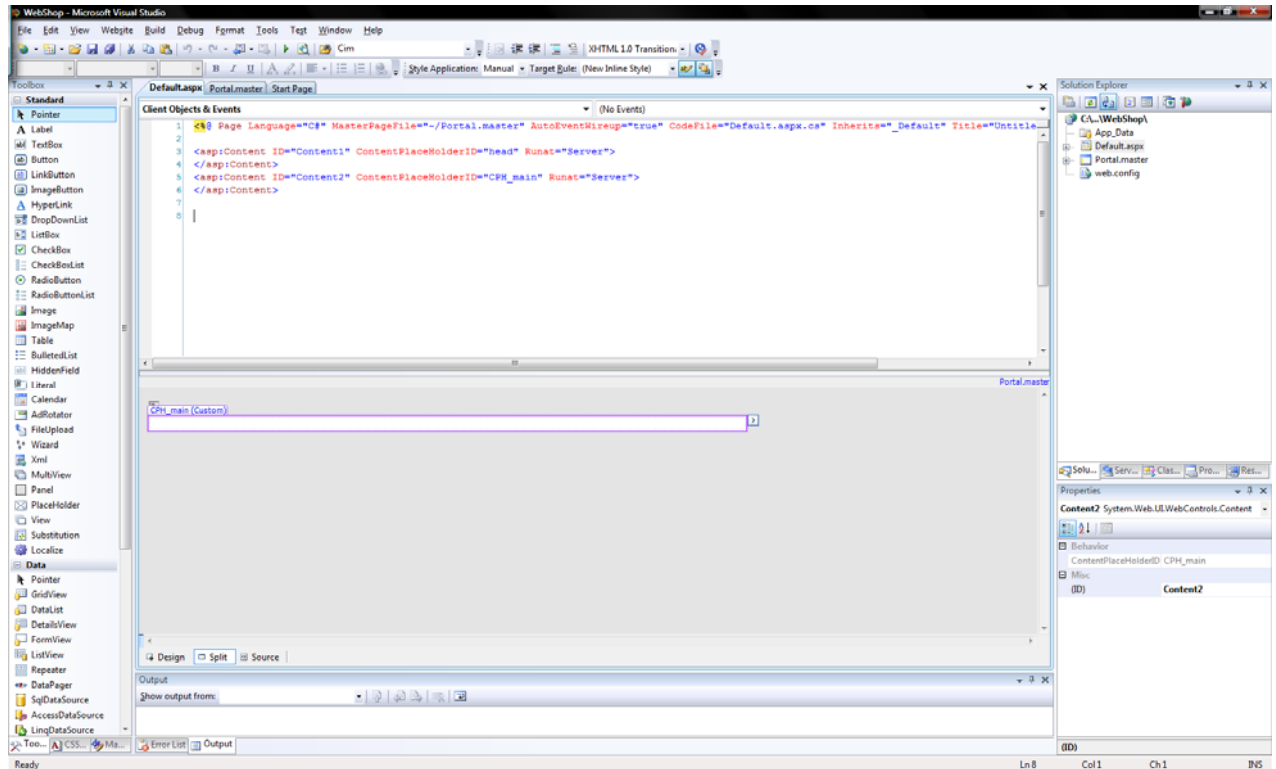
Add New Item...



A Template-k közül válasszuk ki a Web Form-ot, jelöljük ki a Select Master Page jelölőnégyzetet, majd az Add-ra kattintás után válasszuk ki az előzőleg létrehozott mesteroldalt (Portal.master), hogy legyen az oldal sablonja.

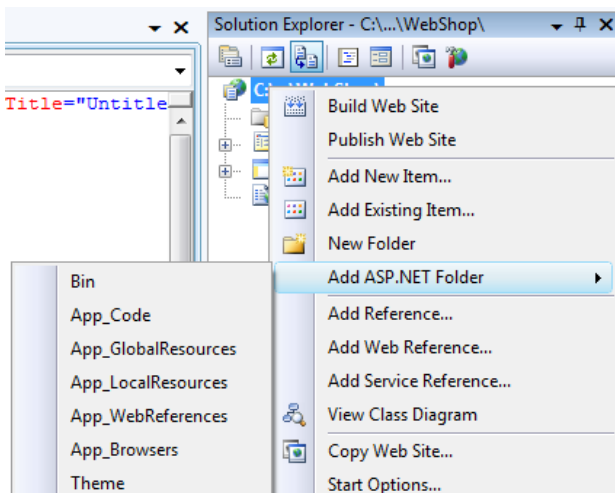


6. A létrejött Default.aspx oldalon csak a <asp:Content> vezérlőbe helyezhetünk el tartalmat.

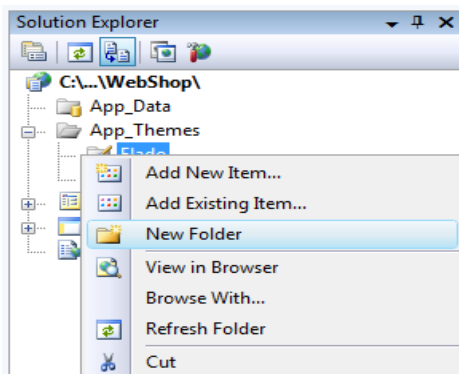


7. Az előzők alapján hozzunk létre az alábbi oldalakat is (a szükséges oldalakat akkor is létrehozhatjuk, amikor kellenek majd): **login.aspx**, **registration.aspx**.
8. Témák létrehozása: A mesteroldal segítségével az egységes elrendezést alakítottuk ki. A témákkal (themes) pedig az egységes színvilágot fogjuk. A témák igazi nagy ereje abban rejlik, hogy futásidőben dinamikusan cserélhetjük! Ez azt jelenti, hogy akár egyetlen egy gombnyomással a teljes színvilág, kinézet megváltoztatható. A téma nem más, mint fájlok gyűjteménye, ahol háromféle fájlípust halmozhatunk fel:
- **skin fájlokat** (bennük a vezérlők alapértelmezett vizuális megjelenését szabhatjuk meg, pl.: minden szövegbeviteli mező legyen 200 pixel széles, és legyen dőlt a benne lévő szöveg),
 - **css fájlokat** (a html objektumok kinézetét ezek segítségével szabhatjuk testre),
 - **erőforrás fájlokat** (ezek általában képfájlok, de bármilyen más fájl lehet, pl.: audio, videó).

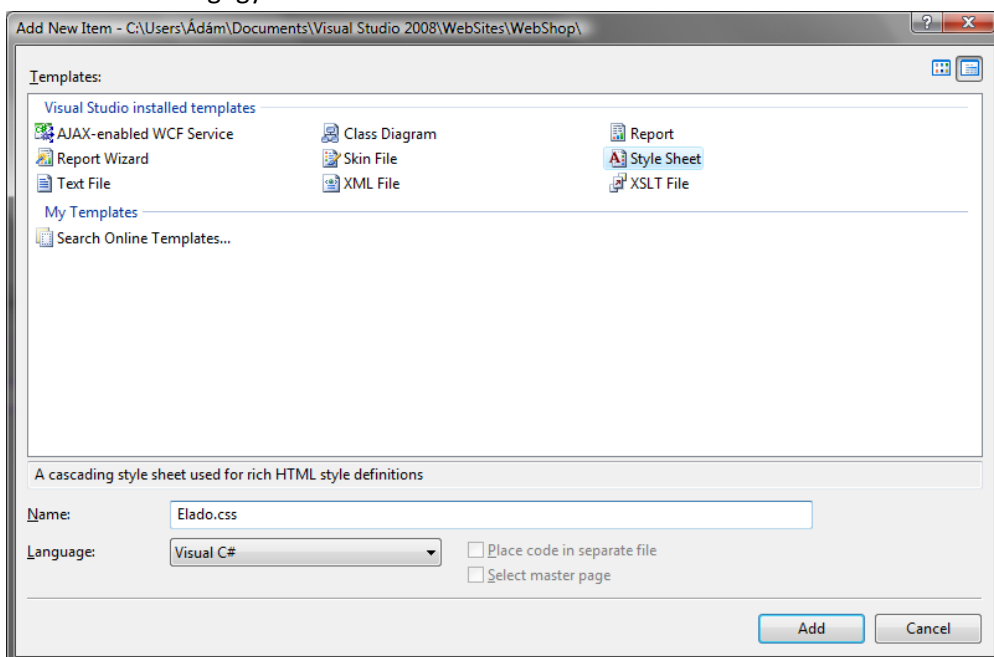
Mivel a témák fájlok gyűjteményei, ezért mappák segítségével reprezentáljuk őket. Az ASP.NET-ben van néhány speciális mappa, melyek bizonyos fájlípusuk összegyűjtésére szolgálnak pl.: App_Data (adatbázis, xml, stb.), App_Themes (témák), App_Code (saját osztályok, webszolgáltatások, stb.), stb.. Ilyen speciális mappákat úgy tudunk létrehozni, hogy jobb egér gombbal kattintunk a projekten, majd megkeressük az Add ASP.NET Folder-t, és ennek az almenüjéből választunk egyet. Mivel App_Themes nincs a listában, ezért válasszuk itt a Theme-t, amely majd automatikusan létrehozza hozzá ezt a mappát is.



9. Kétféle típusú felhasználónk, így témánk lesz (Vevo – normál vevő, Elado – viszonteladó). Adjuk azt a nevet az új mappának, hogy Elado! Kattintsunk most az App_Themes-re jobb egér gombbal, majd Add ASP.NET Folder >> Theme, és ennek pedig azt a nevet adjuk, hogy Vevo.
10. Mindkét témán belül hozzunk létre egy-egy Img nevű könyvtárat! Jobb egér gomb >> New Folder

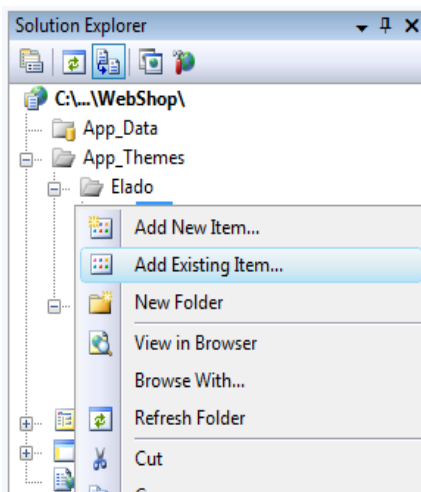


11. Adjunk egy-egy skin fájlt hozzájuk (adott theme-n jobb klikk >> add new item), a nevük legyen mondjuk a téma nevével megegyező.

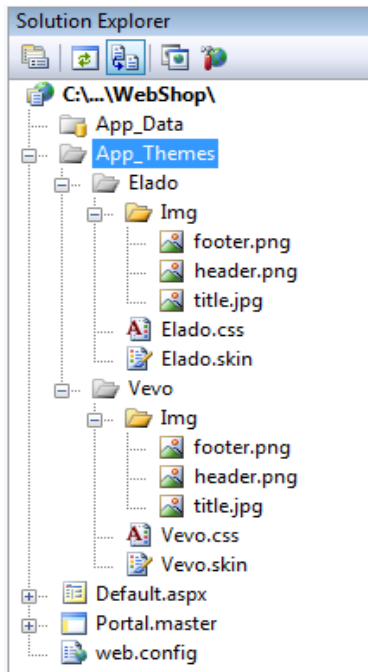


Végül pedig az előzőhöz hasonlóan adjunk egy-egy stíluslapot is (css-t) a témákhoz!

12. A szükséges kép fájlokat az Add Existing Item... menüpont segítségével importálhatjuk be az Img mappákba. (jobb klikk Img-n>> Add Existing Item..)



13. Ha mindent jól csináltunk, ezt kapjuk:



14. Nyissuk meg a css fájljainkat és definiáljuk bennük a **header** illetve **footer** osztályokat az alábbi módon:

```
.header
{
background-image:url("Img/header.png");
height:35px;
}
.footer
{
background-image:url("Img/footer.png");
height:35px;
}
```

15. Két fajta skin (bőr) létezik: alapértelmezett skin, illetve a nevesített skin. A kettő között az a különbség, hogy míg az alapértelmezett skin minden adott típusú vezérlőre érvényre jut, addig a nevesített skin csak az adott SkinID-val ellátott vezérlőkre fog vonatkozni. Jelen esetben mi a mesteroldalon lévő fejléc képet megjelenítő Image vezérlőket szeretnénk beállítani a skin fájlokból. Mivel több kép kontrollunk is lesz a webalkalmazásunkba, ezért az általános (default) skin nem jó, csak a nevesített (named) skin!

16. A skin fájlban belül egy adott típusú vezérlő megjelenését ugyanazzal a kóddal tudjuk leírni, mint ami az aspx fájlban van. A különbség összesen annyi, hogy itt nem szabad megadni az ID tulajdonságot! Mivel jelen esetben egy konkrét kép vezérlőt szeretnénk a bejelentkezett felhasználó típusának megfelelően megjeleníteni, nevesített skin-re van szükségünk, ezért a SkinID nevű attribútummal meg kell szabnunk a skin nevét. Íme a fejlécképünket megjelenítő Image vezérlő nevesített skin-je, ezt kell beleírni mindkét skin fájl végére:

```
<asp:Image runat="server" SkinId="title" ImageUrl="Img/title.jpg" Width="800px" Height="250px" />
```

17. Térjünk vissza most a mesteroldalunkhoz és állítsuk be a megfelelő helyeken a css osztályokat, illetve a SkinID-t:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>WebShop Portal</title>
  <asp:ContentPlaceHolder id="head" runat="server">
  </asp:ContentPlaceHolder>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <table style="width:800px;">
```

```
|  |
| --- |
| <asp:Image ID="img_title" runat="server" SkinID="title"/></td></tr> |

```

18. A témakezeléssel kapcsolatban már csak annyi dolgunk maradt hátra, hogy megszabjuk, mely oldalon melyik skin legyen az alapértelmezett. Ezt beállíthatnánk minden egyes oldalon külön-külön, de arra is van lehetőségünk, hogy az alkalmazás összes oldalára egy helyen beállítsuk. A web.config fájl ez a bizonyos hely, ahol alkalmazásszint beállításokat szabhatunk meg.
19. Nyissuk meg a web.config fájlt, majd keressük meg a system.web tagen belül a pages taget, végül pedig adjuk meg a theme attribútumot Vevo értékkel: `<pages theme="Vevo">`



20. CTRL+F5-el leellenőrizhetjük, hogy minden rendesen működik-e. Ha igen, valami hasonlót kell kapnunk:

