

ASP.NET 3.5 Tutorial I. rész

(Vs telepítés, Design, Login)

Sziasztok! Egy kétrészes tutorial-sorozat keretében szeretnénk bemutatni Nektek, hogy **ASP.NET 3.5**-ben milyen egyszerűen lehet *webalkalmazásokat* készíteni. A tutorialok alatt egy egyszerű kis párkereső oldalt fogunk fejleszteni. Kezdjünk is hozzá!

Előkészületek

1. ASP.NET-ben való fejlesztéshez legelőször is be kell szereznünk a **Visual Studio 2008**-at. Ezt az **MSDNAA** program keretén belül teljesen ingyen és bérmentve megtehetjük. Az alábbi blog bejegyzésből keressük ki az egyetemünk/főiskolánk msdnaa oldalának a címét, majd regisztráljunk az oldalra, végül pedig lépünk be!

<http://msportal.hu/blogs/msdnaa/archive/2007/10/04/msndaa-helyi-kapcsolatok.aspx>

2. A *Software* fül alatt a legördülő listából keressük ki a **Visual Studio 2008 Professional Edition – DVD**-t, majd kattintsunk az *OK* gombra, utána pedig az *Add to Cart*-ra! Fogadjuk el a licencszerződést, majd pedig *Check out*-oljuk a kosarunk tartalmát! Kattintsunk a *Next* gombra, majd végül a *Download*-ra! Ilyenkor nem a teljes 3,3gb-os Visual Studio (VS) fog letöltődni, hanem csak egy pár száz kilóbájtos letöltő program.

3. Indítsuk el a progit, majd pedig állítsuk be, hova szeretnénk lementeni a VS08-at! Letöltés után lesz egy ***.iso képfájlunk**, melyet vagy írunk ki DVD-re, vagy pedig *mount*-oljuk fel valamilyen virtuális meghajtóra. Miután elindult a telepítő, kattintsunk az egyetlen jelenleg elérhető linkre, vagyis az *Install Visual Studio 2008*-ra!

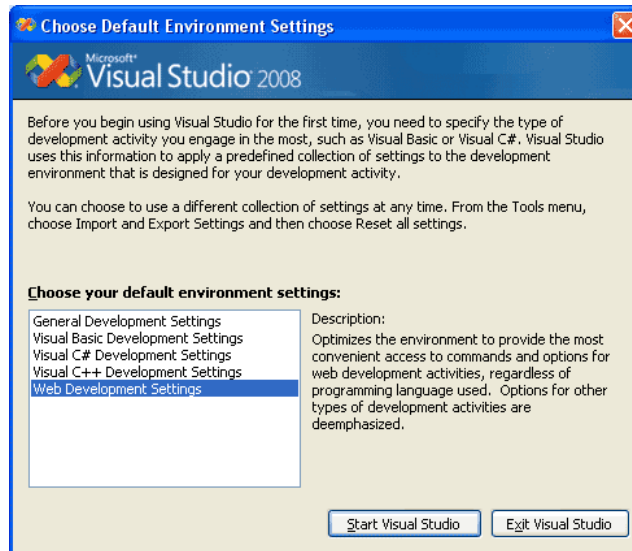


Next, majd ismét fogadjunk el egy licencszerződést! Válasszuk a *Default* telepítő csomagot, végül pedig klikk az *Install*-ra! (Ha esetleg a *Custom* szimpatikusabb Neked, akkor mindképpen hagyjad bepipálva a **Visual C#**-ot, illetve a **Visual Web Developer**-t!!!)

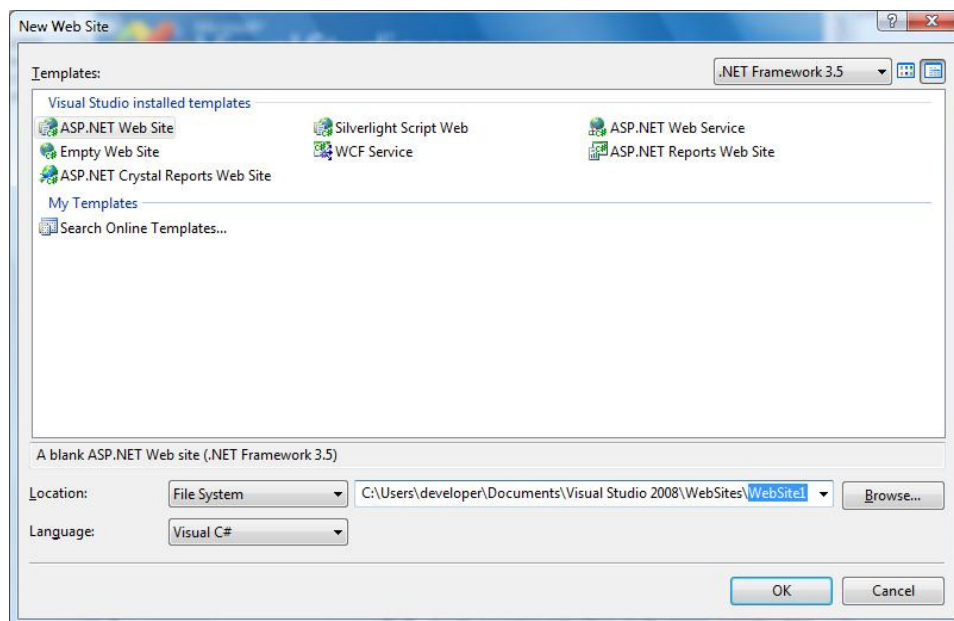
4. Ilyenkor egy jó pár komponens föl fog települni, amely eltart egy ideig. A telepítés végeztével a rendszer felajánlja, hogy újraindítja a gépet, éljünk ezzel a lehetőséggel, mert különben nem fog minden rendeltetésszerűen működni!

Gyors ismerkedés a Visual Studio 2008-cal

1. Indítsuk el a Visual Studio 2008-at! Első indításkor el fog indulni egy *önkonfiguráló folyamat*, mely néhány másodpercig eltart, majd utána megkérdezi a VS, hogy milyen *alapértelmezett környezeti beállításokat* szeretnénk. Ez főleg az alap billentyűzetkiosztást szabja meg, mi most válasszuk a **Web Development Settings**-et! Végül katt a *Start Visual Studio* gombra!

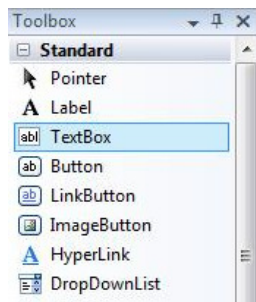


2. Miután elindult a Visual Studio válasszuk a **File >> New Web Site** menüpontot! Ilyenkor az alábbi ablak fog felpattanni:



Nekünk egy **ASP.NET Web Site**-ra lesz szükségünk (ez van alapértelmezetten is kijelölve), így ebben az ablakban mindössze annyi dolgunk van, hogy a **Location** végén a *WebSite1*-t lecseréljünk *partner_finder*-re, illetve a **Language**-t átállítsuk **Visual C#**-ra!

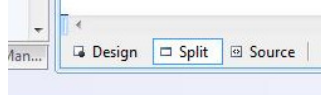
3. Ilyenkor létrejön egy kezdő oldal, amelyet **Default.aspx**-nek neveznek. Ebben fogjuk leírni az oldalaink kinézetét (ezt hívjuk melleleg *markup*-nak). Lesz egy **Default.aspx.cs** fájlunk is, amelyben majd a programkódokat fogjuk írogatni. A többi fájlról/mappáról majd kicsit később. Gyorsan fussuk át, mire is jó ez a halom ablak, ami most hirtelen elének tárult.



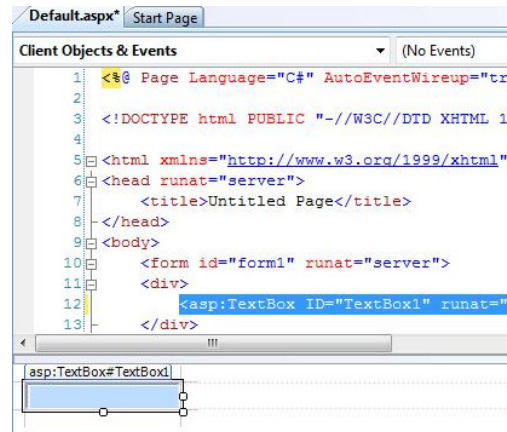
Toolbox – Ez a kifogyhatatlan fegyverraktárunk. Innen egyszerűen *drag&drop* módszerrel tudunk a középen lévő fehér formunkra

vezérlőket húzkodni. Természetesen csak néhányat fogunk ezek közül használni ebben a tutorialban.

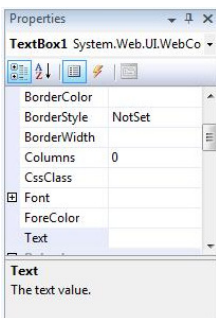
Munkaterület – A képernyő közepén található nagy fehér terület. Ez a weboldalunk, amelyet ezek után csak (*web*)*form*-nak fogunk hívni. Fölötte váltogathatunk a megnyitott fájljaink között, alatta pedig a kinézetváltó található.



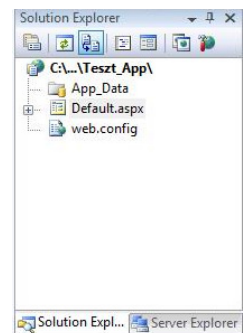
Kinézetváltó – Három nézet közül választhatunk. Van egy úgynevezett *Source* nézet, amikor az oldal kinézetét leíró markup kódot láthatjuk. Van egy *Design* nézet, ahol az oldal végleges megjelenését csekkolhatjuk le. Végül pedig ezen két megjelenítési forma kombinációja a *Split*, ahol felül lesz látható a markup kód, alatta pedig a wysiwyg alapú weboldal megjelenítő.



Solution Explorer – Ebben az ablakban azt láthatjuk, hogy mely fájlok tartoznak a webalkalmazásunkhoz. Alul van egy másik füle is a panelnek, amit **Server Explorer**-nek hívnak, ahonnan tudunk csatlakozni adatbázisokhoz, illetve tudjuk menedzselni is őket minimális szinten.

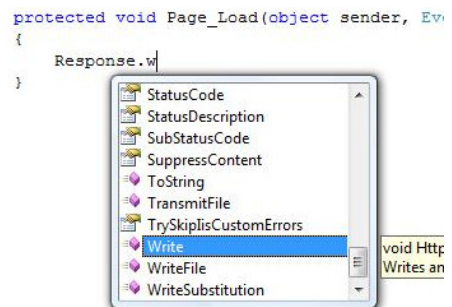


Properties – Itt egy adott vezérlő tulajdonságait szerkeszthetjük. Jelenesetben egy **TextBox** (*beviteli mező*) tulajdonságai láthatóak. Az alapértelmezett tulajdonsága a *Text* (a benne megjelenő szöveg állítható be vele).



4. Végül pedig a VS gyorstalpaló utolsó pontjaként nézzük meg, milyen segítséget kapunk kódírás közben. Kattintsunk kétszer a formra, vagy pedig nyomjuk meg az **F7** billentyűt, hogy megnyíljon a **Default.aspx.cs** fájl.

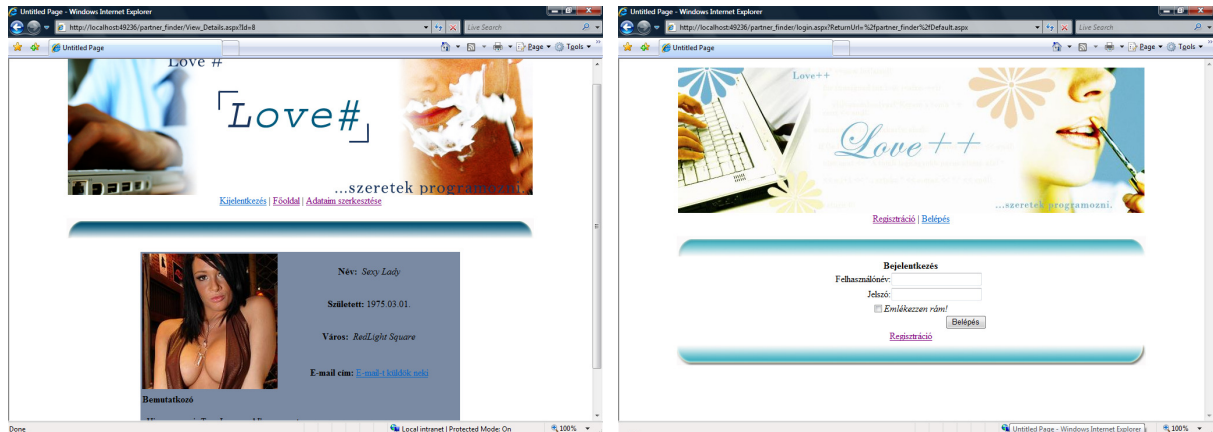
ASP.NET-ben eseményekhez (pl.: gombra való kattintás), illetve akciókhoz (pl.: szövegdobozba beírt szöveg megváltozása) rendelhetünk kódokat. A kódírást egy úgynevezett **IntelliSense** ablak könnyíti meg, amelyből gyorsan kiválaszthatjuk a nekünk szükséges változók nevét, objektumok metódusait, stb. (**Tab** gomb segítségével tudjuk beilleszteni a kiválasztott „elemet”).



Portál felépítése

Most, hogy átrágtuk magunkat az alapokon, ismerkedjünk meg az elkészítendő oldalunk struktúrájával! A párkereső portálunk felépítése eléggé egyszerű lesz, mindössze 5 oldalból fog állni. Lesz egy regisztrációs oldal, illetve egy beléptető oldal. Ezekon kívül szükségünk lesz még egy főoldalra is, ahol az ellenkező nembeli regisztrált tagokat fogjuk kilistázni. Itt csak minimális adat fog megjelenni róluk, ezért innen el lehet majd jutni egy olyan oldalra, ahol minden információ megtekinthető lesz a kiszemelt áldozatról.

Végül pedig egy olyan oldalt is el fogunk készíteni, ahol a saját adatainkat szerkeszthetjük. Íme, két kép ízelítőként a végeredményről:



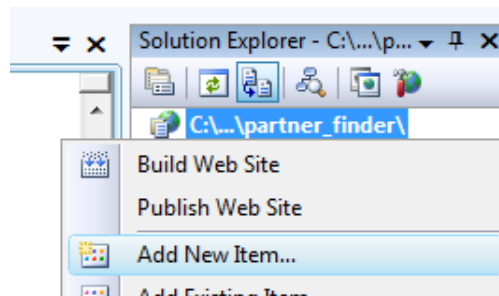
Design kialakítása

Amint az a képekből remélem látszik, kétfajta megjelenése lesz az oldalnak, lesz egy *fiús* (**love #**), illetve egy *lányos* (**love++**). Ezt **témák** segítségével fogjuk megvalósítani. Ahhoz, hogy minden oldalnak egységes legyen a megjelenése (középre zárt, fejléckép, illetve az oldalak szövegét közrefogja két lekerekített sarkú szegély) **mesteroldalt** fogunk használni.

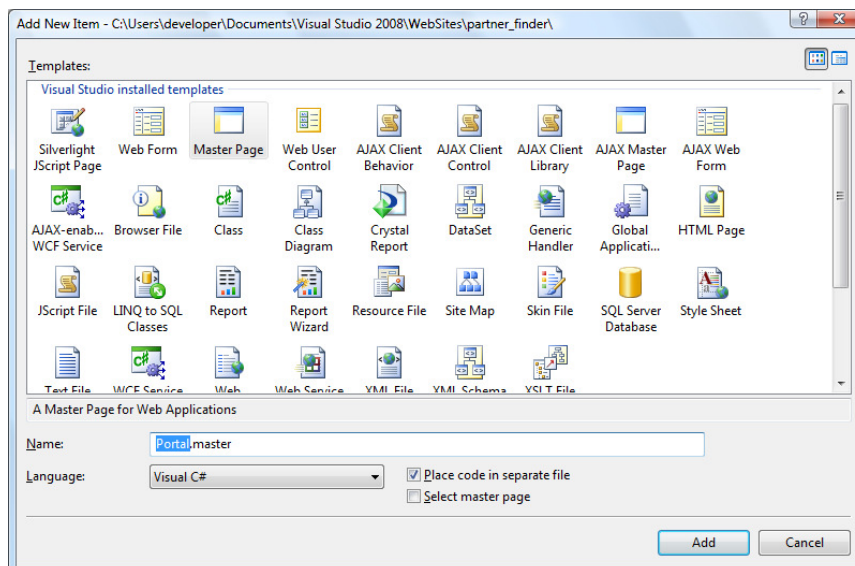
- Mesteroldal létrehozása

1. Töröljük az automatikusan létrehozott Default.aspx oldalt a Solution Explorerben! Jobb klikk a fájlra >> **Delete**.

2. Adjunk egy mesteroldalt (**masterpage**) az egységes megjelenés érdekében az alkalmazásunkhoz! A Solution Explorer legtetején lévő partner_finder projektre kattintsunk jobb egér gombbal, majd pedig itt **Add New Item..** menüpontra.



3. Az ilyenkor felugró új ablak felső részében keressük meg a *Master Page* ikont, majd adjuk neki azt a nevet, hogy **Portal.Master!**



4. A mesteroldalunk most így néz ki:

```

...
<head runat="server">
  <title>Untitled Page</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>
...

```

A master pagek úgy működnek, hogy bennük kialakítjuk az oldalak alapkinézetét (elrendezését) és úgynevezett *helyőrzők* (**ContentPlaceHolder** (CPH) vezérlők) segítségével megszabjuk, hogy hova kerülhetnek majd az oldalak szövegei (tartalmi). Jelen esetben nekünk a *head* azonosítójú CPH-ra nincs szükségünk, ezért töröljük is!

5. Az egységes kinézetünk egyik alapja az, hogy minden középre van igazítva, ezért a div tag **style** attribútumán keresztül ezt állítuk is be: `style="text-align:center;"`.

6. Minden oldal tetején meg akarjuk jeleníteni a fejlécképünket, ezért húzzunk be közvetlenül a div tag alá egy **Image** vezérlőt a Toolbox-ról! Legyen az **ID**-ja *img_title*!

7. A szegélyek elhelyezéséhez táblázatot fogunk használni. 3sora lesz, az elsőben a felső szegély lesz, a másodikban a ContentPlaceHolder-ünk, a harmadikban pedig az alsó szegély. Most egyenlőre még csak annyi formázást végezzünk el, hogy a táblázat szélességét beállítjuk 800pixelre. Nevezzük át a ContentPlaceHolder-ünk azonosítóját **CPH_main**-re!

8. A div-ünknek most így kéne kinéznie:

```

<div style="text-align:center">
  <asp:Image ID="img_title" runat="server" />

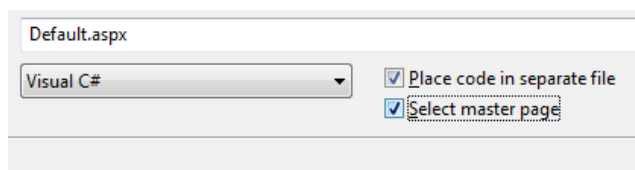
```

```

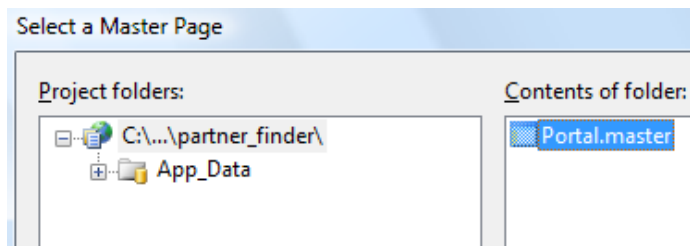
<table style="width:800px;">
  <tr><td/></tr>
  <tr>
    <td >
      <asp:ContentPlaceHolder id="CPH_main" runat="server">
      </asp:ContentPlaceHolder>
    </td>
  </tr>
</table>
</div>

```

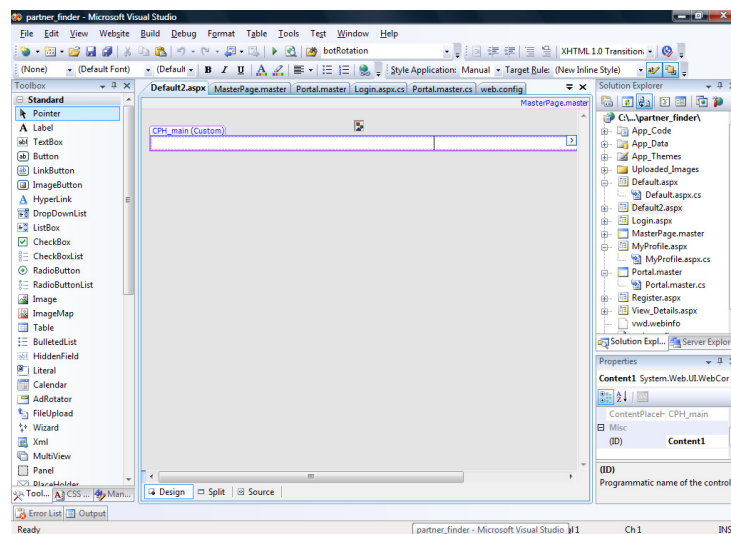
9. Hozzuk újból létre a Default.aspx-t! Hasonlóképpen, mint a mester oldalnál, itt is az Add New Item ablakot használjuk! Válasszuk ki a *Web Form* ikon-t, majd alul kattintsuk be a **Select Master Page** jelölőnégyzetet, végül pedig klikk az **Add**-re!



A következő oldalon lehetőségünk van arra, hogy kiválasszuk, melyik mester oldalhoz tartozzon az oldalunk, de mivel csak egyetlen egy master page-e van az alkalmazásunknak, ezért egyszerűen csak kattintsunk az *OK* gombra!



10. Az ilyenkor létrejött új webformunkban lesz egy **Content** vezérlő, amely hivatkozik a mester oldalon lévő CPH_main helyőrzőre. Csak a Content-en belül helyezhetünk el szöveget, vezérlőt, tartalmat! Ha Design nézetre váltunk, akkor láthatjuk, hogy csak a Content területe szerkeszthető, ezért ez fehér, a mester oldaltól örökölt rész pedig szürke.



11. Adjuk a projektünkhöz a következő oldalakat (ne felejtsük el bekattintani alul a mester oldal használatot!): **View_Details.aspx, My_Profile.aspx, Login.aspx**, illetve **Registration.aspx**

- Témák létrehozása

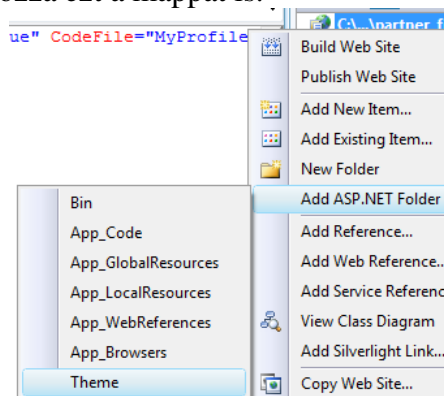
A mesteroldal segítségével az egységes elrendezést alakítottuk ki. A témákkal (**themes**) pedig az egységes színvilágot fogjuk. A témák igazi nagy ereje abban rejlik, hogy futásidőben dinamikusan cserélhetőek! Ez azt jelenti, hogy akár egyetlen egy gombnyomással a teljes színvilág, kinézet megváltoztatható.

A téma nem más, mint fájlok gyűjteménye, ahol háromféle fájltypust halmozhatunk fel:

- **skin fájlokat** (bennük a vezérlők alapértelmezett vizuális megjelenését szabhatjuk meg, pl.: minden szövegdoxoz 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).

1. Mivel a témák fájlok gyűjteményei, ezért mappák segítségével reprezentáljuk őket. Az ASP.NET-ben van néhány *speciális mappa*, melyek bizonyos fájltypusok 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.

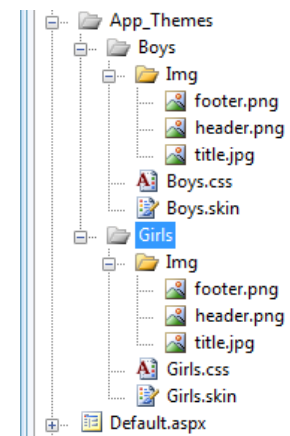


2. Adjuk azt a nevet az új mappának, hogy **Girls!** Kattintsunk most az App_Themes-re jobb egér gombbal, majd Add ASP.NET Folder >> Theme, és ennek pedig azt a nevet adjuk, hogy **Boys!**

3. Mindkét témán belül hozzunk létre egy-egy **Img** nevű könyvtárat! Adjunk egy-egy skin fájlt hozzájuk (adott theme-n jobb klikk >> add new item), a nevük lényegtelen, ezért legyen mondjuk a téma nevével megegyező. Végül pedig adjunk egy-egy stíluslapot is (css-t) a témákhoz!

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

Nálam a fejlécképek **title** nevet kaptak és 800*250 nagyságúak. A szegély képek (**header, footer**) pedig 800*35 mérettel rendelkeznek.



5. 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;
}
```

6. Két fajta 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 fájljokból. Mivel több kép kontrollunk is lesz a webalkalmazásunkba, ezért az általános (*default*) bőr nem jó, csak a nevesített (*named*) skin!

7. A skin fájlban belül egy adott típusú vezérlő megjelenését ugyanazzal a markup kóddal tudjuk leírni, mint ami az aspx fájlban van. A különbség összesen annyi, hogy itt nem szabad megadni az *ID* tulajdonságot! Ellenben mivel mi most nevesített skin-t akarunk készíteni, ezért a **SkinID** nevű attribútummal meg kell szabnunk a bőrünk nevét. Íme a fejlécképünket megjelenítő Image vezérlő nevesített skin-je:

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

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

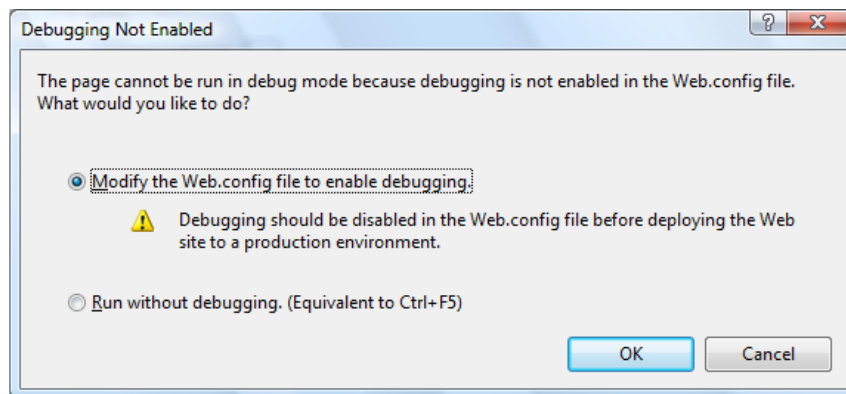
```
<asp:Image ID="img_title" runat="server" SkinID="title" />
<table style="width:800px;">
    <tr><td class="header" /></tr>
    <tr>
        <td>
            <asp:ContentPlaceHolder id="CPH_main" runat="server">
                </asp:ContentPlaceHolder>
            </td>
        </tr>
    <tr><td class="footer" /></tr>
</table>
```

9. A bőrözéssel 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 alkalmazásszintű beállításokat szabhatunk meg.

10. Nyissuk meg a web.config fájlt, majd keressük meg a **system.web** tagen belül a **pages** taget, végül pedig adjuk meg a **theme** attribútumot **Girls** értékkel: `<pages theme="Girls">!`

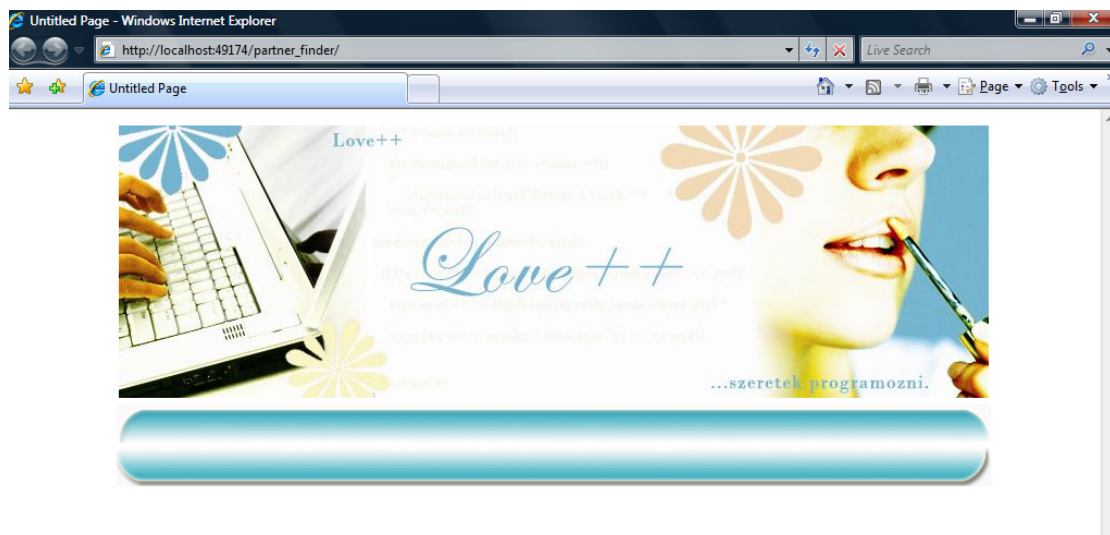
```
web.config Default.aspx Portal.master
<?xml version="1.0"?>
<configuration>
  <configSections>...
  <appSettings/>
  <connectionStrings/>
  <system.web>
    ...
    <compilation ...
    ...
    <authentication mode="
    ...
  <pages theme="Girls">
    <controls>
      <add tagPrefix="asp"
      <add tagPrefix="asp"
    </controls>
```

11. Próbáljuk ki mit is alkottunk idáig, nyomjuk meg az **F5** gombot, vagy az eszköztáron kattintsunk a zöld jobbra mutató nyílra! Ilyenkor az alábbi ablak fog felpattanni:



Itt arról tájékoztat minket a Visual Studio, hogy nincsen beállítva a web.config-ban, hogy az alkalmazásunk *hibakereső módban* fusson, szeretnénk-e, hogy ő beállítsa helyettünk. Jelen esetben mi most nem élünk ezzel a lehetőséggel, mert ha már a DEMO is bugos, akkor ott már nagy bajok vannak ☺. Tehát válasszuk a második jelölőnégyzetet és kattintsunk az *OK* gombra!

12. Ha idáig mindent jól csináltunk, akkor valami hasonlóan kéne most megjelennie:

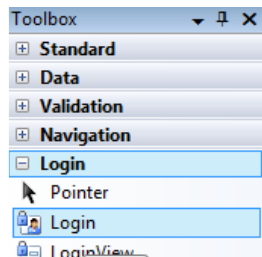


Felhasználók azonosítása/létrehozása

Az alap design-nal nagyjából el is készültünk, a későbbiekben már csak egy-két apróságot fogunk változtatni a megjelenésen, ezért térjünk most át egy másik fontos részére a portálunknak: a **felhasználó kezelésre!**

- Felhasználók hitelesítése

1. Először is nyissuk meg a **Login.aspx** oldalt, majd a Toolbox *Login* tab-jéről húzzunk a formunkra egy **Login** vezérlőt.




2. Sokféleképpen magyárosíthatjuk a Login kontrollunkat, de talán az egyik legegyszerűbb megoldás a *properties* ablak használata. Mi most csak minimális szinten magyárosítunk (a hibaüzeneteket nem fordítjuk le), így az alábbi markup-ot kell, hogy kapjunk:

```
<asp:Login ID="Login1" runat="server"
  TitleText="Bejelentkezés" LoginButtonText="Belépés"
  UserNameLabelText="Felhasználónév:" PasswordLabelText="Jelszó:"
  RememberMeText="&lt;i&gt;Emlékezzen rám!&lt;/i&gt;"
  <TitleTextStyle Font-Bold="True" />
</asp:Login>
```

A fentebbi kód a magyarázaton kívül tartalmaz még két formázási beállítást is: a fejléc szövegét félkövérre vettük, illetve az emlékezzen rám szöveget pedig dőltre állítottuk. Ahhoz, hogy a beléptető oldalról is el lehessen jutni a regisztrációs oldalra, az alábbi két argumentumot adjuk még hozzá a Login kontrollhoz:

```
CreateUserText="Regisztráció" CreateUserUrl="~/Registration.aspx"
```

3. A felhasználói felületével készen is vagyunk a bejelentkeztetésnek, már csak a mögötte lévő adatbázist kell beállítanunk, amelyhez van egy nagyon hasznos kis segédeszköz a **Web Site Administration Tool (WSAT)**. Ezt két helyről is elérhetjük, vagy a Solution Explorer tetején rákattintunk erre az ikonra: , vagy pedig **WebSite** menüen belül az **ASP.NET Configuration** menüpontra.

4. Miután rákattintunk valamelyikre, egy weboldal fog betöltődni a böngészőnkbe, amelyek a főoldala így néz ki:



Welcome to the Web Site Administration Tool

Application:/partner_finder
Current User Name:DEVELOPER-PC\DEVELOPER

| | |
|---|--|
| Security | Enables you to set up and edit users, roles, and access permissions for your site. Site is using windows authentication for user management. |
| Application Configuration | Enables you to manage your application's configuration settings. |
| Provider Configuration | Enables you to specify where and how to store administration data used by your Web site. |

5. Kattintsunk a **Security** linkre, majd itt a **User** nevű dobozban a **Select authentication type** -ra!

Click the links in the table to manage the settings for your application.

| Users | Roles | Access Rules |
|--|---|--|
| The current authentication type is Windows . User management from within this tool is therefore disabled. Select authentication type | Roles are not enabled Enable roles Create or Manage roles | Create access rules Manage access rules |

Ezek után a *From a local network* választókapcsolóról váltsunk át a *From the internet* radiobutton-re. Itt most azt állítottuk be, hogy ne az alapértelmezett Windows hitelesítést használja a rendszerünk a felhasználók azonosításához, hanem a Login.aspx oldalunkat! Végül kattintsunk lent a *Done* gombra!

6. A gomb megnyomása után a WSAT visszatér a Security oldalra, ahol láthatjuk, hogy a Users doboz kicsit megváltozott

Click the links in the table to manage the settings for your application.

| Users | Roles | Access Rules |
|---|---|--|
| Existing users: 0 Create user Manage users Select authentication type | Roles are not enabled Enable roles Create or Manage roles | Create access rules Manage access rules |

Mivel a felhasználók hitelesítését (illetve kezelését) átvettük a Windows-tól, ezért nekünk kell gondoskodnunk a userekről. Hozzunk létre egy teszt felhasználót, hogy majd ki tudjuk próbálni a rendszerünket, tehát kattintsunk a **Create user** linkre!

7. A felhasználónév legyen mondjuk **Admin**. A jelszó pl.: **M\$_Port@1** (fontos, hogy 7 karakternél hosszabb legyen és tartalmazzon legalább egy nem alfanumerikus karaktert is!). Az e-mail címnek adjuk meg a saját e-mail címünket! Biztonsági kérdésnek, illetve válasznak pedig valamilyen olyan párost találjunk ki, amit csak mi tudhatunk, ugyanis erre a jelszó emlékeztető rendszernek van szüksége!

Create User

Sign Up for Your New Account

User Name:

Password:

Confirm Password:

E-mail:

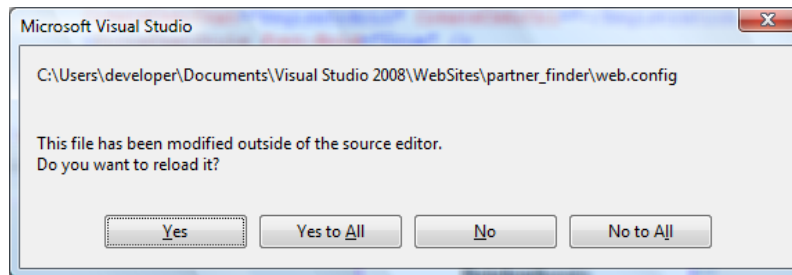
Security Question:

Security Answer:

8. Kattintsunk a *Create User* gombra, hogy létrejöjjön a felhasználónk. Ha sikerült létrehozni a rendszernek, akkor egy új ablakban tájékoztat minket erről, itt klikkeljünk majd a *Continue* gombra. Végül kattintsunk a *Back*-re!

9. ASP.NET-es alkalmazások esetén *mappa alapú biztonság* állítható be. Ez azt jelenti, hogy ha egy mappába bepakoljuk a védeni kívánt oldalainkat, akkor az adott mappában lévő tartalmakhoz a hozzáférést korlátozhatjuk. A mi portálunknál nem fogjuk korlátozni semelyik oldalhoz a hozzáférést, egyszerűen csak más adatot fogunk megjeleníteni a belépett és az *anonymous*(vendég) felhasználóknak, ezért zárjuk is be a WSAT-ot!

10. Ilyenkor az alábbi ablak fog felpattanni, mely arról tájékoztat minket, hogy a web.config megváltozott, és szeretnénk-e, hogy újra betöltse a Visual Studio:



Kattintsunk a *Yes* vagy a *Yes to All* gombra!

11. Indítsuk el a webalkalmazásunkat és navigáljuk el a Login.aspx oldalra, majd próbáljunk meg bejelentkezni. Ha sikerül belépünk, akkor automatikusan a Default.aspx oldalra fog minket átirányítani a rendszer.

12. A főoldalunk még eléggé gyér tartalomügyileg, ezért nem is tudjuk eldönteni az oldal láttán, hogy be vagyunk-e jelentkezve vagy sem. Ezért most egy olyan menüt fogunk elkészíteni a mesteroldalon, amely mást fog mutatni, ha vendékként nézzük az oldalt, és megint mást, ha bejelentkezett felhasználóként. Tehát nyissuk meg újból a Portal.master fájlnkat, és az Image vezérlőnk alá közvetlen helyezzünk el egy **LoginView** vezérlőt!

13. Ez a kontroll arra való, hogy egy adott helyen különböző tartalmakat jelenítsünk meg attól függően, hogy belépett-e már a felhasználó vagy még nem. Úgynevezett **sablonok** (*template*-ek) segítségével szabhatjuk meg, hogy mik jelenjenek meg a két különböző állapotban. A templatek használata eléggé egyszerű: az adott vezérlő (jelen esetben a LoginView) nyitó és záró tag-jei között definiálni kell a template-t, a tag nyitó és záró részével, majd itt a kettő között készíthetjük el a tartalmat.

14. A LoginView-n belül kétfajta sablont definiálhatunk: egy **AnonymousTemplate**-t, illetve egy **LoggedInTemplate**-t. Az első template-n belül („*vendég módban*”) két linkünk lesz: az egyik a regisztrációs oldalra fog mutatni, míg a másik a bejelentkező oldalra. A login oldalra mutató linkünk nem egy szimpla HyperLink-kel lesz megoldva, mint a reg oldalra mutató, hanem egy **LoginStatus** vezérlővel. Ez a kontroll attól függően, hogy bejelentkezett felhasználóként nézzük az oldalt vagy nem, egy **Login** vagy egy **Logout** linket jelenít meg. Nézzük meg, hogyan néz ki most a markup-unk, ha mindezt összerakjuk:

```
...<asp:Image ID="img_title" runat="server" SkinID="title" /><br/>
<asp:LoginView ID="LV" runat="server">
  <AnonymousTemplate>
    <asp:HyperLink ID="HL_reg" NavigateUrl="~/Registration.aspx"
runat="server">Regisztráció</asp:HyperLink> |
    <asp:LoginStatus ID="LS_login" runat="server" LoginText="Belépés" />
  </AnonymousTemplate>
  <LoggedInTemplate>
  </LoggedInTemplate>
</asp:LoginView>
<table style="width:800px;">...
```

15. A LoggedInTemplate definiálása van már csak hátra! Itt három linkünk lesz: egy kijelentkezés (loginstatus), egy főoldalra mutató (hyperlink), illetve egy myprofile-ra hivatkozó. Íme a sablonunk, tartalommal feltöltve:

```
<LoggedInTemplate>
  <asp:LoginStatus ID="LS_logout" runat="server"
LogoutPageUrl="~/Default.aspx" LogoutText="Kijelentkezés" />
```

```

| <asp:HyperLink ID="HL_main" NavigateUrl="~/Default.aspx"
runat="server">Főoldal</asp:HyperLink> |
<asp:HyperLink ID="HL_edit" NavigateUrl="~/MyProfile.aspx"
runat="server">Adataim szerkesztése</asp:HyperLink>
</LoggedInTemplate>

```

16. Ha most megpróbálunk bejelentkezni a portálra, akkor már kapni fogunk (igaz még eléggé csekély) visszajelzést a dolog sikerességéről, illetve már ki is fogunk tudni jelentkezni!

- Felhasználók létrehozása

Hasonlóan, mint a felhasználók azonosításához, a létrehozásukhoz is van egy alapban beépített vezérlőnk, amelyet erre a feladatra találtak ki. Ez a kontroll nem más, mint a **CreateUserWizard (CUW)**. Ennél a vezérlőnél alpból ugyanazok a mezők találhatóak meg, mint a WSAT create user oldalán (név, jelszó, e-mail cím, biztonsági kérdés/válasz). Lássuk be, ennyi információ édes kevés egy társskereső portál működtetéséhez! Ezért ki fogjuk bővíteni a CUW-ot, és vele együtt az adatbázisunkat is! Barkácsolásra fel!

1. Nyissuk meg a **Registration.aspx** oldalt és a Content vezérlőn belül helyezzünk el egy CreateUserWizard kontrollt! Kezdjük el magyarosítani az alábbi tulajdonságok beállításával:


```

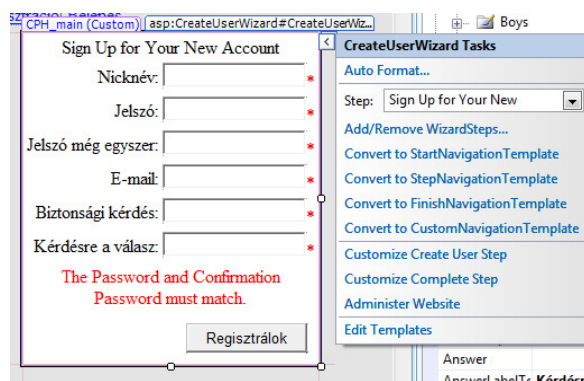
UserNameLabelText="Nicknév:"
PasswordLabelText="Jelszó:" ConfirmPasswordLabelText="Jelszó még egyszer:"
QuestionLabelText="Biztonsági kérdés:" AnswerLabelText="Kérdésre a válasz:"
CreateUserButtonText="Regisztrálok" FinishCompleteButtonText="Ok"
CompleteSuccessText="&lt;i&gt;Most már bejelentkezhetsz!&lt;/i&gt;"
FinishDestinationPageUrl="~/Default.aspx" ContinueButtonText="Tovább"

```

(Az utolsó két sorról majd csak lentebb lesz szó, de már most állítsuk be őket!)

Itt ellentétben a Login vezérlővel a fejlécsort tulajdonságon keresztül sajnos nem tudjuk átírni magyarra. (Bár megjegyzem, van egy *HeaderText* tulajdonsága, de ezzel csak a Sing Up... sor elé tudunk szöveget beszúrni!) Ezért egy másik módszerhez kell folyamodnunk, hogy a fejléctet is lefordíthassuk.

2. Jelöljük ki Design nézetben a CUW-unkat! Ilyenkor a vezérlő jobb felső sarka mellett meg fog jelenni egy nyíl, kattintsunk rá ()! Ekkor az alábbi **Smart Tag** nevezetű ablak fog megjelenni:



Kattintsunk a **Customize Create User Step** linkre!
Néhány másodperc elteltével azt láthatjuk, hogy a vezérlőnk első lapja (két lapja van a kontrollnak: egy, ahol az adatokat kell megadni, illetve egy másik, ahol a regisztráció sikerességéről tájékoztat minket) átalakult táblázatos formájúvá. Ez nekünk nagyon jó, ugyanis nem csak a fejléc szövegét tudjuk így átírni, hanem ki is tudjuk bővíteni a mezők listáját!

3. Töröljük ki a legfelső cellában lévő *Sign Up for Your New Account* szöveget, és írjuk be a helyére „**Regisztráció**”, majd jelöljük ki, és nyomjuk le a **Ctrl + B** billentyűkombinációt!

4. Most a társkeresőrendszerünk alapjául szolgáló adatok bekérését fogjuk elkészíteni. Itt a regisztrációs formon az alábbi pár adatot fogjuk bekérni: *Nem, Város, Születési idő*. Természetesen normális párkereső oldalnál, ennél több információt kell magunkról kiszolgáltatnunk, de mivel ez csak egy DEMO, ezért mi most megelégszünk ennyivel is.

5. Váltunk át Split, vagy Source nézetbe, majd keressük meg a Biztonsági kérdés-es sorhoz tartozó `<tr>` tag-et. Jelöljük ki a teljes tr-t, majd **Ctrl+C** és közvetlenül fölé másoljuk be kétszer! Az első másolt sort írjuk át ilyenre:

```
<tr>
  <td align="right">
    <asp:Label ID="lbl_city" runat="server"
AssociatedControlID="txtbox_city">Város:</asp:Label>
  </td>
  <td>
    <asp:TextBox ID="txtbox_city" runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RFV_city" runat="server"
ControlToValidate="txtbox_city" ErrorMessage="City is required."
ToolTip="City is required." ValidationGroup="CreateUserWizard1">*
</asp:RequiredFieldValidator>
  </td>
</tr>
```

6. A címke vezérlők **AssociatedControlID** tulajdonsága a képernyőolvasó programoknak segít abban, hogy az adott label, melyik beviteli mezőhöz tartozik. Ha figyelmesen néztük a CUW vezérlőnk, akkor láthattuk, hogy minden egyes TextBox mögött ott virít egy piros csillag. Ez azért van így, mert mindegyik beviteli mező össze van kötve egy úgynevezett *kötelező mezőérvényesítővel (RequiredFieldValidator (RFV))*.

7. Ez a vezérlő arra szolgál, hogy addig nem engedi tovább a felhasználót (vagyis, ha rákattint a Regisztrálok gombra, akkor nem fog történni semmi), amíg az adott mezőbe nem ír be valamilyen szöveget. Magyarul kötelezzük a felhasználót, hogy töltsen ki a formunkat, ha ezt nem teszi meg, akkor egy piros csillaggal jelezzük, hogy az adott mezőt kihagyta, emiatt nem mehet tovább.

A **ControlToValidate** tulajdonság segítségével kapcsolhatjuk össze a validátort a megfelelő TextBox-szal. A többi tulajdonság számunkra most igazándiból nem fontos, ezért azokkal nem is kell különösebben foglalkoznunk. (Az **ErrorMessage**-en, illetve a **ToolTip**-en belül esetleg átírhatjuk a *Security question*-t City-re.)

8. A születési dátum bekérésével folytassuk tovább a barkácsolást! A második másolt biztonsági kérdéses sort írjuk át az első analógiájára! Az alábbi eredményt kell, hogy kapjunk:

```

<tr>
  <td align="right">
    <asp:Label ID="lbl_born" runat="server"
AssociatedControlID="txtbox_born">Születési idő
(<i>xxxx.xx.xx</i>):</asp:Label>
  </td>
  <td>
    <asp:TextBox ID="txtbox_born" runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RFV_born" runat="server"
      ControlToValidate="txtbox_born" ErrorMessage="Born Date is
required."
      Tooltip="Born Date is required."
ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
  </td>
</tr>

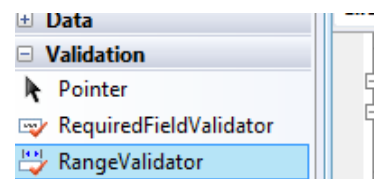
```

Fontos, hogy az TextBox vezérlők ID tulajdonságát is átírjuk, ugyanis kódból majd később hivatkozni fogunk rájuk!

9. A születési dátum kicsit eltér a városnévtől, ugyanis ez számokból áll, amiknek van egy meghatározott formátuma (xxxx.xx.xx). Mivel ez sémával leírható, ezért le is lehet ellenőriztetni, hogy megfelelő-e! A dátum helyességének az ellenőrzéséhez ismét egy *érvényesítő vezérlőt (validátort)* fogunk használni, még pedig a *tartományérvényesítőt (RangeValidator-t)*.

10. A Toolbox-unk **Validation** tab-jéről húzzunk a születési dátumhoz kapcsolt RequiredFieldValidator alá egy RangeValidator-t!

Állítsuk be a ControlToValidate tulajdonságát a *txtbox_born*-ra, a Text-et pedig *-ra! Keressük meg a properties ablakban a **Type** tulajdonságot, majd állítsuk át *String*-ről **Date**-re. Így tudni fogja a validátor, hogy dátumtípus ellenőrzést kell végrehajtani.



11. Ez idáig tök jó, de még mindig beírhat olyat a vicces kedvű felhasználó, hogy 1001.01.01.-én született. Az ilyen matuzsálemeknek már nem túl sok esélyük van egy új párkapcsolatra, ezért őket kizárjuk a portálunkról. Emellett értelemszerűen a jövőből érkezett terminátorokat se engedjük regisztrálni. Ezt a két eléggé ritka etnikai csoportot úgy tudjuk kiszűrni, ha a RangeValidator **MaximumValue**, illetve **MinimumValue** tulajdonságát is beállítjuk. Az alsó küszöbnek 1910.01.01-t adjunk meg, felső korlátnak pedig 2000.12.31-t.

12. Ha ezt mind beállítottuk a tartományérvényesítőnkön, akkor az alábbi markup kódot kell, hogy lássuk Source nézetben:

```

<asp:RangeValidator ID="RV_born" runat="server"
ErrorMessage="RangeValidator" ControlToValidate="txtbox_born"
MaximumValue="2000.12.31" MinimumValue="1910.01.01"
Type="Date">*</asp:RangeValidator>

```

13. Még egy dolgot kell beállítanunk a Születési dátumnál. A validátor vezérlők hiba esetén megjelenő szövege (jelenesetben a *-ok) alapján *statikusan* jelenik meg, vagyis ahova letettük őket, ott fognak megjeleni. Ez két érvényesítő-vezérlő esetén már nem megfelelő, ugyanis ha csak a második talál hibát, akkor a csillag egy szóköznyi távolságra fog megjeleni a TextBox-tól. Ez nem szép, ezért mind a RequiredFieldValidator, mind a RangeValidator **Display** tulajdonságát állítsuk át *Static*-ről **Dynamic**-ra!

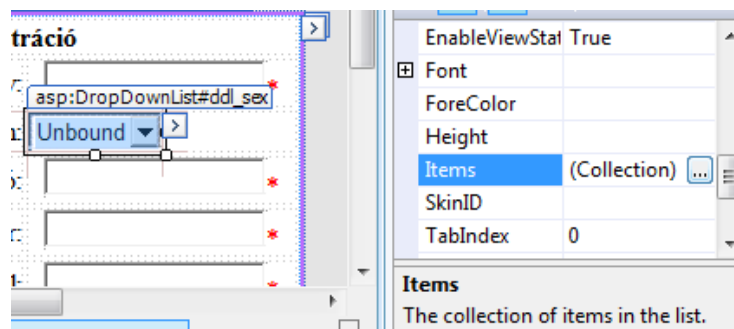
14. Jelenleg így néz ki a regisztrációs formunk:



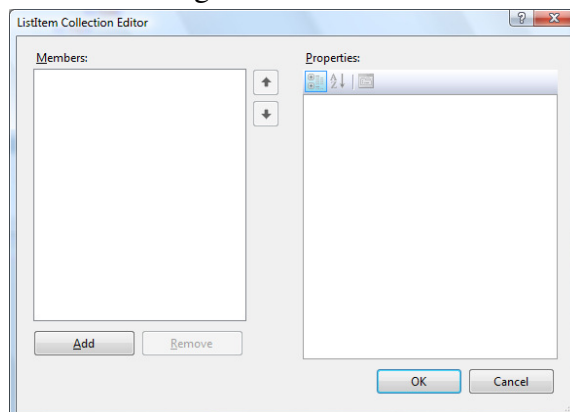
15. Már csak egyetlen egy új mezővel kell kiegészítenünk a CUW-ot, még pedig a nemmel. Az egyszerűség kedvéért kétféle nemet különböztetünk meg: a *Csajokat* és a *Pasikat*. Mivel csak e két lehetőség közül választhat a felhasználó, ezért egy *legördülő-lista* (**DropDownList** (DDL)) segítségével megkíméljük őket a gépeléstől.

16. Készítsünk egy másolatot a Nicknév sorról. Töröljük ki a másolt sor második cellájából a TextBox-ot, illetve az RFV-t, és a helyükre húzzunk be a Toolbox **Standard** Tab-jéről egy DropDownList-et. Az első cellában a *Nicknév* szót írjuk át „**Nem:**”-re!

17. A DDL-nek azt az azonosítót adjuk, hogy *ddl_sex*! Keressük meg az **Items** tulajdonságát, majd kattintsunk bele a mellette lévő (Collection) szövegbe, ekkor meg fog jelenni egy ... gomb.



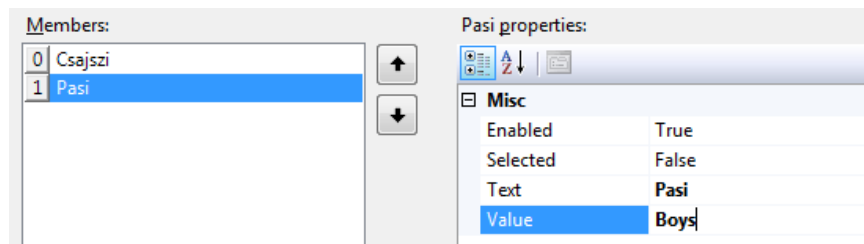
18. Kattintsunk rá, majd egy **ListItem Collection Editor** nevezetű ablak fog felpattanni! Itt klikkeljünk kétszer az alul lévő *Add* gombra.



Ekkor a **Members** listába bekerül két **ListItem**. Válasszuk ki a nulladikat, majd jobb oldalt a **ListItem Properties**-nél állítsuk a **Text**-et „*Csajszí*”-ra, a **Value**-t pedig „*Girls*”-re. (A **Text**

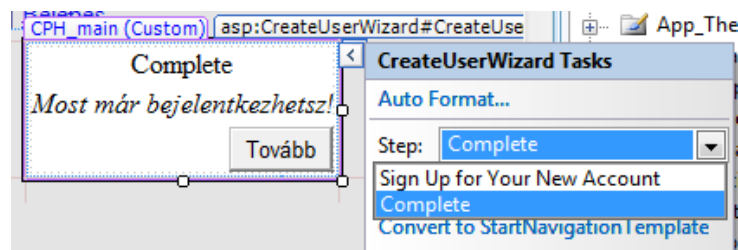
tulajdonság szövege fog majd megjeleni a DDL-ben, de mi a Value értékekkel fogunk majd dolgozni kódból! /téma beállításánál/).

A másik listaelem Text-jét állítsuk „Pasi”-ra, a Value-ját pedig „Boys”-ra. Végül kattintsunk alul az OK gombra!



19. Az egységes megjelenés érdekében vegyük a DDL szélességét 155px-re, és az öt tartalmazó cella szövegigazítás attribútumát állítsuk be balrára! (<td align="left">). Ezzel a regisztrációs űrlapunk első oldala el is készült.

20. A második oldalon összesen csak a fejléct szeretnénk magyarosítani! (A gomb szövegét a **ContinueButtonText**-tel, a fölötte lévő hablatyot pedig a **CompleteSuccessText**-tel már beállítottuk!) Jelöljük ki megint a teljes CUW-ot, majd a Smart Tag-en belül a **Step** legördülő-listából válasszuk a **Complete**-t!



21. Ismét kattintsunk a smart tag-et lenyitó nyílra, és most válasszuk a **Customize Complete Step**. Hasonlóan az első oldalnál itt is egy táblázatos forma generálódik a **CompleteWizardStep ContentTemplate**-jén belül. Írjuk át a *Complete* szót arra, hogy „**Sikerés regisztráció**”, majd vegyük félkövérre.

A CUW-nál még a legelején beállítottuk a **FinishDestinationPageUrl** tulajdonságot, amely összesen annyit állít be, hogy melyik oldalra irányítsuk át a felhasználót akkor, amikor rákattint a Tovább gombra. Jelenesetben a Default.aspx oldalra lesznek majd átirányítva!

22. Utolsó lépésként állítsuk vissza a Smart tag-ben az aktív oldalt a **Sign Up for Your New Account** lapra. Ha ezt elfelejtenénk, akkor a Sikerés regisztrációs oldal fog alpból megjeleni a Registration oldalon, ami valljuk be, nem a legszerencsésebb!

El is készültük a regisztrációs oldal felhasználói felületével! Az így nyert új adatokat majd még el kell tárolnunk az adatbázisban, de ez már a jövő zenéje. Ezért most így félkészen hagyjuk abba a fejlesztést, vegyünk egy nagy levegőt, majd nyissuk meg a tutorial második részét tartalmazó dokumentumot! ☺