

SELÇUK ÖZDEMİR

# ASP.NET 2.0



**netsoft**  
Bilgisayar Sistemleri Tic. Ltd.Şti.

KONU 1

ASP.NET

## ASP.NET NEDİR?

### .NET FRAMEWORK NEDİR?

.Net Framework .Net platformunun bir dayanak noktasıdır. .Net Framework gücünü içerisinde barındırdığı yüzlerce sınıf (class) kütüphanesinden alır. Bize sunucu tarafında çalışan bir yapı kurabilmemizi sağlar, olaya dayalı programlama modeli sayesinde daha hızlı ve verimli uygulamalar geliştirebilmemizi ve bunları istediğimiz programlama diliyle yazabilmemizi sağlar.

.Net'in destek verdiği diller

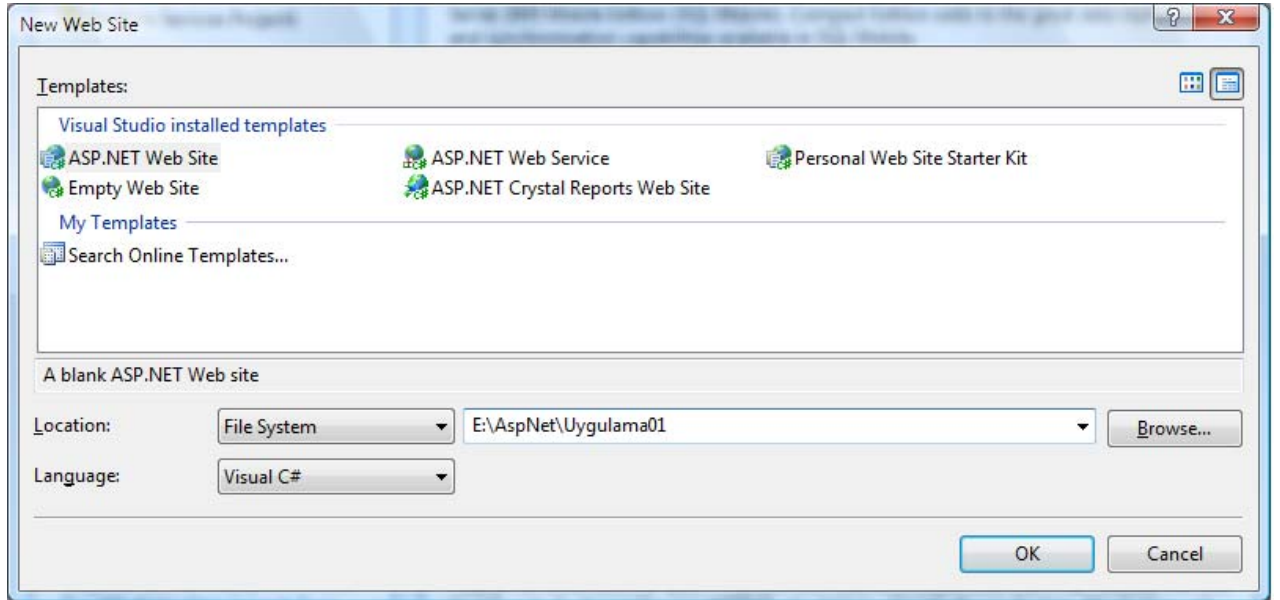
|                        |        |                          |
|------------------------|--------|--------------------------|
| Visual Basic .Net      | C#     | Microsoft Visual J# .NET |
| Microsoft Jscript .Net | Oberon | Scheme                   |
| Perl                   | Pyhton | Cobol                    |
| Haskell                | Pascal | ML                       |
| Mercury                | Eiffel | Ada                      |
| SmallTalk              | Caml   | Microsoft Visual C++     |
| APL                    | C      | Oz                       |

Çoklu dil desteğinin avantajları

- ❖ Kod modülleri yeniden kullanılabilir. Bir kod modülü, bir dil içinde yazılıp farklı diller içinde kullanılabilir.
- ❖ Tüm diller için object kütüphanesi aynıdır ve aynı object modeli kullanılır.
- ❖ Eski dillerde yazılmış programlar güncellenebilir.
- ❖ .NET Framework içindeki tüm diller eşit performansa sahiptir. Bütün .NET tabanlı diller Microsoft Intermediate Language (MSIL)'a derlenir. Tüm MSIL ler native kod'a derlenir. Bu run-time'dır ve hepsi aynı derleyici kullanır.

### ASP.NET WEB UYGULAMALARI

Bir web projesi açmak için VS.NET 2005'i açtıktan sonra üst menüden File > New > Web Site seçeneğinin seçtikten sonra karşımıza web site açma penceresi gelecektir. Açılan pencerede Templates kısmından ASP.NET Web Site seçeneğini seçin. Language seçeneğinin C# olmasına dikkat edin. Location kısmından File System'i oradanda nereye kaydedecekseniz konumu belirtiniz ve OK düğmesine basınız. (Şekil 1.1)

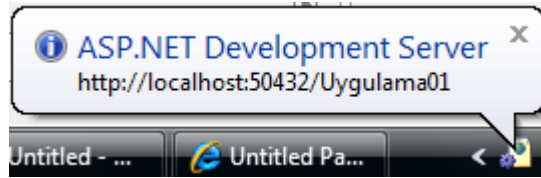


Şekil 1.1

Artık ASP.Net sayfamız açılmış olacaktır. Hiçbirşey yazmadan klavyeden F5 tuşuna basınız. Karşımıza bir soru gelirse OK dememiz yeterli.

### Visual Web Developer Web Server Nedir?

Biraz önceki uygulamamızda F5 e bastıktan sonra taskbarımızın üstünde bir baloncuk çıktı (Şekil 1.2) . Bu ifade web uygulamalarını geliştirmek için kullanılan local Web Server olup, IIS yapısının extra özelliklerini içermez. Önceleri IIS de ayarlama sorunları yaşadığımız olmuştur. .NET 2005 platformu kendi IIS ini çalıştırdığından IIS bazındaki sorunlar ortadan kalkmıştır.

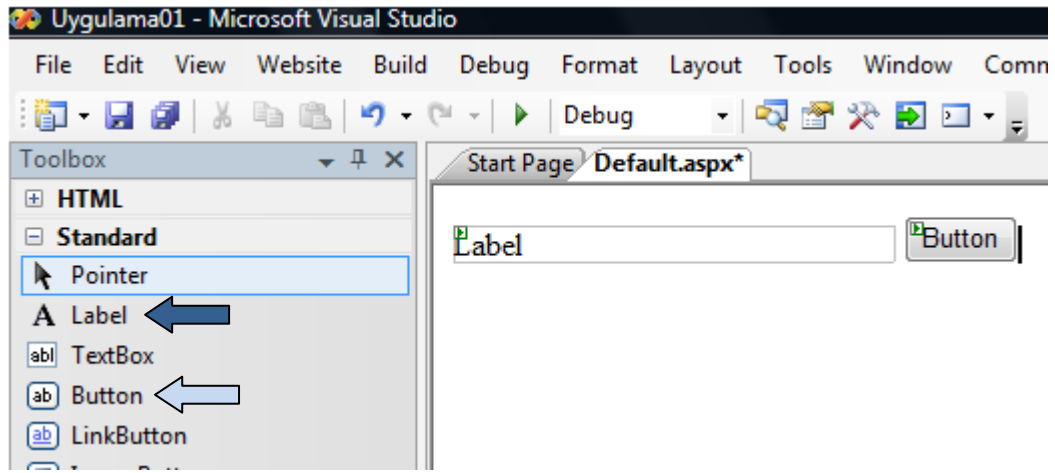


Şekil 1.2

Evet artık ASP.Net uygulamalarımızı geliştirebiliriz. Bunu için açtığınız projenin çalışması durduğundan emin olun. Çalışma durumunu durdurmak için açılan internet explorer penceresini kapatmak yeterli olacaktır.

### ASP.NET NESNELERİ

Sayfamıza Toolbox penceresinden Design kısmında iken sayfamıza bir label ve buton ekleyelim. (Şekil 1.3). Ekledikten sonra butonumuzun üstünü çift tıklayalım.

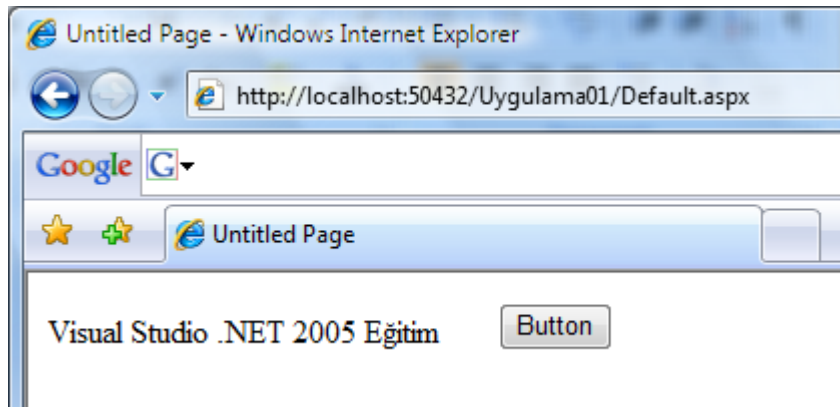


Şekil 1.3

Butonumuzu çift tıkladıktan sonra Button1\_Click olayına kodumuzu yazalım.

```
protected void Button1_Click(object sender, EventArgs e)
{
    Label1.Text="Visual Studio .NET 2005 Eğitim";
}
```

Kodumuz yazıp F5(çalıştır)'e batıktan sonra karşımıza gelen web sayfasından butonu tıklayalım. (Şekil 1.4)



Şekil 1.4

Böylece ASP.Net uygulamalarımızda ilk örneğimizi yapmış olduk. (Tabii ki ASP.Net bu kadar basit değil ☺)...

# KONU 2

## STANDART KONTROLLER

## Server Kontrol Nedir?

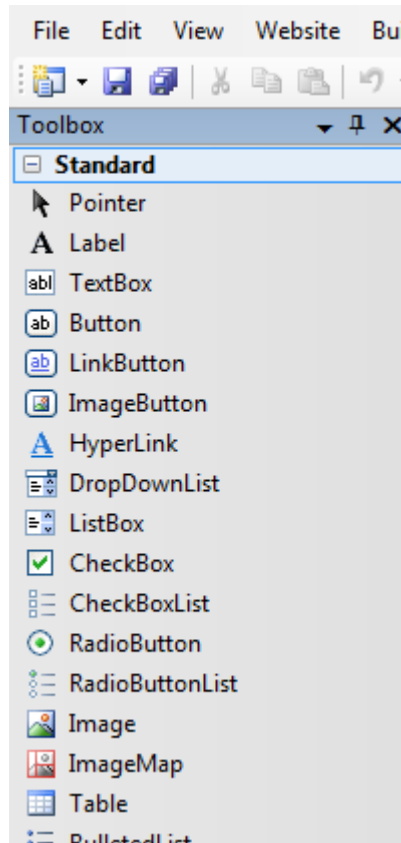
Kullanıcı arabirimi veya diğer ilişkisel fonksiyonlar oluşturmak üzere Server üzerinde çalışan componentlere denir. Server kontroller **runat= "server"** özelliğine sahiptirler. (Şekil 2.1)

```
<form id="form1" runat="server">
<div>
    <asp:Label ID="Label1" runat="server" Text="Label" Width="221px"
    <asp:Button ID="Button1" runat="server" OnClick="Button1_Click"
</div>
</form>
```

Şekil 2.1

## Standart Kontroller

Bu bölümde ASP.Net platformumuzu açtığımızda **Toolbox** penceresindeki **Standart** bölümündeki kontrollerden bahsedeceğiz.(Şekil 2.2)



Şekil 2.2

### ❖ LABEL

Web sayfasında kişilere mesaj iletmek için kullanılır.(Şekil 2.3) Properties penceresinden bazı özelliklerinin açıklaması ise;

| Özellik Adı | Tanımı                        |
|-------------|-------------------------------|
| BackColor   | Taban Rengi                   |
| BorderColor | Çerçeve Rengi                 |
| BorderStyle | Çerçeve Tipi                  |
| BorderWidth | Çerçeve Kalınlığı             |
| CssClass    | Sitil Atama                   |
| Font        | Yazı karakteri ayarları       |
| ForeColor   | Yazı Rengi                    |
| Visible     | False olursa ekranda görünmez |
| Width       | Genişlik Değeri               |

A Label

Şekil 2.3



Bazı özellikler diğer kontroller içinde aynen geçerlidir. Bu yüzden benzer özellikleri olan kontrollerde tekrardan yazmayacağım.

#### ❖ TEXTBOX

Kullanıcıların bilgi girişi için kullanılan kontroldür.(Şekil 2.4) Özellikle üyelik kayıt, şifre doğrulamalarda yaygın olarak kullanılır.

abl TextBox

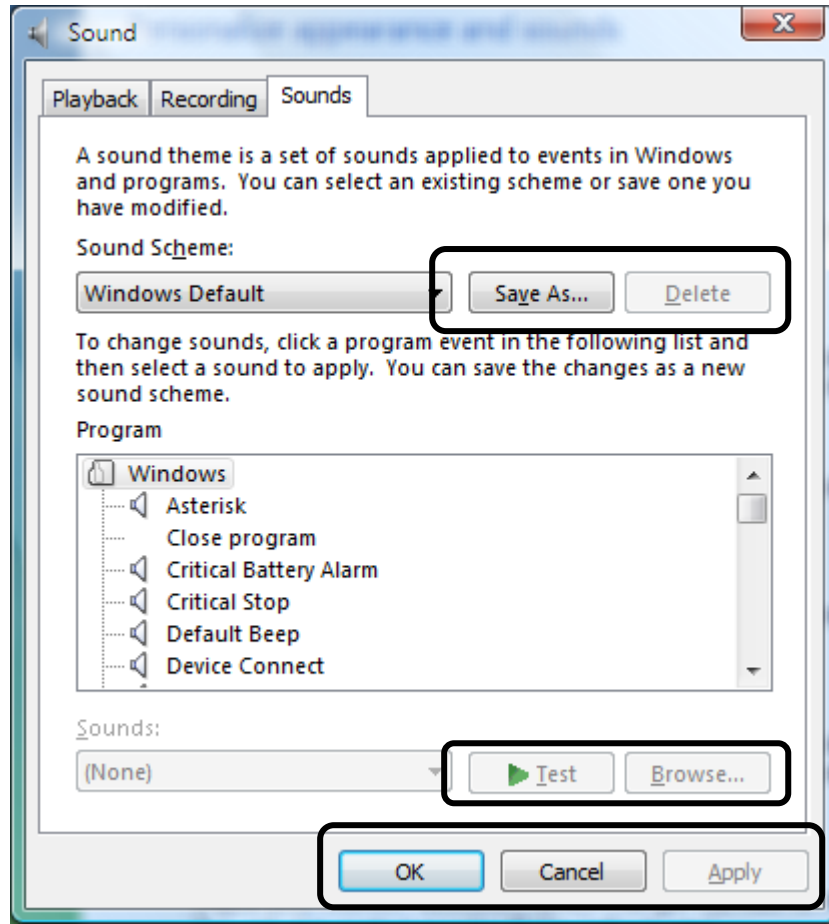
Şekil 2.4

| Özellik Adı | Tanımı   |
|-------------|--|
| Text        | İçerisine Değer Atama  |
| TextMode    | Bir / Birden Çok Satır Ayarı                                     |
| MaxLenght   | Maksimum Girilecek Karakter Sayısı                               |
| TextMode    | Single(Tek Satır), Multiple (Çok Satır), Password (Şifreli yazı) |
| Wrap        | False Olduğunda Alt Taraf Kaydırma Çubuğu                        |
| Enable      | False Olduğunda Bilgi Girişi Yapılamaz                           |

#### ❖ BUTTON

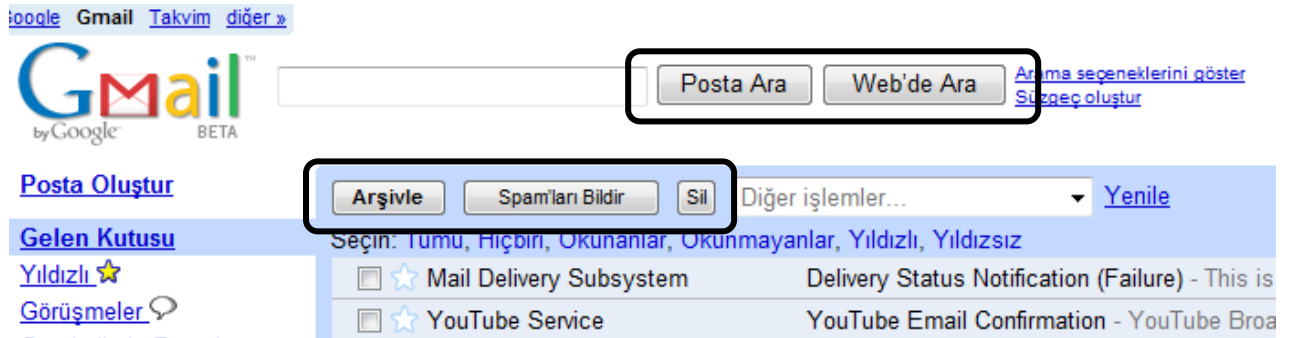
Tüm windows uygulamalarında Tamam, Uygula, İptal vb. olarak gördüğümüz basma (click) özelliğine sahip kontroldür. (Şekil 2.5)





Şekil 2.5

Tabiki konumuz ASP.Net olduğu için bu butonu biz web sayfalarımızda bir işlemi onaylatmak, form göndermek, kayıt eklemek, giriş yapmak gibi işlemlerde kullanacağız. (Şekil 2.6)



Şekil 2.6

Butonumuzu eklemek için Toolboxdaki Button ismi projeye sürüklememiz yeterli (Şekil 2.7)



Şekil 2.7

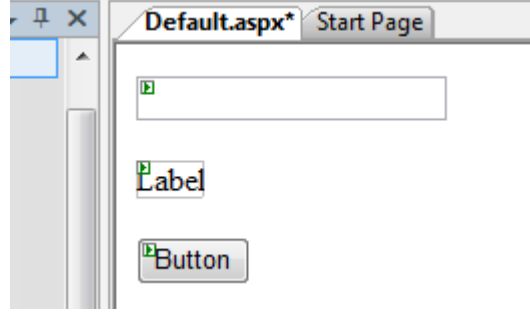
| Özellik Adı | Tanımı                              |
|-------------|-------------------------------------|
| Text        | Butonun Üstündeki Yazıyı Değiştirir |
| Enabled     | False Olduğunda Buton Pasif Olur.   |



Kontrollerin tüm özellikleri kod ortamında yazarakta yönetebilirsiniz.  
Örn: **TextBox1.Enabled=True;**

#### UYGULAMA 1

- 1) File>New>Web Site seçeneğini seçerek yeni bir web sitesi oluşturalım
- 2) Web sitemize birer adet button, textbox ve label yerleştirelim. (Şekil 2.8)



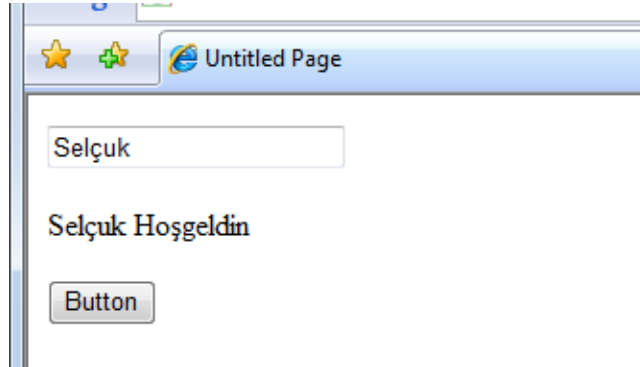
Şekil 2.8

- 3) Butonumuzun üstünü çift tıklayarak Button1\_Click özelliğine aşağıdaki kodu yazalım.

```
protected void Button1_Click(object sender, EventArgs e)
{
    this.Label1.Text = this.TextBox1.Text + " Hoşgeldin";

    //Label1 in yazını textbox a yazılan yazıyı yaz.
}
```

- 4) Kodumuzu yazdıktan sonra çalıştıralım. Textbox'a bir isim yazdıktan sonra butonu tıklatalım. (Şekil 2.9)



Şekil 2.9

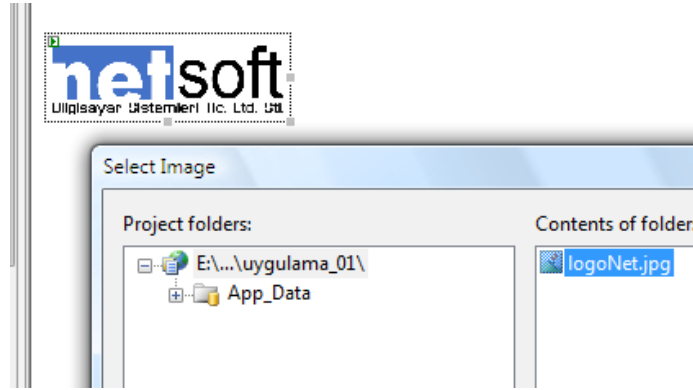
#### ❖ LINK BUTTON

Bir çok özelliği butona benzeyen Link buttonu link vermek için kullanır. Click özelliğine kodumuzu yazarak yönlendirme yapabiliriz.



#### ❖ IMAGE BUTTON

LinkButton ile işlevi aynıdır. Tek farkı Link yazısının yerine resim gelir. Resim getirmek için Properties penceresinden ImageURL seçeneğinden resmi belirleyebilirsiniz. (Şekil 2.10)



Şekil 2.10

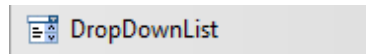
Image Button'a resim ekledikten sonra bir sayfaya yönlendirmek için ImageButton'u çift tıklayınız. Click özelliğine aşağıdaki kodu yazınız.

```
protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect("http:\\\\www.netsoft.com.tr");
}
```

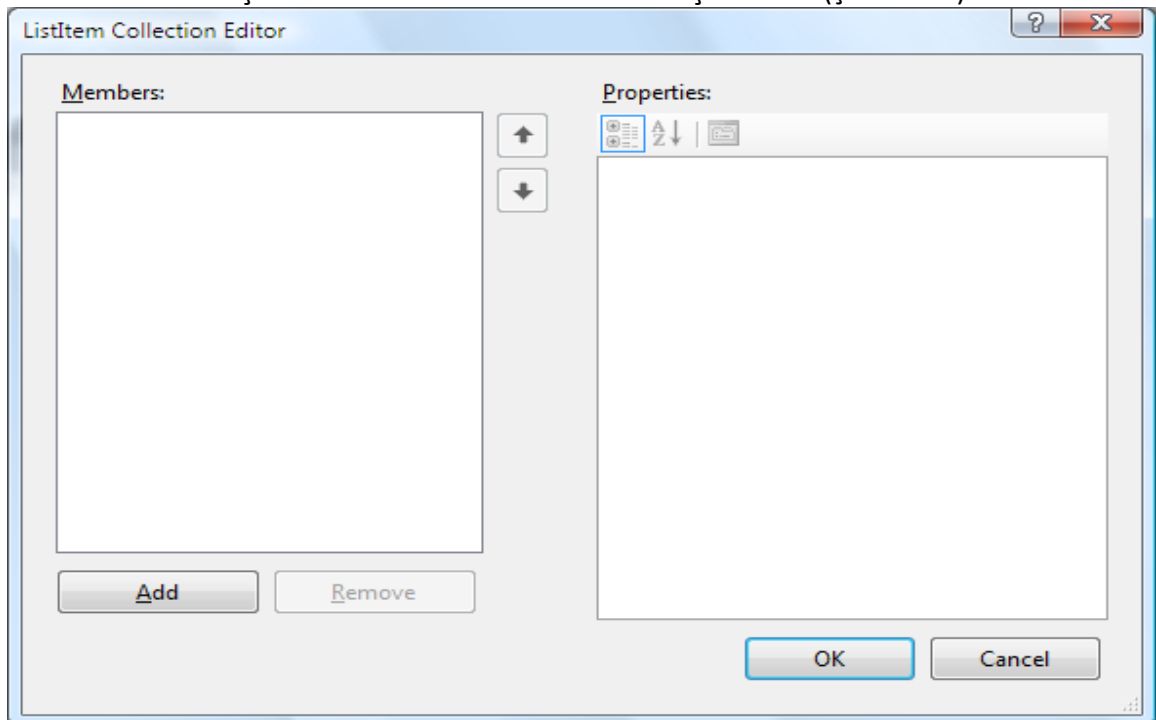
Kodumuzu yazıp çalıştırdan sonra resmi tıkladığımızda web sayfası açılacaktır.

#### ❖ DROPDOWNLIST

Açılan liste şeklinde seçim yapmak. Birden fazla seçeneğin sadece bir tanesinin seçilmesi durumunda kullanılır. Örneğin; Web sitelerin “Şehir” seçimleri gibi...

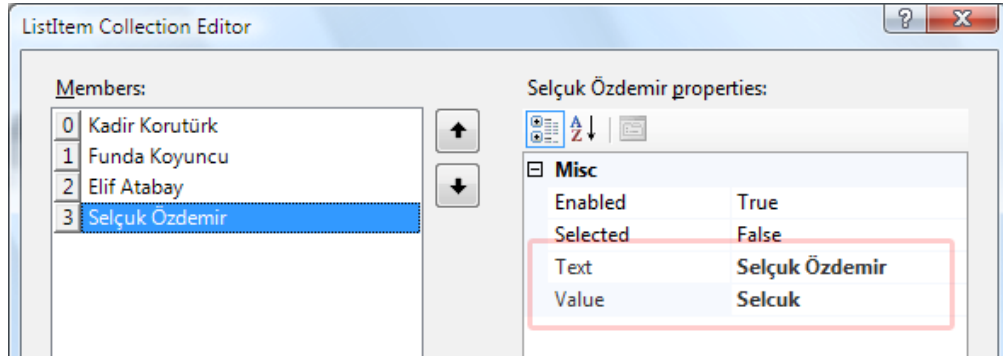


- 1) Yeni bir Web Sayfası açalım.
- 2) Toolboxdan DropDownList ekleyelim.
- 3) DropDownList'i seçtikten sonra Properties penceresinden Items seçeneğini tıklayınız. Karşımıza ListItems Collection Editor açılacaktır. (Şekil 2.11)



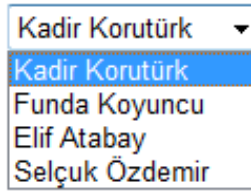
Şekil 2.11

- 4) Add butonuna basarak DropDownList için elemanlar ekleyelim. Add dedikten sonra Text ve Value değerlerini dolduralım.



Şekil 2.12

- 5) Text özelliğine yazdığımız değerler DropDownList de ekranda gorunecektir.(Şekil 2.13)



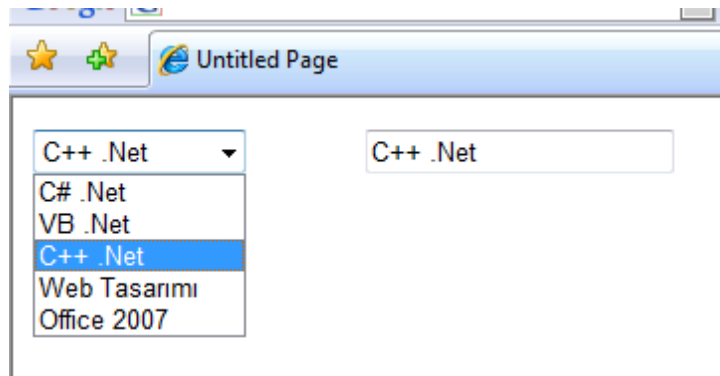
Şekil 2.13

## UYGULAMA 2

- 1) Yeni bir Web sayfası açalım. Web sayfamıza TextBox ve DropDownList koyalım. (Şekil 2.14)
- 2) DropDownList'e değerlerimizi ekleyelim.
- 3) Daha sonra DropDownList'i çift tıklayarak code-behind alanına geçiniz.
- 4) DropDownList'in SelectedIndexChanged olayına aşağıdaki kodumuzu yazalım.

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    this.TextBox1.Text = this.DropDownList1.SelectedItem.ToString();
}
```

Yukarıdaki kodu çalıştırdığımızda olması gereken durum DropDownList seçeneklerinden birini seçtiğimizde o değerin TextBox da görünmesi sağlanacaktır.



Şekil 2.14



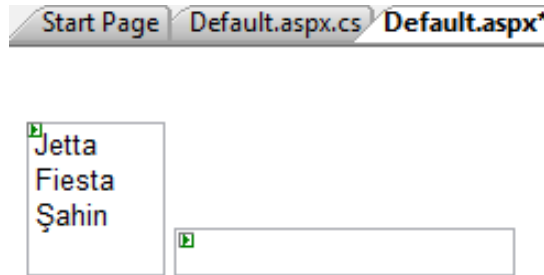
Kodu çalıştırdığımızda, DropDownList'den seçmemize rağmen TextBox'a değer **aktarılmayacaktır**. ASP.Net sayfalarında bu özelliğin algılanması için, DropDownList1'in properties penceresinde bulunan **AutoPostBack** özelliğinin **True** olması gerekmektedir.

#### ❖ LISTBOX

Tüm özellikleri DropDownList ile aynıdır. Tek fark karşımıza direk olarak liste şeklinde gelmesidir. Kodlama olarakda DropDownList de geçerli olan tüm kodlar ListBox içinde geçerlidir.

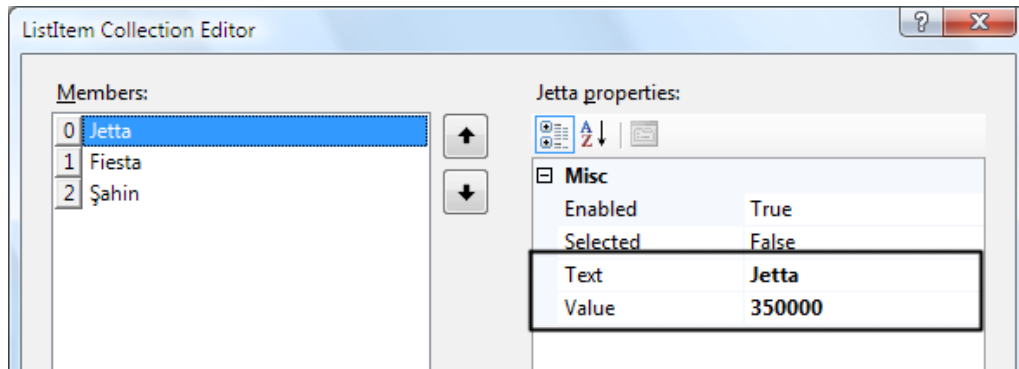
### UYGULAMA 1

1) Yeni bir Web sayfası açalım. Web sayfamıza TextBox ve ListBox koyalım. (Şekil 2.15)



Şekil 2.15

2) ListBox'ımıza Properties penceresinden Items seçeneğini kullanarak değerlerimizi ekleyelim. (Şekil 2.16)

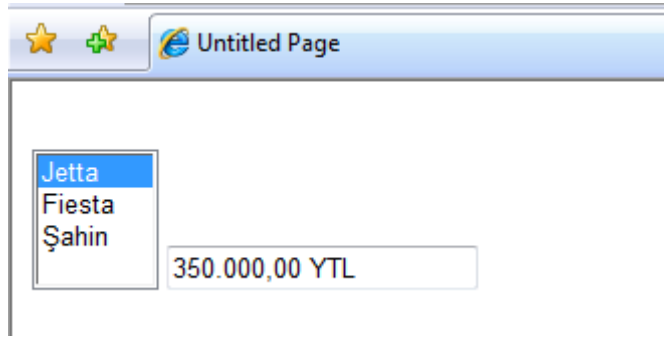


Şekil 2.16

3) Değerlerimizi girdikten sonra ListBox'ımızı çift tıklayarak kod alanımıza geçelim. Aşağıdaki kodumuzu yazarak çalıştıralım (Şekil 2.17). Tabi bu arada Listbox'ımızı seçerek AutoPostBack seçeneğini True yapmayıda unutmayın.

```
protected void ListBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    int deger = Convert.ToInt32(this.ListBox1.SelectedItem.Value);
    this.TextBox1.Text = string.Format("{0:N} YTL", deger);
}
```

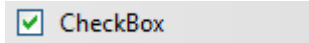
Buradaki **string.Format("{0:N} YTL",deger)** kısmı gönderdiğimiz değeri araç olarak ayarlayacaktır.



Şekil 2.17

## ❖ CHECKBOX

Genel kullanımı bir formda birden fazla seçim yapmak için kullanılır (Şekil 2.18). Aşağıdaki örneğimizde seçtiğimiz dersleri bir textboxa ekleyelim.



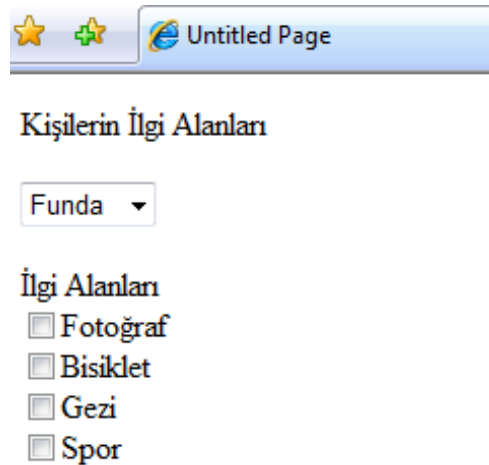
Şekil 2.18

CheckBox nesnemizin ait özellikler aşağıdaki listede yer almaktadır.

| Özellik Adı | Tanımı                   |
|-------------|--------------------------|
| Checked     | İşaretili Olup Olmaması  |
| Text        | Ekranda Görünecek Yazısı |
| Enabled     | Aktif Olup Olmaması      |
| BorderWidth | Çerçeve Kalınlığı        |

## UYGULAMA 1

1. Web Sayfamıza 4 CheckBox ve bir adet DropDownList ekleyelim(Şekil 2.19). CheckBox larımızın properties penceresinden Text özelliğini kullanarak değerleriniz değiştirilim.

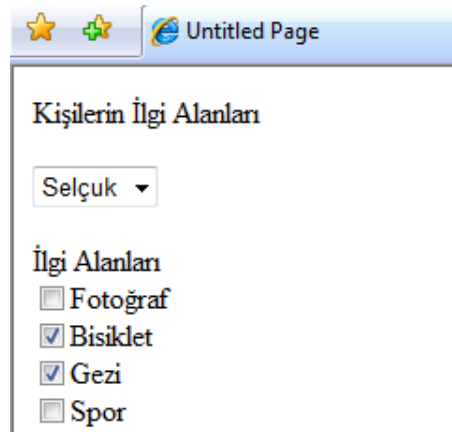


Şekil 2.19

2. DropDownList'e items özelliğinden üç isim girelim (Örn: Funda,Selçuk,Ahmet) ve AutoPostBack özelliğini True yapalım.
3. Daha sonra DropDownList'i çift tıklayarak kod alanına geçelim. Aşağıdaki kodu ekleyelim.

```
protected void DropDownList2_SelectedIndexChanged(object sender, EventArgs e)
{
    if (this.DropDownList2.SelectedIndex == 0)
    {
        this.CheckBox1.Checked = true;
        this.CheckBox2.Checked = false;
        this.CheckBox3.Checked = false;
        this.CheckBox4.Checked = true;
    }
    else if (this.DropDownList2.SelectedIndex == 1)
    {
        this.CheckBox1.Checked = false;
        this.CheckBox2.Checked = true;
        this.CheckBox3.Checked = true;
        this.CheckBox4.Checked = false;
    }
    else
    {
        this.CheckBox1.Checked = true;
        this.CheckBox2.Checked = true;
        this.CheckBox3.Checked = true;
        this.CheckBox4.Checked = true;
    }
}
```

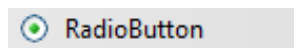
Kodu çalıştırdığımızda DropDownList'den bir isim seçelim. Seçtiğimiz ismin index numarasına göre CheckBox'lar işaretlenecektir (Şekil 2.20).



Şekil 2.20

#### ❖ RADIOBUTTON

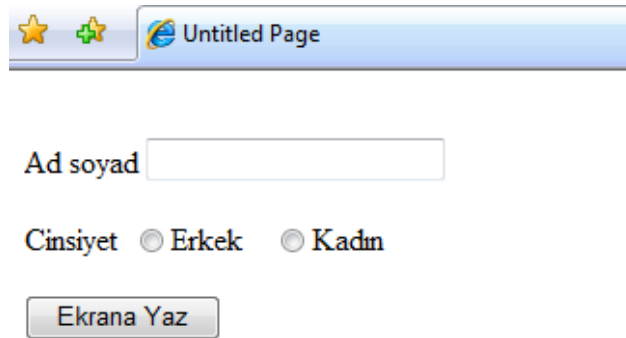
CheckBox'a benzer, tek farklılık bir grup ismi verilen radiobuttonlardan sadece bir tanesini seçme durumumuz olacaktır (Şekil 2.21).



Şekil 2.21

#### UYGULAMA 1

1) İlk örneğimizde açmış olduğumuz web sayfasına 2 RadioButton, birer adette TextBox ve Button Ekleyelim. RadioButtonların text özelliğinden Erkek – Kadın olarak değiştirelim. (Şekil 2.22).



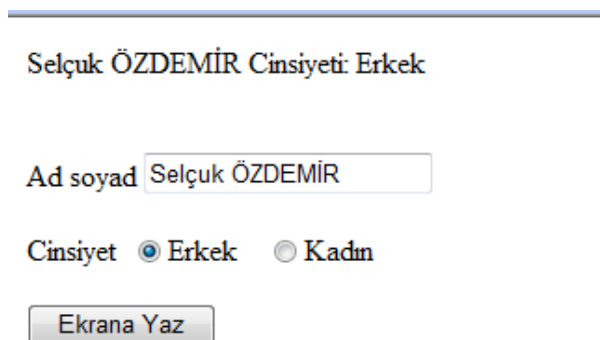
Şekil 2.22

2) RadioButtonlarımızı tıklayarak Properties penceresinden GroupName kısımlarına cinsiyet yazalım. Böylece bu gruba adında iki üye olacağından sadece bir tanesi işaretlenecektir.

3) Butonumuzu çift tıklayarak aşağıdaki kodumuzu yazalım.

```
protected void Button1_Click(object sender, EventArgs e)
{
    string yazi = this.TextBox1.Text;
    if (this.RadioButton1.Checked == true)
    {
        Response.Write(yazi + " Cinsiyeti: Erkek");
    }
    else
    {
        Response.Write(yazi + " Cinsiyeti: Kadın");
    }
}
```

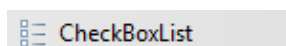
Burada Response.Write komutumuz ekrana yazmak için kullanılır. RadioButton1 eğer işaretlendi ise TextBoxdaki ismi alarak cinsiyetini ekrana yazacaktır (Şekil 2.23).



Şekil 2.23

#### ❖ CHECKBOXLIST

CheckBox'a benzer bir yapıya sahiptir. Listbox ile CheckBox'ın birleşmeside denilebilir (Şekil 2.24).



Şekil 2.24

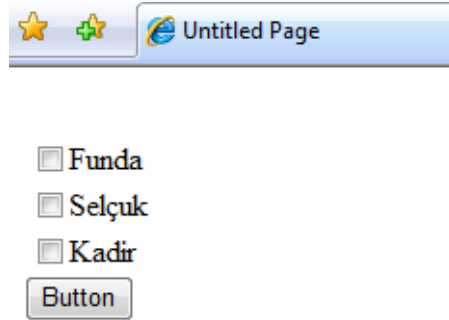
CheckBoxlist'e ait özellikler aşağıdaki listede yer almıştır.



| Özellik Adı     | Tanımı                        |
|-----------------|-------------------------------|
| Items           | CheckBox Ekleme Penceresi     |
| RepeatDirection | Yatay/Dikey Konumlandırma     |
| TextAlign       | Yazı Hizalama / Konumlandırma |

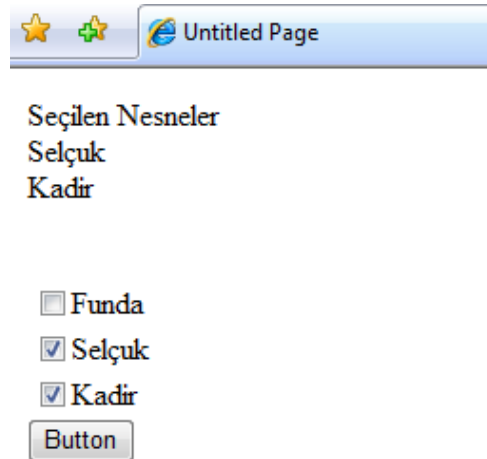
## UYGULAMA 1

- 1) Web sayfamıza CheckBoxlist ekleyerek Items özelliğinden sırasıyla, Funda,Selçuk,Kadir isimlerini ekleyelim.
- 2) Kodumuzu çalıştırmak içinde bir Button ekleyelim (Şekil 2.25). Butonumuzu çift tıklayarak kod alanına geçelim.



Şekil 2.25

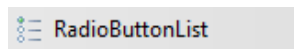
- 3) Aşağıdaki kodu ekleyerek çalıştıralım. CheckBoxlist'ten isimleri seçtikten sonra butona bastığımızda seçtiğimiz isimleri ekranda görebiliriz (Şekil 2.26).



Şekil 2.26

## ❖ RADIOBUTTONLIST

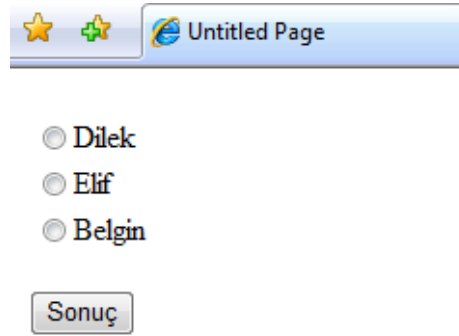
RadioButton ile ListBoxın birleşmiş hali diyebiliriz (Şekil 2.27). Items özelliğinden seçenek ekleyebiliriz. Bu RadioButtonList listesinden sadece bir tane seçebiliriz.



Şekil 2.27

## UYGULAMA 1

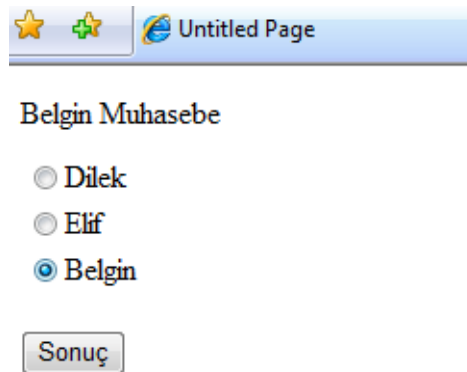
- 1) Web sayfamıza RadioButtonList ekleyerek Items özelliğinden sırasıyla, Dilek, Elif, Belgin isimlerini yazınız.
- 2) Daha sonra bir buton ekleyerek butonumuzun çift tıklayarak kod alanına geçiniz (Şekil 2.28).



Şekil 2.28

3) Kod alanına geçtikten sonra aşağıdaki kodu yazdıktan sonra çalıştırınız. Radio buttonlarından birini seçtikten sonra butona basınız (Şekil 2.29) .

```
protected void Button1_Click(object sender, EventArgs e)
{
    if (this.RadioButtonList1.SelectedIndex == 0)
    {
        Response.Write(this.RadioButtonList1.SelectedValue + " Lisans Satış");
    }
    else if (this.RadioButtonList1.SelectedIndex == 1)
    {
        Response.Write(this.RadioButtonList1.SelectedValue + " Satış Müdürü");
    }
    else
    {
        Response.Write(this.RadioButtonList1.SelectedValue + " Muhasebe");
    }
}
```

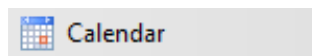


Şekil 2.29

Böylece ListBoxdaki gibi index numarasından seçtiğimiz RadioButton algılanarak, ekrana bilgiler yazılmıştır.

#### ❖ CALENDAR

Ekrana bir takvim getirerek tarih bilgilerini almamızı sağlar. Ayrıca kullanıcıların tarih formatlarında hata yapmalarını engellemek amaçlı kullanılır (Şekil 2.30).

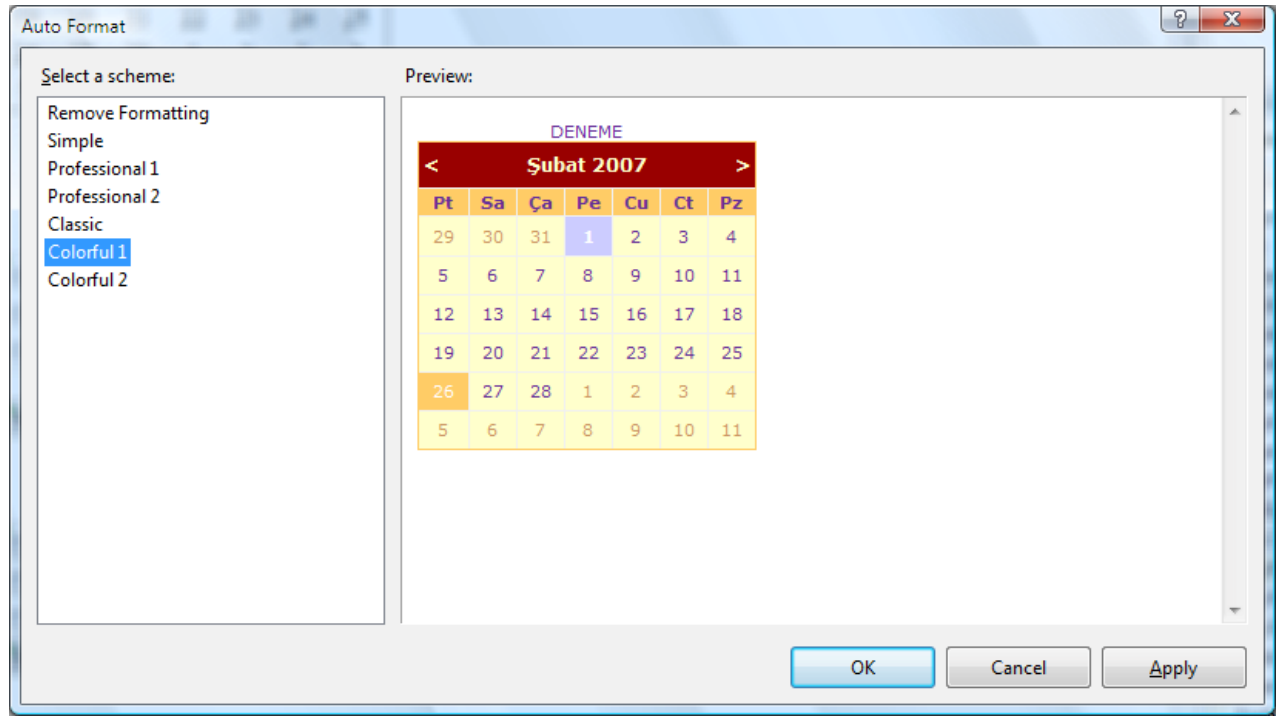


Şekil 2.30

Calendar nesnemizin özellikleri aşağıdaki listede yer almaktadır.

| Özellik Adı       | Tanımı                             |
|-------------------|------------------------------------|
| SelectedDate      | Belirlediğiniz Gün Seçili Gelir    |
| SelectedDateSytle | Seçilen Günü Renk / Biçim Ayarları |
| Caption           | Takvimin Başlık Yazısı             |

Calendar nesnemin üstünden mouse ile sağ tuşa bastığınızda gelen menüden Auto Format seçeneğini seçiniz (Şekil 2.31). Karşımıza gelen pencereden bize sunulmuş olan biçimlerden birini seçerek OK düğmesine basınız.



Şekil 2.31

#### UYGULAMA 1

- 1) Calendar nesnemizde seçtiğimiz bir tarihi TextBox nesnemize aktarmak için, web sayfamıza birer adet Calendar ve TextBox nesnelerimizi ekleyelim.
- 2) Daha sonra Calendar nesnemizi çift tıklayarak kod alanına geçelim ve kodumuzu yazalım.

```
protected void Calendar1_SelectionChanged(object sender, EventArgs e)
{
    this.TextBox5.Text = this.Calendar1.SelectedDate.ToShortDateString();
}
```

Kodumuz yazıp çalıştırdıktan sonra Calendar nesnemizden seçtiğimiz tarihi TextBox'a aktaracaktır (Şekil 2.32).

| Şubat 2007 |    |    |    |    |    |    |
|------------|----|----|----|----|----|----|
| Pt         | Sa | Ça | Pe | Cu | Ct | Pz |
| 29         | 30 | 31 | 1  | 2  | 3  | 4  |
| 5          | 6  | 7  | 8  | 9  | 10 | 11 |
| 12         | 13 | 14 | 15 | 16 | 17 | 18 |
| 19         | 20 | 21 | 22 | 23 | 24 | 25 |
| 26         | 27 | 28 | 1  | 2  | 3  | 4  |
| 5          | 6  | 7  | 8  | 9  | 10 | 11 |

23.02.2007

Şekil 2.32

## UYGULAMA 2

- 1) Bu uygulamamızda ise Çarşamba günlerinde toplantılarımız var ve bunu takvimimize ekleyelim.
- 2) Bunu için Calendar nesnemizin Calendar1\_DayRender event'ına aşağıdaki kodu yazalım.

```
protected void Calendar1_DayRender(object sender, DayRenderEventArgs e)
{
    if (e.Day.Date.DayOfWeek == DayOfWeek.Wednesday)
    {
        e.Cell.Controls.Add(new LiteralControl("Toplantı"));
    }
}
```

Kodu yazıp çalıştırdığımızda Çarşamba günlerinde Toplantı yazısının eklendiğini görebiliriz (Şekil 2.33).

| Şubat 2007 |    |            |    |    |    |    |
|------------|----|------------|----|----|----|----|
| Pt         | Sa | Ça         | Pe | Cu | Ct | Pz |
| 29         | 30 | 31Toplantı | 1  | 2  | 3  | 4  |
| 5          | 6  | 7Toplantı  | 8  | 9  | 10 | 11 |
| 12         | 13 | 14Toplantı | 15 | 16 | 17 | 18 |
| 19         | 20 | 21Toplantı | 22 | 23 | 24 | 25 |
| 26         | 27 | 28Toplantı | 1  | 2  | 3  | 4  |
| 5          | 6  | 7Toplantı  | 8  | 9  | 10 | 11 |

Şekil 2.33

## ❖ MULTIVIEW - VIEW

Kullanımı tab menüye benzeyen, oldukça kullanışlı olan Multiview ve View nesneleri beğeneceğinizi düşünüyorum (Resim 2.34).

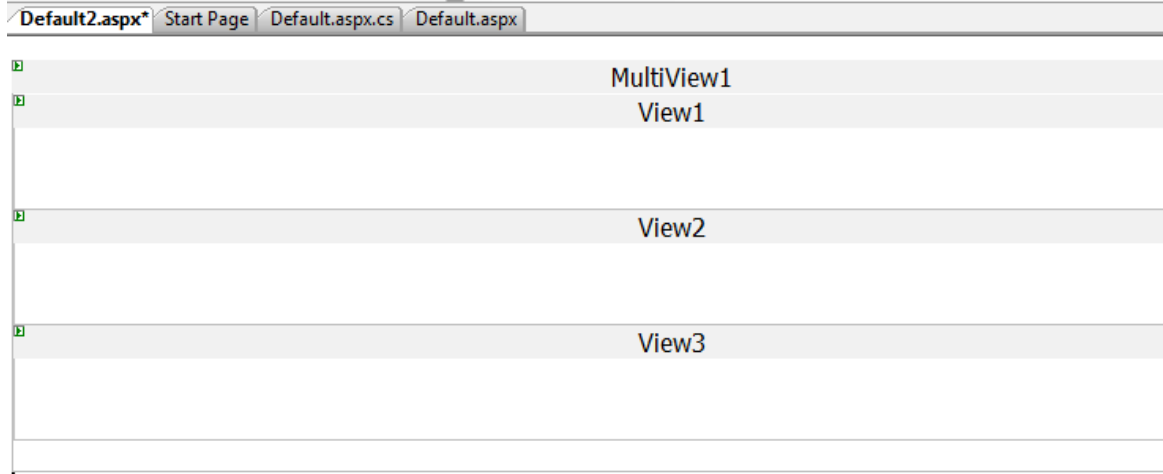


Resim 2.34

Multiview bilmemiz gerek bir ActiveViewIndex özelliği, hangi View nesnesinin görünmesini istiyorsak onun index değerin yazmamız gerekmektedir. Index numaraları “ 0 “ dan başlamaktadır. Eğer -1 değerini belirlersek ekranda hiçbir View görünmeyecektir.

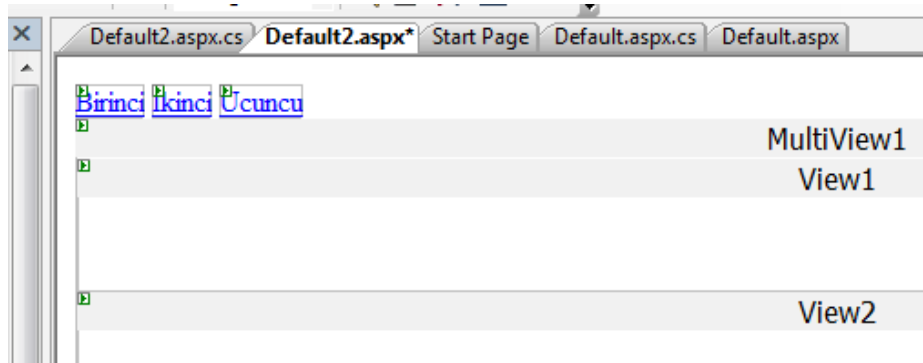
#### UYGULAMA 1

- 1) Bir web sayfası açalım ve bir tane Multiview ekleyeyelim.
- 2) Daha sonra Multiview’in içerisine 3 adet alt alta View ekleyelim (Şekil 2.35).



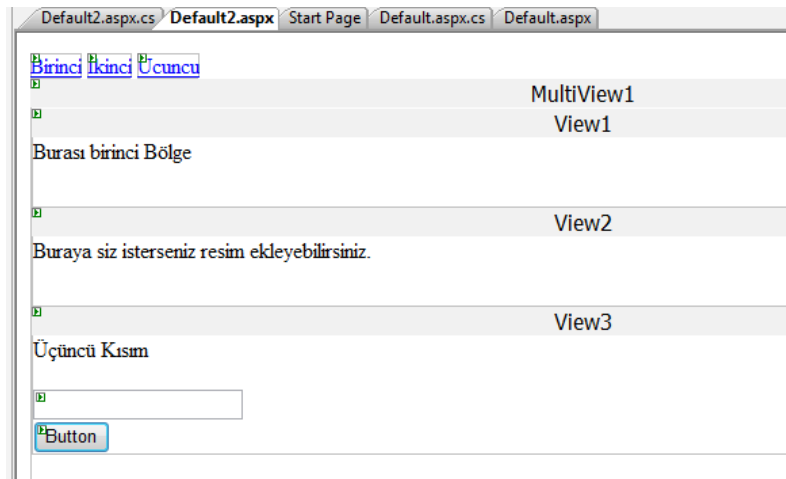
Şekil 2.35

- 3) Daha sonra MultiView’in üstüne 3 adet Link Button ekleyerek değerlerini değiştirelim (Şekil 2.36).



Şekil 2.36

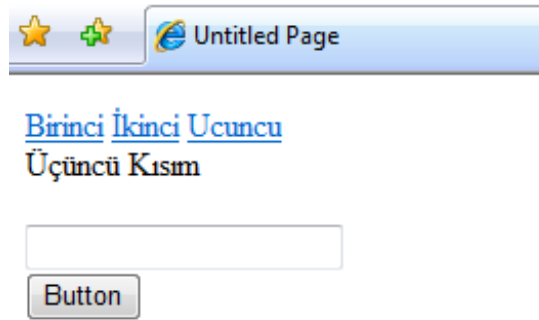
- 4) View1, View2 ve View3 ün içerisine istediğiniz nesneleri ekleyebilir veya yazabilirsiniz (Şekil 2.37).



Şekil 2.37

5) Daha sonra LinkButton'larımıza kodlarımızı yazarak çalıştıralım. Böylece hangi linki tıklarsak ona ait View görünecektir (Şekil 2.38).

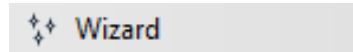
```
protected void LinkButton1_Click(object sender, EventArgs e)
{
    this.MultiView1.ActiveViewIndex = 0;
}
protected void LinkButton2_Click(object sender, EventArgs e)
{
    this.MultiView1.ActiveViewIndex = 1;
}
protected void LinkButton3_Click(object sender, EventArgs e)
{
    this.MultiView1.ActiveViewIndex = 2;
}
```



Şekil 2.38

#### ❖ WIZARD

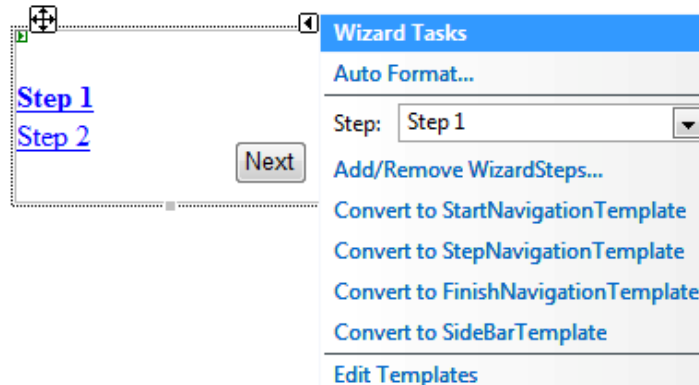
Web sayfalarında kayıt olurken; kullanıcıya sorulan soruları farklı formlar da gösteririz. Bunu yapmak için ya her formu bir sayfaya atarız, yada paneller kullanarak visible özelliklerini true-false yaparak paneli ayarladık. Bunun yerine Asp.Net 2.0 da Wizard kontrolü (Şekil 2.39) bu açıdan oldukça işimize yarayacaktır. Wizard 4 bölümden oluşur. Wizard adımları (Wizard Steps), Hareket düğmeleri (Navigation buttons), sidebar ve başlık (header) kısımlardan oluşur.



Şekil 2.39

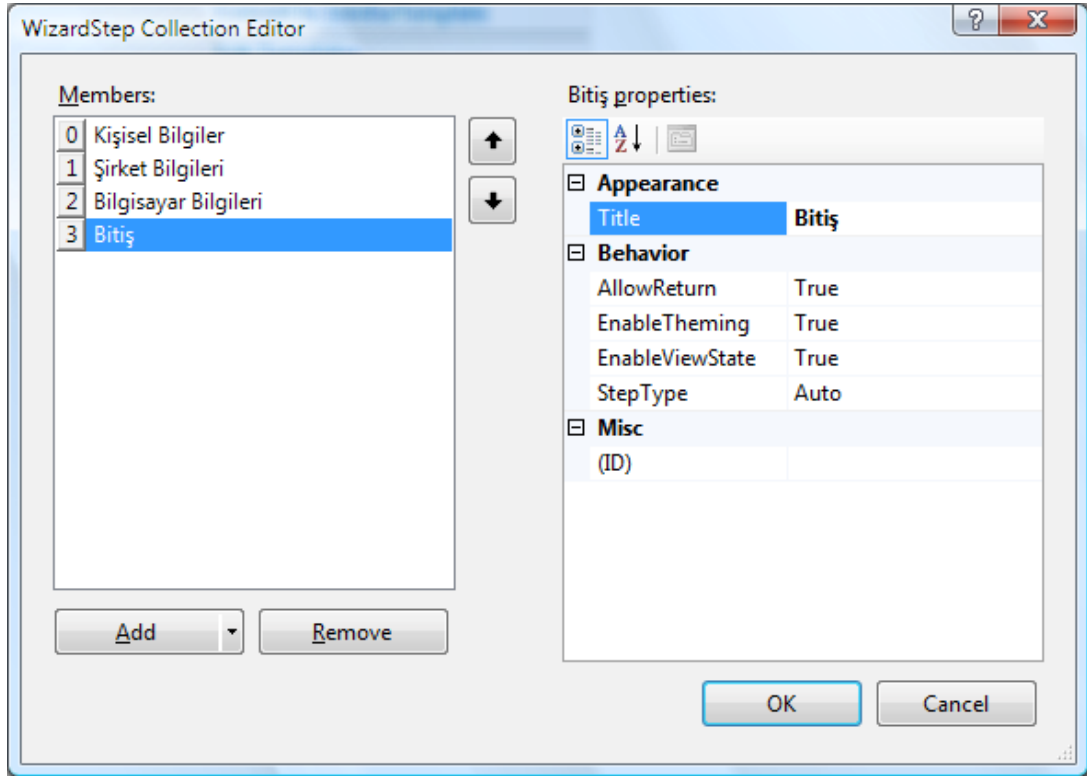
#### UYGULAMA 1

1) Web sayfamıza Wizard kontrolü ekleyelim.(Şekil 2.40)



Şekil 2.40

2) Karşımıza gelen pencereden Add/Remove WizardSteps seçeneğini seçiniz. Karşımıza bir pencere gelecektir (Şekil 2.41). Bilgileri şekildeki gibi değiştirip OK butonuna basınız.



Şekil 2.41

3) Daha sonra wizard ın Kişisel Bilgileri yazısını tıklayarak. Şekil 2.42 gibi bir görüntü oluşturunuz.

Şekil 2.42

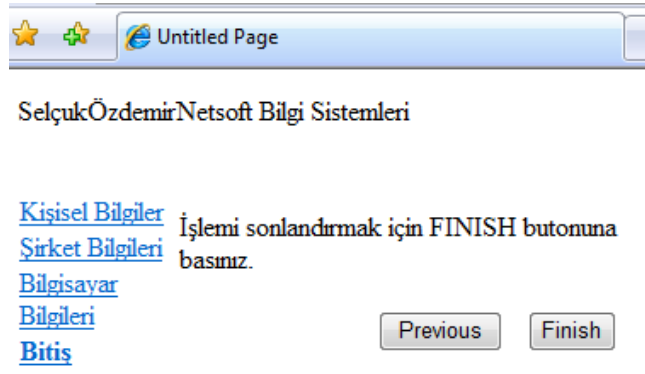
4) Daha sonraki adımlarıda tek tek yazılarını tıklayarak gerekli biçimlendirmeyi yapınız.(Şekil 2.43)

Şekil 2.43

Daha sonra FINISH butonumuzun üstünü çift tıklayarak code-behind alanına geçelim. Aşağıdaki kodumuzu yazalım.

```
protected void Wizard1_FinishButtonClick(object sender, WizardNavigationEventArgs e)
{
    Response.Write(this.TextBox1.Text);
    Response.Write(this.TextBox2.Text);
    Response.Write(this.TextBox3.Text);
    //Kişisel bilgiler ve Şirket bilgilerindek textbox lar için kodu yazdım
}
```

5) Kodumuzu çalıştırdıktan sonra textboxlarımıza değer girerek sonucu görebiliriz. (Şekil. 2.44)



Şekil 2.44

Böylelikle bir kaç işlemimizi tek bir alanda yaparak işlemlerimizi daha hızlı gerçekleştirebiliriz. Finish butonunu veritabanı işlemine bağlayarak Wizard kontrolü işinize yarayacaktır.

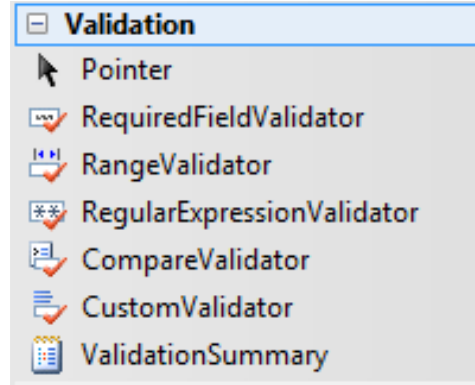


KONU 3

VALIDATION

## VALIDATION KONTROLLER

Web sayfalarında en önemli bir konu halini alan bilgi girişi ve bunların kontrollerini yapan kontrollerdir (Şekil 3.1).



Şekil 3.1

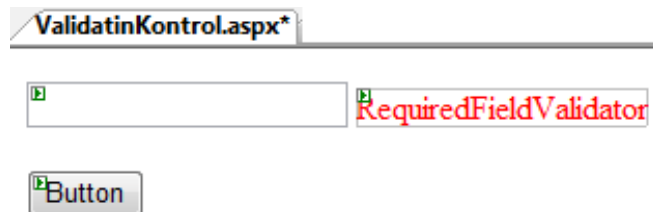
Kullanıcıların hatasız ve eksiksiz olarak formların doldurulmasında önemli rol oynamaktadır. Bu kontroller Javascript veya DHTML kullanılarak da yapılabilirdi. Fakat Asp.NET 2.0 bize sunmuş olduğu bu validation sayesinde daha anlaşılır ve hızlı olmamızı sağlamaktadır. Bu kontrollerin çalışması için Internet Explorer 4 ve üstü olması gerekmektedir. Bu kontrolleri genel olarak tanımlayalım.

| Validation Kontrol Adı       | Açıklama  |
|------------------------------|---|
| Required Field Validator     | Kontrol içerisine bir değer girilip girilmediğini kontrol eder. Atamış olduğumuz kontrol boş geçemez. |
| Range Validator              | Belirlediğimiz iki değer arasında kontrol eder. Örneğin yaş aralığı için kullanılabilir.              |
| Reguler Expression Validator | Karakter kontrolleri yapar. Mail adresinde @ işareti gib kontrollerde kullanılır.                     |
| Compare Validator            | İki giriş kontrolü arasındaki değeri, tipini veya dosyanı eşitliğini kontrol eder.                    |
| Custom Validator             | Validation için kendi kodumuzu yazmaya izin verir.  |

Validationları daha iyi anlamak için uygulamalar yapalım.

### UYGULAMA 1

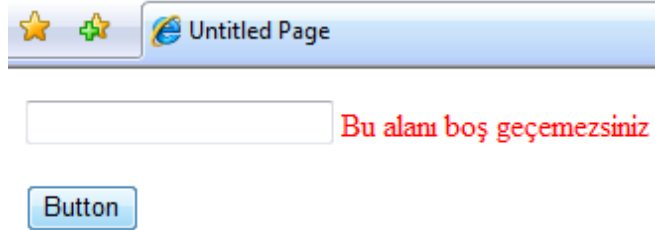
1) İlk uygulamamızda RequiredFieldValidator kontrolünü kullacağız. Bunun için web sayfamıza birer adet textbox, button ve RequiredFieldValidator kontrollerini ekleyelim (Şekil 3.2)



Şekil 3.2

2) RequiredFieldValidator kontrolünü tıklayalım ve properties penceresinden ErrorMessage özelliğini kullanarak hata mesajını “Bu alanı boş geçemezsiniz” olarak değiştirelim. Böylece kullanıcı hata yaptığında bu ileti çıkacaktır.

3) Daha sonra ControlToValidate özelliğinden TextBox1 i seçiniz ve kodumuzu çalıştırınız. Eğer TextBox’ı boş geçerek butona basarsanız hata çıkacaktır (Şekil 3.3). Fakat TextBox’ımıza birşeyler yazıp butona bastığımızda hata mesajı kaybolacaktır.



Şekil 3.3

## UYGULAMA 2

1) Bu uygulamamızda ise RangeValidator kontrolümüzü anlatacağım için. Web sayfamızda birer adet textbox, button ve RangeValidator kontrolü yerleştirelim.

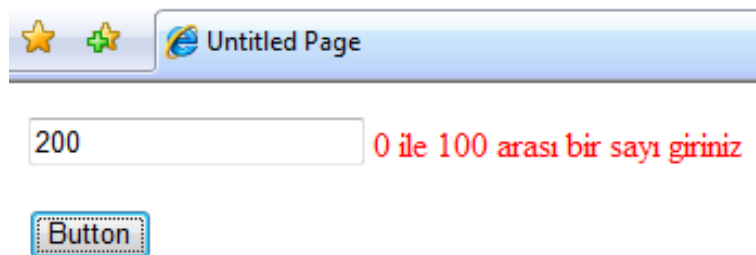
2) RangeValidator kontrolü tıklayarak properties penceresinden ErrorMessage kısmına “0 ile 100 arası bir sayı giriniz” mesajını yazınız.

3) Daha sonra yine RangeValidator kontrolünün MinimumValue özelliğine “0”, MaksimumValue özelliğine ise “100” değerlerini yazınız (Şekil 3.4).

|                 |                                  |
|-----------------|----------------------------------|
| EnableNaming    | true                             |
| EnableViewState | True                             |
| ErrorMessage    | 0 ile 100 arası bir sayı giriniz |
| Font            |                                  |
| ForeColor       | Red                              |
| Height          |                                  |
| MaximumValue    | 100                              |
| MinimumValue    | 0                                |
| SetFocusOnError | False                            |

Şekil 3.4

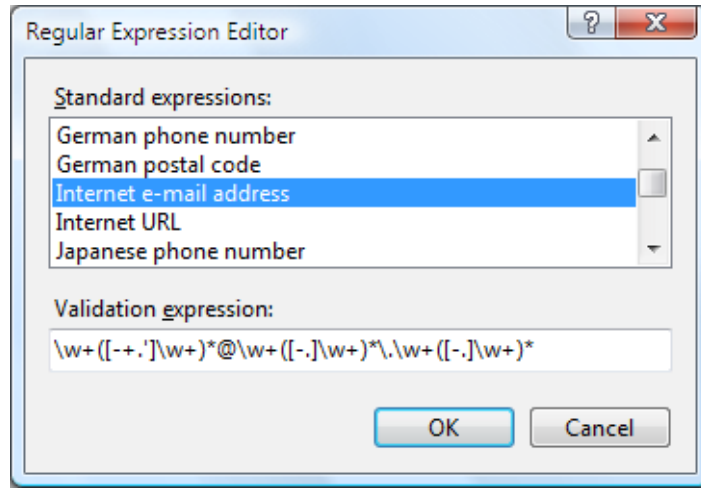
4) Değerleri girdikten sonra CompareToValidate seçeneğinden Textbox objenizi seçiniz. Seçtikden sonra kodumuzu çalıştıralım ve textbox a “200” gibi büyük bir rakam girelim. RangeValidation hata verecektir (Şekil 3.5). Eğer siz aralık olarak iki tarih aralığı belirlemek istiyorsanız Range Validation’nın Type özelliğini Date yapmanız gerekmektedir.



Şekil 3.5

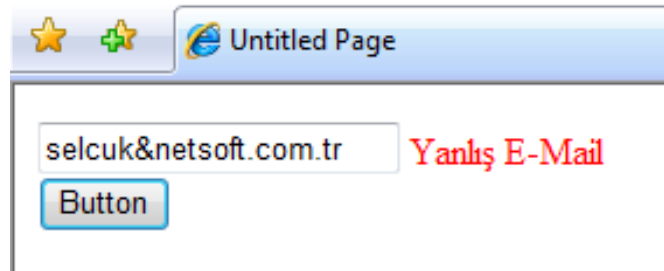
## UYGULAMA 3

- 1) Bu uygulamamızda ise RegularExpressionValidator kontrolünü anlatacağım. Bunu için Web sayfamıza birer adet textbox ve buton nesneleri yerleştiriniz. Textbox nesnemizin yanınada RegularExpressionValidator kontolunden koyalım.
- 2) RegularExpressionValidator kontrolüne ait properties penceresinden ErrorMessage kısmına “Yanlış E-Mail” yazısını, ContolToValidate seçeneğinden de textbox1’i seçiniz.
- 3) Properties penceresinde bulunan ValidationExpression seçeneğinin tıkladığınızda karşımıza Regular Expression Editor penceresi gelecektir. Gelen seçeneklerden telefon numarası, e-mail, web adresi,posta kodu gibi hazır kalıplar içermektedir. Biz be seçeneklerden Internet e-mail address seçeneğini seçiniz (Şekil 3.6).



Şekil 3.6

- 4) Burada textbox’a bilgi girişi yapıldığında “@” ve “.” işaretlerini kontrol edecektir (Şekil 3.7).



Şekil 3.7

Kalıp değerler dışında kendimize göre giriş karakter kontrolü yaptırabiliriz. Regular Expression Editor penceresinde üstte Custom seçeneğini işaretleyerek Validation expression alana ifadeleri yazınız. Karakterleri aşağıdaki tabloda inceleyebilirsiniz.

| Karakter | Tanımı  | Örnek                |
|----------|---|----------------------|
| a        | Yazdığımız bir karakter a ise a değerini bekler   | a = a                |
| 1        | Rakamsal değer girişi 1 ise 1 değeri bekler       | 1 = 1                |
| *        | 0 veya daha fazla önceki karakterin tekrarlanması | 5* = ,5,55,555       |
| +        | 1 veya daha fazla önceki karakterin tekrarlanması | 5+ = 5,55,555        |
| ()       | Gruplama  | (55)+ = 55,5555      |
|          | İki seçenekten birisi yazılacak                   | 1 2 = 1 veya 2       |
| []       | İki değer ve arasındaki sayılar girilecek         | [1-4] = 1,2,3 veya 4 |

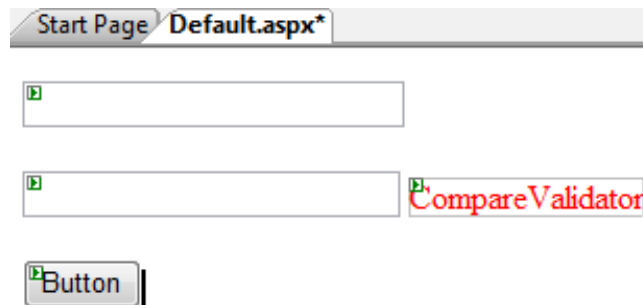
|     |  |  |
|-----|--|--|
| [^] | Belirtilen aralık dışındaki tür değerler girilecek | [^1-3] =1,2 ve3 dışındaki tüm değerler |
| \s  | Boşluk Karakteri                                   | \s = , ,                               |
| \S  | Boşluk dışındaki tüm karakterler                   | \S+ = abc,123                          |
| \d  | Rakamsal değer girilir                             | \d+ = 1234567890                       |
| \D  | Rakam olmayan tüm karakterler                      | \D+ = abcdefgh                         |
| \w  | Kelimelerde kullanılan tüm karakterler             | \w+ = netsoft                          |
| {}  | Belirtilen karakterin tekrarlanma adeti            | a{1} = a<br>b {1-3} = b,bb veya bbb    |

Bunların dışında sizlere bir kac regular expressionlar yazılır.

| Veri           | Expression Kodu            | Tanımı                                 |
|----------------|----------------------------|--|
| E-Posta Adresi | \S+@\S+\.\S+               | @ ve . işaretleri kontrol edilecektir. |
| Şifre          | \w{4-8}                    | En az 4 en fazla 8 karakterlik şifre   |
| Telefon Girişi | \d{3}\s\d{3}\s\d{2}\s\d{2} | 312 473 20 74 formata uyacaktır.       |

#### UYGULAMA 4

1) Bu uygulamamızda CompareValidator kontrolünden bahsedeceğim. Uygulamamız için Web sitemize alt alta iki adet textbox ve bir adet de buton koyalım. İkinci textbox 'ımızın yanına CompareValidator kontrolü koyalım (Şekil 3.8).



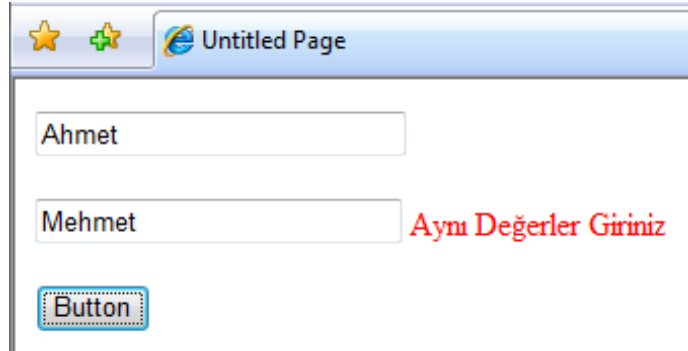
Şekil 3.8

2) CompareValidator kontrolünün properties penceresinde ErrorMessage yazısını "Aynı Değerler Giriniz" olarak değiştiriniz. Daha sonra ControlToCompare seçeneğinde textbox1'i, ControlToValidate seçeneğinde ise textbox2'i seçiniz (Şekil 3.9).

|                        |                       |
|------------------------|-----------------------|
| ControlToCompare       | TextBox1              |
| ControlToValidate      | TextBox2              |
| CssClass               |                       |
| CultureInvariantValues | False                 |
| Display                | Static                |
| EnableClientScript     | True                  |
| Enabled                | True                  |
| EnableTheming          | True                  |
| EnableViewState        | True                  |
| ErrorMessage           | Aynı Değerler Giriniz |

Şekil 3.9

3) Daha sonra kodumuzu çalıştıralım. Textbox1'e ve Textbox2 kontrollerine aynı değerleri yazmamız gerekmektedir. Buradaki seçeneklerden ControlToCompare seçeneğinde TextBox1'i seçtiğimiz için kontrol bu nesne üzerinden olmaktadır (Şekil 3.10).



Şekil 3.10

Kontrolleri kullanarak Web formlarımızda kullanıcı hatalarını büyük oranda engelleyerek doğru bilgilerin gönderilmesini sağlayabiliriz.

KONU 4

VERİTABANI

## VERİTABANI NEDİR?

Bir veritabanı isminden de anlaşılacağı gibi bilgilerin depolandığı hiyerarşik bir yapıdır. Başka bir tanımla digital ortamda (bilgisayar) saklanan düzenli verilerdir de denilebilir.

## ADO.NET

Web üzerinden veritabanına bağlanmak ve veri işlemleri gerçekleştirmek için ADO.NET teknolojiden faydalanarak işlemlerimizi gerçekleştireceğiz. ADO.NET her ne kadar ADO teknolojisine benzesede aslında çok fazla ilgisi yoktur. ADO.NET (Activex Data Obje) tamamiyle yeniden geliştirilen yapısı ile verilere ulaşır. ADO.NET verilere erişirken ADO gibi sadece OLEDB Provider değil, onun dışında SQL Provider yapısını da kullanır. ADO.NET veri geçişlerinde XML yapısına tam uyumlu olduğundan sağlam ve performansı yüksek her ortama uyumludur.

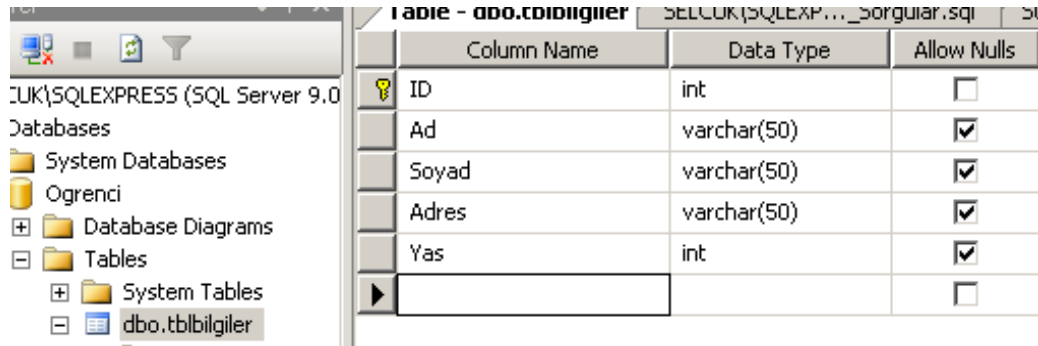
.NET Platformunu yüklediğimizde sahip olduğumuz Providerlar Odbc, Oledb, SqlClient gelmektedir. Fakat Oracle veritabanınıza bağlanmak istiyorsanız [www.oracle.com](http://www.oracle.com) adresinden OPD.NET'i indirmeniz gerekmektedir.

## VERİTABANINA BAĞLANTI

Bir web formu üzerinden veritabanına bağlantımızı anlatacağım.

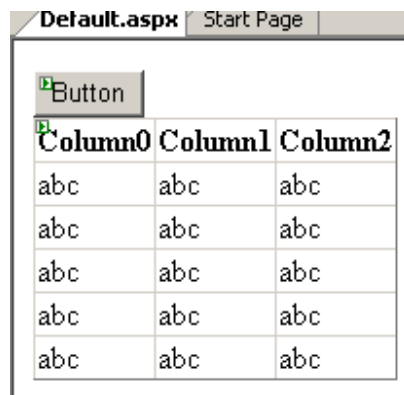
## UYGULAMA 1

Bu uygulamamızda Sql Server 2005'de hazırlamanız gereken veritabanı örneği aşağıda yer almaktadır. Örneğimizde "Ogrenci" adında bir database (veritabanı), bunun içerisinde "tblBilgiler" adında bir tablomuz yer almaktadır. Veritabanımızı yaptıktan sonra içerisine bilgiler giriniz.



| Column Name | Data Type   | Allow Nulls                         |
|-------------|-------------|-------------------------------------|
| ID          | int         | <input type="checkbox"/>            |
| Ad          | varchar(50) | <input checked="" type="checkbox"/> |
| Soyad       | varchar(50) | <input checked="" type="checkbox"/> |
| Adres       | varchar(50) | <input checked="" type="checkbox"/> |
| Yas         | int         | <input checked="" type="checkbox"/> |
|             |             | <input type="checkbox"/>            |

1) Web sayfamıza birer GridView ve Button nesnelerimizi ekleyelim (Şekil 4.1).



| Column0 | Column1 | Column2 |
|---------|---------|---------|
| abc     | abc     | abc     |
| abc     | abc     | abc     |
| abc     | abc     | abc     |
| abc     | abc     | abc     |
| abc     | abc     | abc     |

Şekil 4.1

2) Butonu çift tıklayarak kod alanına geçelim. Buton1\_Click olayına aşağıdaki kodumuzu yazalım.

```
using System.Data; //EKLEMİYİ UNUTMAYIN
```



```
using System.Data.SqlClient; //EKLEMEYİ UNUTMAYIN
```

```
protected void Button1_Click(object sender, EventArgs e)
{
    string yol=@"Data Source=SELCUK\SQLEXPRESS;initial Catalog=Ogrenci;integrated
security=sspi";
    SqlConnection baglan=new SqlConnection(yol);

    string sorgu = "select * from tblbilgiler";
    SqlDataAdapter sadp = new SqlDataAdapter(sorgu,baglan);

    DataSet ds = new DataSet();
    sadp.Fill(ds);

    this.GridView1.DataSource = ds.Tables[0];
    this.GridView1.DataBind();
}
```

3) Kodumuzu çalıştırmadan önce biraz anlatalım.

Bir veritabanından bilgi almak için bazı kurallarımız söz konusudur. Bunlar birincisi veritabanımızla bağlantı kurmaktır.

Bağlatımızı SqlConnection sınıfı ile gerçekleştiriyoruz. Sınıfımızın aldığı parametrede ise;

**Data Source:** Veritabanımızın (SQL Server) ismini veya kurulu olduğu bilgisayarımızın IP sini yazmamız gerekmektedir.

Data Source= Selcuk

Data Source=Selcuk\\SqlExpress

Data Source=192.168.1.100

**initial Catalog:** Bu alana Sql Serverda oluşturduğumuz veritabanımızın ismini yazmamız gerekiyor.

initial Catalog=Ogrenci

**integrated security:** Burası bizim güvenlik kısımlarını yazacağımız alan, SQL Server kurulma esnasında eğer System Account ile kuruldu ise integrated security=sspi dememiz yeterlidir. Fakat kullanıcı adı ve şifre girildi ise uid ve pwd kısımları eklenmelidir.

integrated security=sspi

integrated security= true

integrated security= true;uid=sa;pwd=sa

integrated security=true;uid=sa;pwd=; //şifrenin boş olması durumunda

İkinci aşamada SqlDataAdapter kısmı SQL sorgularımızı yorumlaması gereken bir sınıf olarak değerlendirebiliriz. Böylece select sorgusu yorumlanıyor ve sonuc Dataset e aktarılaraq sanal bir tablonun oluşturulması sağlanmaktadır. Artık DataSet'i GridView'e aktararak sonuçların görüntülenmesi sağlanacaktır.

4) Artık kodumuzu çalıştırarak butonumuza bastığımızda karşımıza datalarımız gelecektir (Şekil 4.2).



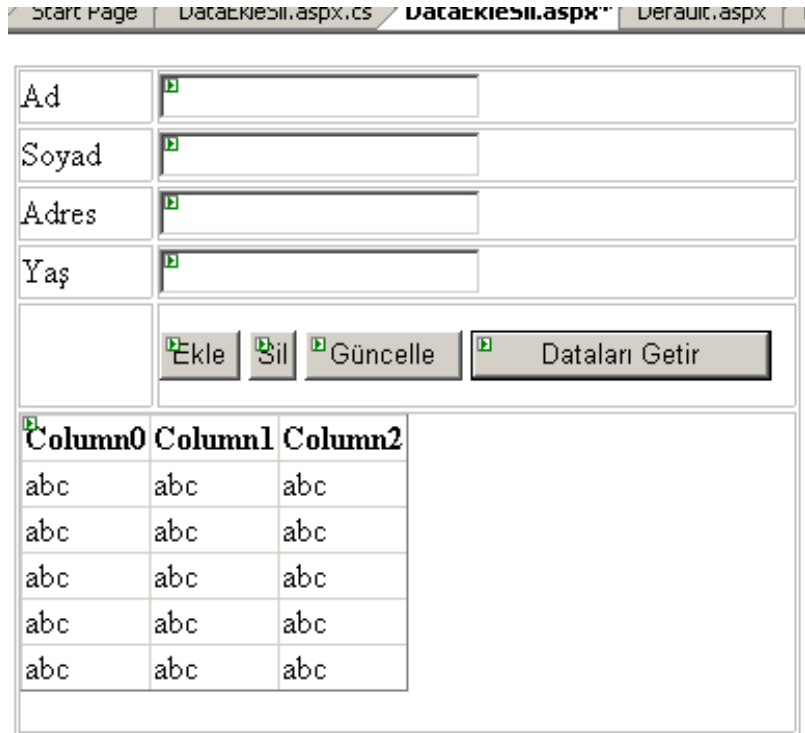
| ID | Ad     | Soyad         | Adres    | Yas |
|----|--------|---------------|----------|-----|
| 1  | Kadir  | KORUTÜRK      | Ankara   | 37  |
| 2  | Funda  | KOYUNCU       | Ankara   | 29  |
| 3  | Selçuk | ÖZDEMİR       | Ankara   | 26  |
| 4  | Cem    | BAYRAKTAROĞLU | Istanbul | 27  |
| 5  | Mehmet | EĞİTMEN       | Izmir    | 39  |

Şekil 4.2

Datalarımızı getirdikten sonra, veritabanımıza kayıt eklemek çıkartmak gibi sql sorgularla işlemlerimizi gerçekleştirebiliriz.

## UYGULAMA 2

1) Bu uygulamamızda Web sitemizi dört textbox, dört buton ve bir tanede GridView yerleştirelim (Şekil 4.3).



Şekil 4.3

2) Aşağıdaki kodu yazdıktan sonra çalıştırılalım

```
using System.Data; //Eklemeyi unutma
using System.Data.SqlClient; //Eklemeyi Unutma
```

```
string yol = "Data Source=Selcuk\\SqlExpress;"
            +"initial Catalog=Ogrenci;integrated security=sspi";
```

```
SqlConnection baglanti;
SqlDataAdapter sadp;
DataSet verial;
```

```

protected void btnEkle_Click(object sender, EventArgs e)
{
    string ad, soyad, adres, yas;
    ad = this.txtAd.Text;
    soyad = this.txtSoyad.Text;
    adres = this.txtAdres.Text;
    yas = this.txtYas.Text;

    baglanti = new SqlConnection(yol);
    string ekle = "insert into tblBilgiler"
        + " (ad,soyad,adres,yas) values "
        + "("+ad+"','"+soyad+"','"+adres+"','"+yas+"")";

    sadp = new SqlDataAdapter(ekle, baglanti);
    verial = new DataSet();
    sadp.Fill(verial);

    Response.Write("Kayıt Eklendi");
    //Textboxlara Değerlerimizi Girip
    //Ekle Butonuna bastığınızda Kayıt Eklendi
    //Yazısı Çıkacaktır
}
protected void btnSil_Click(object sender, EventArgs e)
{
    string ad;
    ad = this.txtAd.Text;

    baglanti = new SqlConnection(yol);
    string sil = "delete from tblbilgiler where Ad='"+ad+"'";

    sadp = new SqlDataAdapter(sil, baglanti);
    verial = new DataSet();
    sadp.Fill(verial);

    Response.Write("Adı "+ad+" olan Kayıt Silindi");
    // Ad kısmında girilen isim silinecektir
}
protected void btnGuncelle_Click(object sender, EventArgs e)
{
    string ad;
    ad = this.txtAd.Text;

    baglanti = new SqlConnection(yol);
    string guncelle = "Update tblbilgiler set yas=30 where ad='"+ad+"'";
    //Adını girdiniz kişinin YAŞI 30 olacaktır.
    sadp = new SqlDataAdapter(guncelle, baglanti);
    verial = new DataSet();
    sadp.Fill(verial);

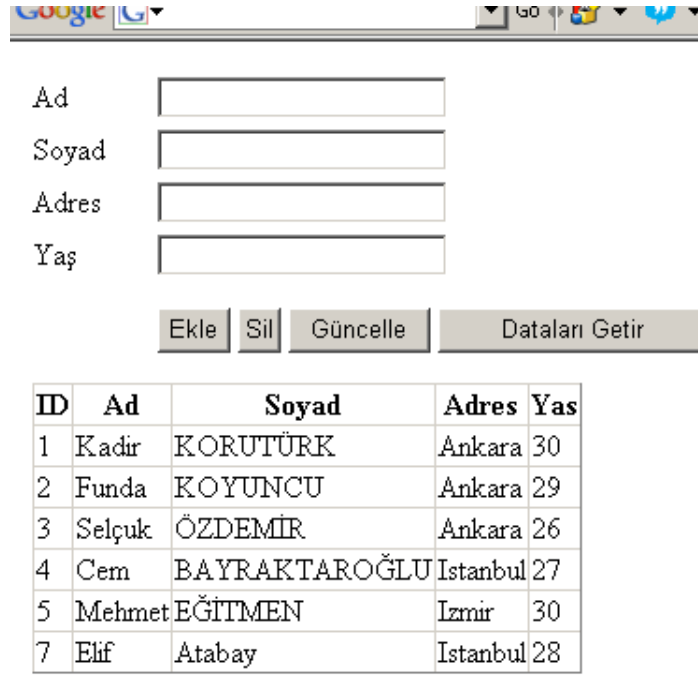
    Response.Write("Yaş Güncellendi");
    //Adını girdiğimiz Kayıt da yaş 30 olacaktır
}
protected void btnDataGetir_Click(object sender, EventArgs e)
{
    baglanti = new SqlConnection(yol);
    string guncelle = "select * from tblbilgiler";

    sadp = new SqlDataAdapter(guncelle, baglanti);
    verial = new DataSet();
    sadp.Fill(verial);

    this.GridView1.DataSource = verial.Tables[0];
    this.GridView1.DataBind();
}

```

Kodu yazdıktan sonra çalıştırdığımızda veritabanımıza kayıt işlemlerini gerçekleştirmiş olacaktır (Şekil 4.4).



| ID | Ad     | Soyad         | Adres    | Yas |
|----|--------|---------------|----------|-----|
| 1  | Kadir  | KORUTÜRK      | Ankara   | 30  |
| 2  | Funda  | KOYUNCU       | Ankara   | 29  |
| 3  | Selçuk | ÖZDEMİR       | Ankara   | 26  |
| 4  | Cem    | BAYRAKTAROĞLU | Istanbul | 27  |
| 5  | Mehmet | EĞİTMEN       | Izmir    | 30  |
| 7  | Elif   | Atabay        | Istanbul | 28  |

Şekil 4.4



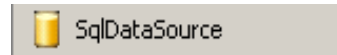
Access veritabanına bağlanmak için Data.SqlClient sınıfını yerine Data.OleDb sınıfını getirmeniz gerekmektedir.

Provider satırı Sql Server bağlantısından farklıdır.

string baglan= "Provider=Microsoft.Jet.OleDb.4.0;Data Source=c:\\veritabani.mdb";  
şeklinde yazılmalıdır.

### SQLDATASOURCE

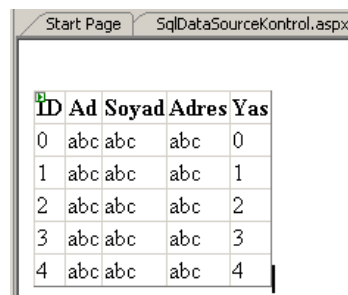
Aslında tüm veritabanlarına bağlanmak için kullanılır (Şekil 4.6). Aslen Sql Server ve Oracle gibi server tabanlı veritabanlara bağlanmak için kullanılır. Uygulamamızda SqlDataSource nesnesini kullanarak GridView nesnemize datalarımızı getireceğiz.



Şekil 4.6

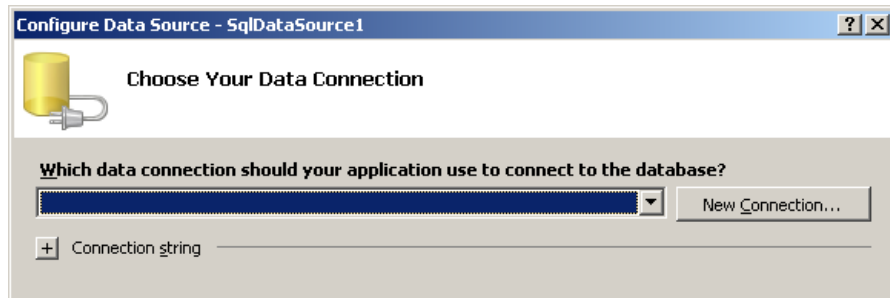
### UYGULAMA 3

1) Bu uygulamamız için web sayfamıza birer adet GridView koyalım (Şekil 4.5).



Şekil 4.5

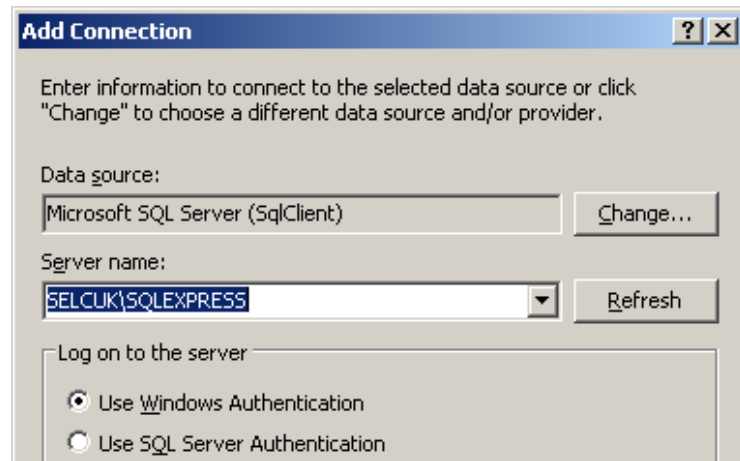
2) Web sayfamıza SqlDataSource nesnesi koyunuz. Daha sonra SqlDataAdapter den Configure Data Source kısmında tıklayarak Configure Data Source penceresini açınız (Şekil 4.6).



Şekil 4.6

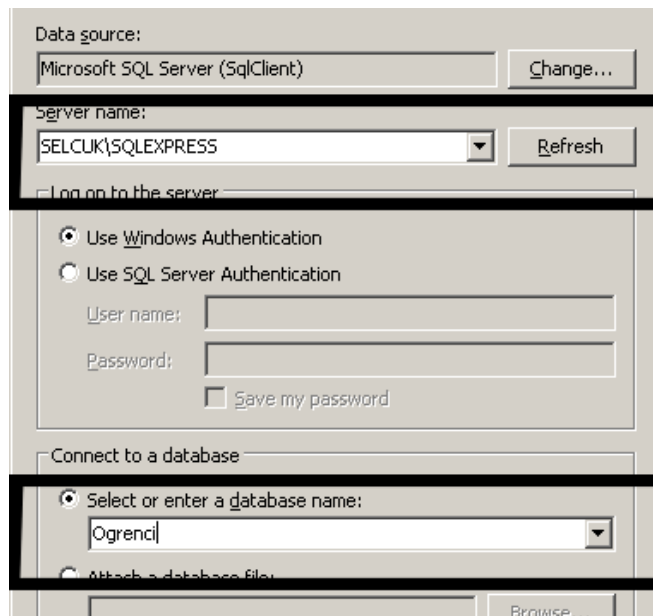
3) Açılan pencereden New Connection butonunu tıklayınız.

4) Karşımıza gelen kısımdan Add Connection penceresinden Server Name kısmından Sql Server adını seçiniz (Şekil 4.7).



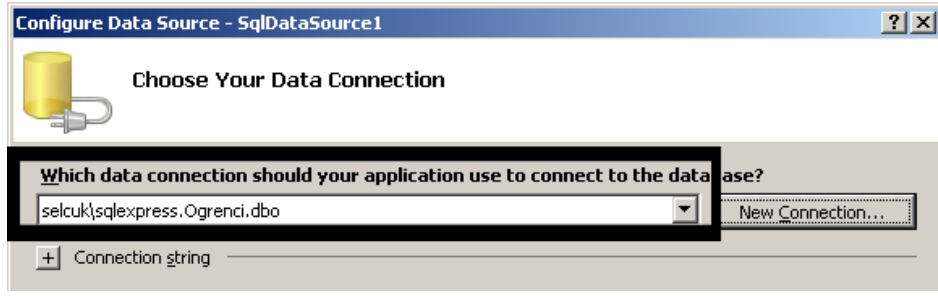
Şekil 4.7

5) Aynı penceresinin altında bulunan Select or enter a databa name kısmından Ogrenci veritabanımızı seçiniz (Şekil 4.8).



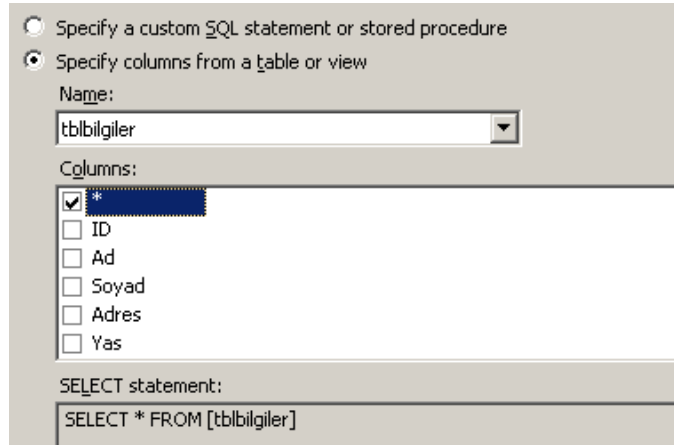
Şekil 4.8

- 6) Veritabanımızda seçtikten sonra OK butonuna basarak pencereyi kapatalım.  
 7) Daha sonra seçilen database ismi ekranda görünecektir Şekil(4.9).



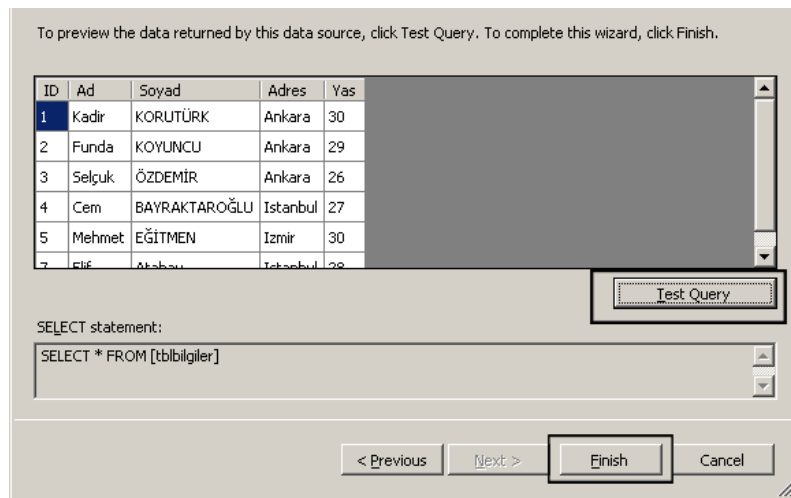
Şekil 4.9

- 8) Daha sonra Next butona basarak Connectionstring parametresi gelecektir. Burayı geçmek içinde next butonuna basınız.  
 9) Daha sonraki pencerede işlemlerin görülmesi gereken kolonları seçmemiz gerekecektir. Bütün kolonları seçmek için “ \* ” checkbox’ını tıklayarak next butonuna basınız (Şekil 4.10).



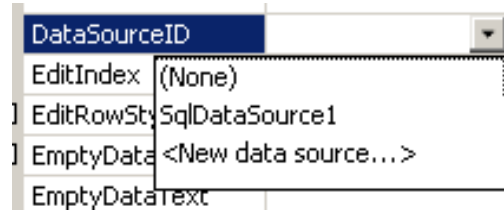
Şekil 4.10

- 10) Daha sonraki pencerede işlemimizin sonuna gelmiş bulunmaktayız. Test Query butonuna bastığınızda karşımıza tblbilgiler deki tüm datalar gelecektir. İşlemimize Finish diyerek bitirelim (Şekil 4.11).



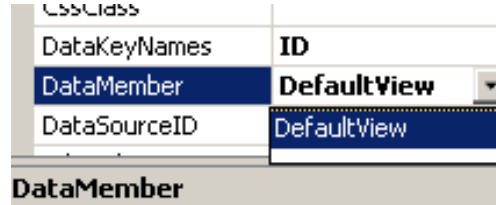
Şekil 4.11

- 11) Böylece SqlServer üzerinde bulunan Ogrenci veritabanındaki tblOgrenciler tablosuna bağlanmış durumdayız.  
 12) GridView nesnemizi tıklayarak properties penceresinden DataSourceID özelliğinden SqlDataSource1 seçeneğini tıklayınız (Şekil 4.12).



Şekil 4.12

13) DataMember özelliğinden ise DefaultView seçeneğini seçiniz (Şekil 4.13).



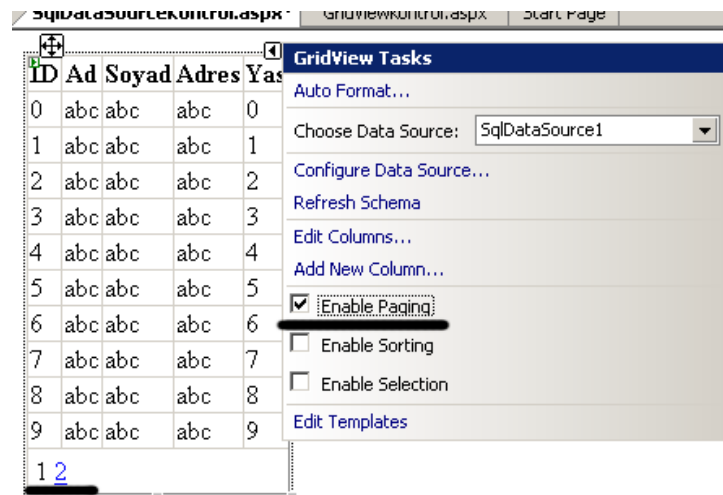
Şekil 4.13

14) Artık kodumuzu çalıştırarak GridView'e datalarımızın aktarıldığını görebiliriz (4.14).

| ID | Ad     | Soyad         | Adres    | Yas |
|----|--------|---------------|----------|-----|
| 1  | Kadir  | KORUTÜRK      | Ankara   | 30  |
| 2  | Funda  | KOYUNCU       | Ankara   | 29  |
| 3  | Selçuk | ÖZDEMİR       | Ankara   | 26  |
| 4  | Cem    | BAYRAKTAROĞLU | Istanbul | 27  |
| 5  | Mehmet | EĞİTMEN       | Izmir    | 30  |
| 7  | Elif   | Atabay        | Istanbul | 28  |

Şekil 4.14

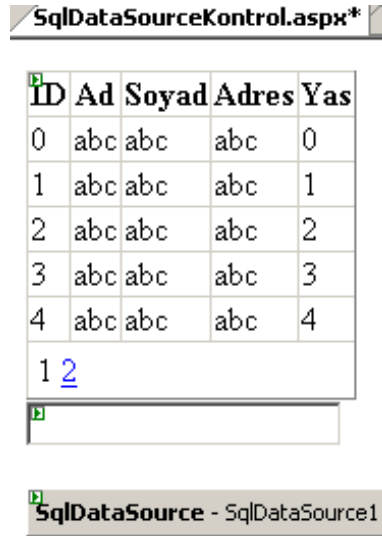
15) Database bağlantı işlemimiz bittikten sonra GridView seçeneğini tıkladığınızda üstte çıkan ok işaretini tıkladığınızda karşımıza gelen pencereden Enable Paging seçeneğini işaretleyerek sayfalarını oluşmasını sağlayabiliriz (Şekil 4.15). Her sayfada görünecek kayıt sayısını properties penceresinden Page Size değerini değiştirerek ayarlayabilirsiniz.



Şekil 4.15

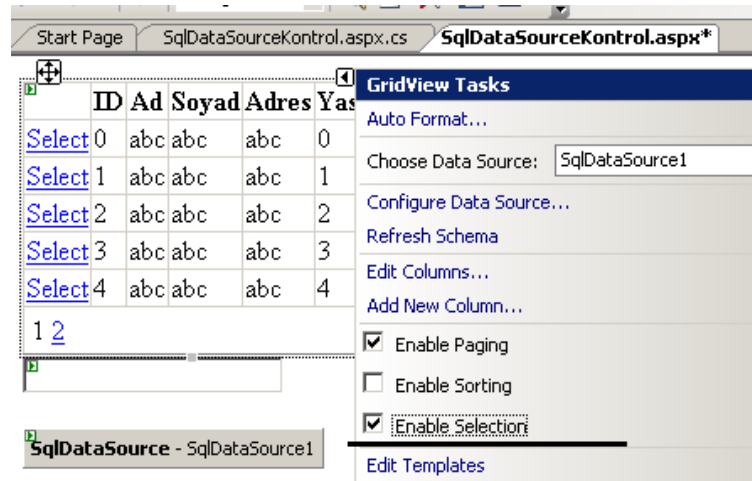
#### UYGULAMA 4

1) Uygulamamızda bir önceki uygulamamızın devamı olacaktır. Bunun için bir önceki projemizi açalım ve ek olarak bir textbox nesnesi ekleyelim (Şekil 4.16).



Şekil 4.16

2) GridView'i dan Enabled Selection seçeneğini seçiniz (4.17). Böylelikle seçtiğimiz satırı belirlemiş olacağız.



Şekil 4.17

3) Daha sonra GridView nesnemizi çift tıklayarak GridView1\_SelectedIndexChanged olayına aşağıdaki kodumuzu yazalım.

```
protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
{
    int deger = this.GridView1.SelectedIndex;
    this.TextBox1.Text = this.GridView1.Rows[deger].Cells[2].Text;
}
```

4) Böylece seçtiğimiz satıra ait Ad kolonu textbox'a aktarılacaktır (Şekil 4.18).



|        | ID | Ad     | Soyad         | Adres    | Yas |
|--------|----|--------|---------------|----------|-----|
| Select | 1  | Kadir  | KORUTÜRK      | Ankara   | 30  |
| Select | 2  | Funda  | KOYUNCU       | Ankara   | 29  |
| Select | 3  | Selçuk | ÖZDEMİR       | Ankara   | 26  |
| Select | 4  | Cem    | BAYRAKTAROĞLU | Istanbul | 27  |
| Select | 5  | Mehmet | EĞİTMEN       | Izmir    | 30  |
| 1 2    |    |        |               |          |     |
|        |    | Kadir  |               |          |     |

Şekil 4.18

### DATALIST

En çok ve oldukça faydalı olan bir kontrol olan DataList bir çok site yapma aşamanızda faydalı olacaktır. Bende bu doğrultuda sizlere bir portal mantığı ile anlatacağım. Şimdi düşünelim bir bir sitede sol tarafta kategorilerin bulunduğu ve bu kategorilere bağlı makale başlıkları ve bu makalelerinde detaylarını kullanıcılara okutmanın olduğunu kolay göreceksiniz.

Uygulamamıza başlamadan önce Öğrenci veritabanımıza bir tablolar ekleyelim. Tablolarımızın yapıları aşağıda yer almaktadır. Bu tabloların biri sol menü için kategoriler bulunacaktır. Diğer tablomuzda ise makale durumlarını bildirecek bir tablo olacaktır (Şekil 4.19).

| Table - dbo.tblKategori |             | Summary     |                                     |
|-------------------------|-------------|-------------|-------------------------------------|
|                         | Column Name | Data Type   | Allow Nulls                         |
| 🔑                       | ID          | int         | <input type="checkbox"/>            |
|                         | Kategori    | varchar(50) | <input checked="" type="checkbox"/> |
|                         | KatDurum    | varchar(20) | <input checked="" type="checkbox"/> |
| ▶                       |             |             | <input type="checkbox"/>            |

| Table - dbo.tblMakale |             | Table - dbo.tblKategori |                                     | Summary |             |
|-----------------------|-------------|-------------------------|-------------------------------------|---------|-------------|
|                       | Column Name | Data Type               | Allow Nulls                         |         | Column Name |
| 🔑                     | ID          | int                     | <input type="checkbox"/>            |         |             |
|                       | Baslik      | varchar(30)             | <input checked="" type="checkbox"/> |         |             |
|                       | Tanitim     | varchar(150)            | <input checked="" type="checkbox"/> |         |             |
|                       | Yazar       | varchar(50)             | <input checked="" type="checkbox"/> |         |             |
|                       | Detay       | ntext                   | <input checked="" type="checkbox"/> |         |             |
|                       | KayDurum    | varchar(20)             | <input checked="" type="checkbox"/> |         |             |

tblKategori

tblMakale

Şekil 4.19

Tablolarımıza girilen değerler aşağıda yer almaktadır (Şekil 4.20).

| Table - dbo.tblMakale |              | Table - dbo.tblKategori |                  | Summary |  |
|-----------------------|--------------|-------------------------|------------------|---------|--|
| ID                    | Kategori     | KatDurum                |                  |         |  |
| 1                     | C# .NET      | csharp                  | } Dikkatli yazın |         |  |
| 2                     | ASP.NET      | aspnet                  |                  |         |  |
| 3                     | C++ .NET     | cplus                   |                  |         |  |
| 4                     | Web Tasarımı | web                     |                  |         |  |
| ►*                    | NULL         | NULL                    |                  |         |  |

| Table - dbo.tblMakale |                | Table - dbo.tblKategori |                | Summary         |          |
|-----------------------|----------------|-------------------------|----------------|-----------------|----------|
| ID                    | Baslik         | Tanitim                 | Yazar          | Detay           | KayDurum |
| 1                     | C# Nedir?      | Bu makalemizde ...      | Selçuk ÖZDEMİR | Microsoft tar.  | csharp   |
| 2                     | ASP.NET Nedir? | Makalemizde AS...       | Selçuk ÖZDEMİR | .NET Framew.    | aspnet   |
| 3                     | ASP.NET Ekranı | Web sayfalarımı...      | Selçuk ÖZDEMİR | Web sayfalar... | aspnet   |
| ►*                    | NULL           | NULL                    | NULL           | NULL            | NULL     |

tblKategorideki KatDurum kolonlarını dikkatle yazınız.

tblMakale tablosundaki KayDurum kolonuna bağlı olarak sorgu çekilecektir.

Şekil 4.20

Veritabanı işlemlerimizi bittiğine göre artık kodlama ile ilgili işlemlerimize devam edebiliriz.

## UYGULAMA 5

1) Yeni bir web sayfası açalım ve sayfamıza bir adet datalist nesnemizi ekleyelim. Daha sonra web sayfamızı çift tıklayarak Form\_Load olayına aşağıdaki kodumuzu yazalım.

using System.Data.SqlClient; //Eklemeyi unutmayınız.

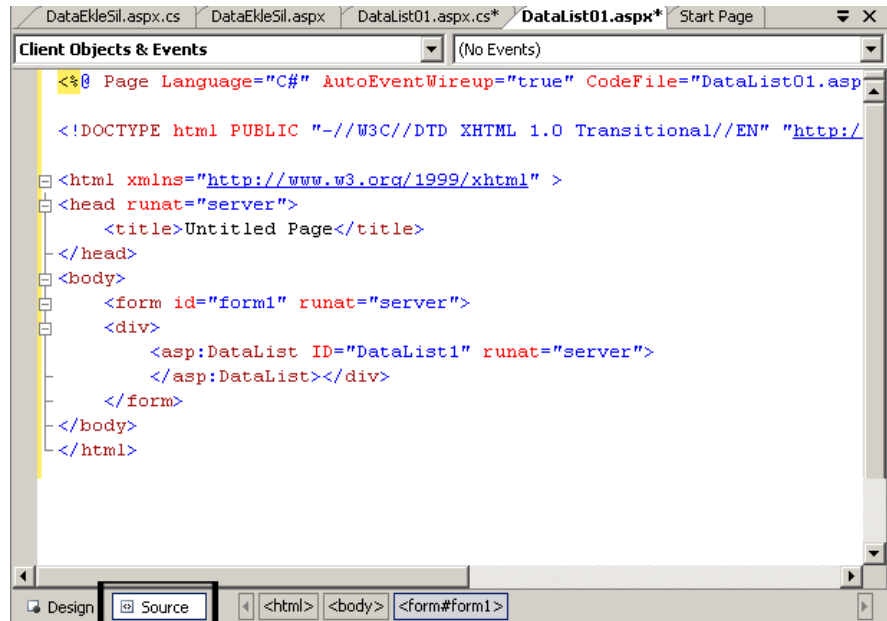
```
protected void Page_Load(object sender, EventArgs e)
{
    string yol = "Data Source=Selcuk\\SqlExpress;"
    + "initial Catalog=Ogrenci;integrated security=sspi";

    SqlConnection baglan = new SqlConnection(yol);

    string sorgu = "select * from tblKategori";
    //Kategori tablosunun tüm kolonları getir.
    SqlDataAdapter sadp = new SqlDataAdapter(sorgu,baglan);
    DataSet ds = new DataSet();
    sadp.Fill(ds);

    this.DataList1.DataSource = ds.Tables[0];
    this.DataList1.DataBind();
}
```

2) DİKKAT! Yukarıdaki kodu yazıp çalıştırdığımızda karşımıza bilgiler gelmeyecektir. Bilgilerin gelmesi için kolonları tek tek istediğimiz şekilde çağırabiliriz. Bunun için Code-Behind alanından çıkarak Source kısmına geliniz (Şekil 4.21).



Şekil 4.21

3) DataList kod içerisine gelerek aşağıdaki kodu yazınız (Şekil 4.22).

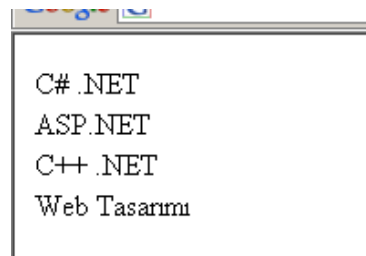
```

</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:DataList ID="DataList1" runat="server">
                <ItemTemplate>
                    <tr>
                        <td>
                            <%#DataBinder.Eval(Container.DataItem, "Kategori") %>
                        </td>
                    </tr>
                </ItemTemplate>
            </asp:DataList></div>
        </form>
    </body>

```

Şekil 4.22

4) Kodumuzu yazdıktan sonra çalıştırdığımızda Kategoriler ekrana gelecektir (Şekil 4.23).



Şekil 4.23

5) Bundan sonraki yapacağımız işlem bu kategorilere link vererek tıkladığımız birime göre makalelerin gelmesini sağlayacağız. Web sayfamıza Web Site > Add New Item menüsünden yeni bir web form ekleyelim. Web formumuzun ismi Default2.aspx olarak kalabilir.

6) Default.aspx deki source kısmındaki kodu değiştirerek kodumuzu çalıştıralım (Şekil 4.24).

```

<body>
    <form id="form1" runat="server">
        <div>
            <asp:DataList ID="DataList1" runat="server">
                <ItemTemplate>
                    <tr>
                        <td>
                            <a href="Default2.aspx?Gonder=<%#DataBinder.Eval(Container.DataItem, "KatDurum") %>">
                                <%#DataBinder.Eval(Container.DataItem, "Kategori") %></a>
                            </td>
                        </tr>
                    </ItemTemplate>
                </asp:DataList></div>
            </form>
        </body>

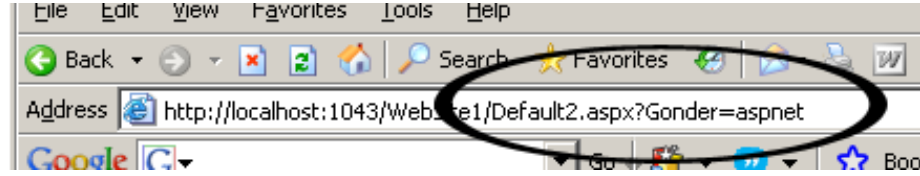
```

Şekil 4.24

7) Bu kodumuzda yer alan önemli bir kısım hangi sayfaya gideceği ve hangi değeri taşıyacağını belirlemiş olacaktır.

Default2.aspx : Hangi dosyayı açacağımızı belirtebiliriz.

?Gonder= : Gonder değişkenine göndereceğimiz değeri belirler. Biz burada gelen datamıza ait KatDurum kolonunu gönderiyoruz (Şekil 4.25).



Şekil 4.25

- 8) Artık Default2.aspx sayfamızda Gonder değişkenine ait Asp.Net linkini tıkladığımızdan dolayı aspnet değerini getirmiştir.
- 9) Şimdi ise Default2.aspx sayfasından bu değeri alarak buna göre makalelerimizin gelmesini sağlayacağız.
- 10) Default2.aspx sayfamıza bir tane Repeater kontrolü yerleştirelim. Default2.aspx'in Form\_Load olayına kodumuzu yazalım.

```
protected void Page_Load(object sender, EventArgs e)
{
    string gelenveri=Request.QueryString["Gonder"];
    //Gelen GONDER değişkenini gelenveri ye aktar
    string yol = "Data Source=Selcuk\\SqlExpress;"
    + "initial Catalog=Ogrenci;integrated security=sspi";

    SqlConnection baglan = new SqlConnection(yol);
    string sorgu = "select * from tblMakale where kaydurum='"+gelenveri+"'";
    //Kategori tablosunun tüm alanlarını getir.
    SqlDataAdapter sadp = new SqlDataAdapter(sorgu, baglan);
    DataSet ds = new DataSet();
    sadp.Fill(ds);

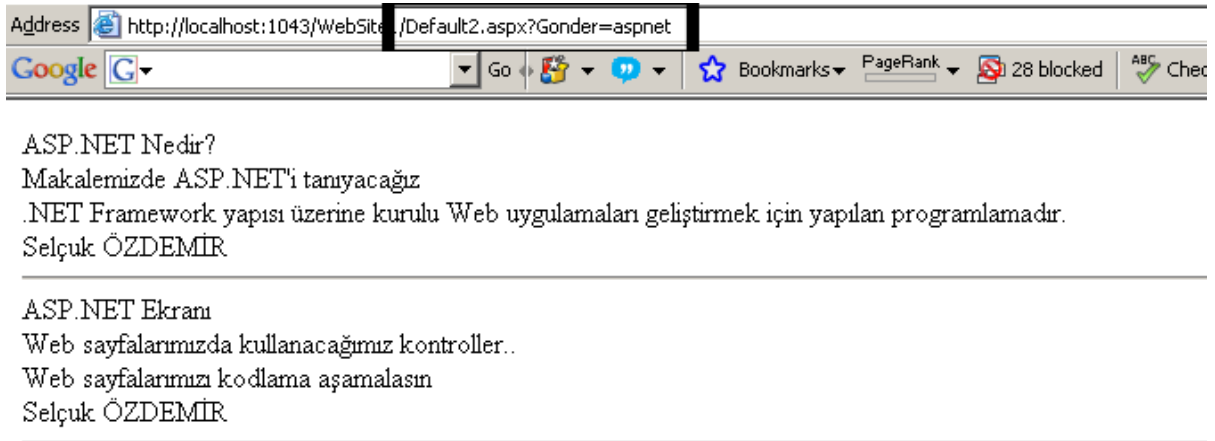
    this.Repeater1.DataSource = ds.Tables[0];
    this.Repeater1.DataBind();
}
```

- 11) Kodumuzu yazdıktan sonra datalarımızın gelmesi DataList nesnemize benzemektedir. Bunun için Source kısmına gelerek kodu yazınız (Şekil 4.26).

```
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Repeater ID="Repeater1" runat="server">
                <ItemTemplate>
                    <tr>
                        <td bgcolor="#33ccff">
                            <%#DataBinder.Eval(Container.DataItem,"Baslik") %><br />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <%#DataBinder.Eval(Container.DataItem,"Tanitim") %><br />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <%#DataBinder.Eval(Container.DataItem,"Detay") %><br />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <%#DataBinder.Eval(Container.DataItem,"Yazar") %><br /><hr />
                        </td>
                    </tr>
                </ItemTemplate>
            </asp:Repeater>
        </div>
    </form>
</body>
```

Şekil 4.26

12) Default.aspx sayfamızı çalıştırarak kategorilerden birini tıkladığımızda tblMakale den tıklanan linke göre makaleler listelenecektir (Şekil 4.27).



Şekil 4.27

Bu uygulama sonucunda data üzerinde daha esnek kodlama söz konusu olacaktır.

## REPORTVIEWER

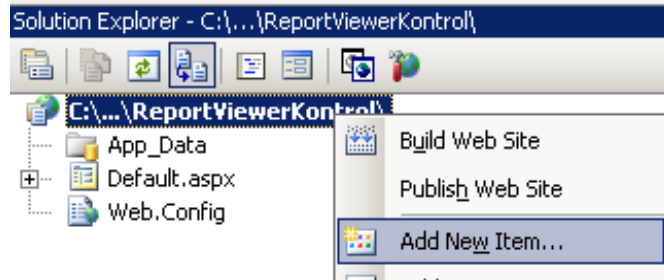
Raporlarımızı görüntülemek için kullanılan bir kontroldür. Aslen raporlamada Crystal Report veya Reporting Services kullanmak profesyonel bir raporlama yapmak için gereklidir. Reportviewer bu kontrollerin daha basitleştirilmiş halidir (Şekil 4.28).



Şekil 4.28

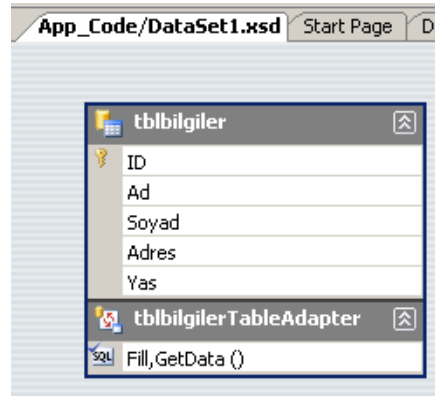
## UYGULAMA 6

1) Yeni bir Web sayfası açınız. Solution Explorerda bulunan projemizin ismini sağ tıklayarak Add New Item seçeneğini tıklayınız (Şekil 4.29).



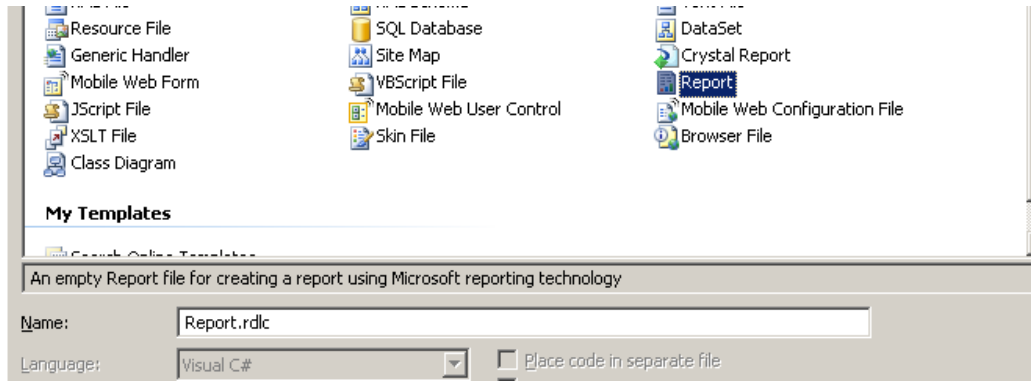
Şekil 4.29

2) Karşımıza gelen Add New Item penceresinden DataSet objesini seçerek Add butonuna basınız. Karşımıza gelen uyarı penceresinde Yes diyerek geçelim. Solution explorerda App\_Code klasörü oluşacak ve içerisine DataSet1.xsd dosyasını eklendiğini göreceksiniz. Eklendikten sonra karşımıza bağlantı sihirbazı gelecektir. Sql Server dan Öğrenci veritabanını seçerek bağlantımızı sağlayalım. Tablolardan tblBilgiler'i seçiniz. Tablo ekledikten sonra karşımıza tablonun kolonları gelecektir (Şekil 4.30).



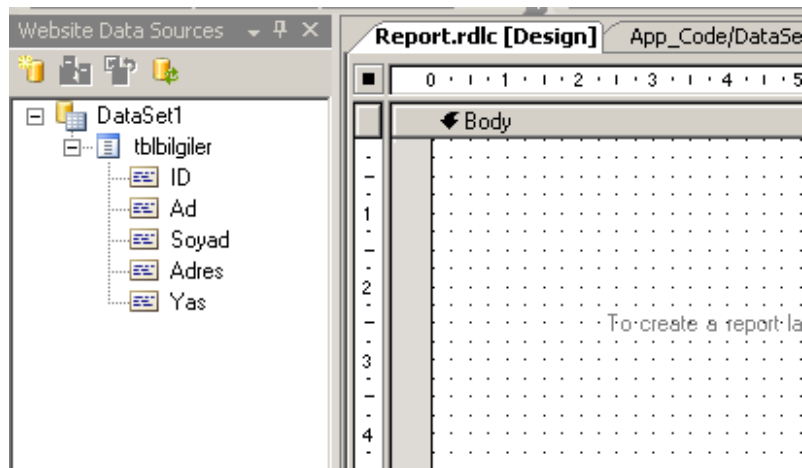
Şekil 4.30

3) Yine projemizin üstünde sağ tıklayarak bu sefer Report nesnesini ekleyelim (Şekil 4.31).



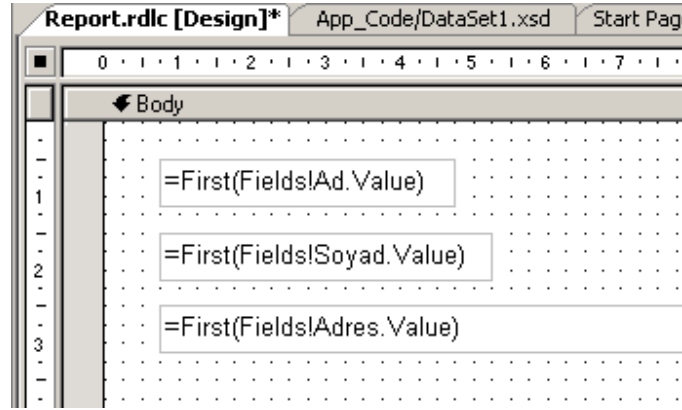
Şekil 4.31

4) Report nesnemizi ekledikten sonra karşımıza Report Designı gelecektir. Kolonlarımızı DataSource kısmında görmemiz gerekiyor (Şekil 4.32).



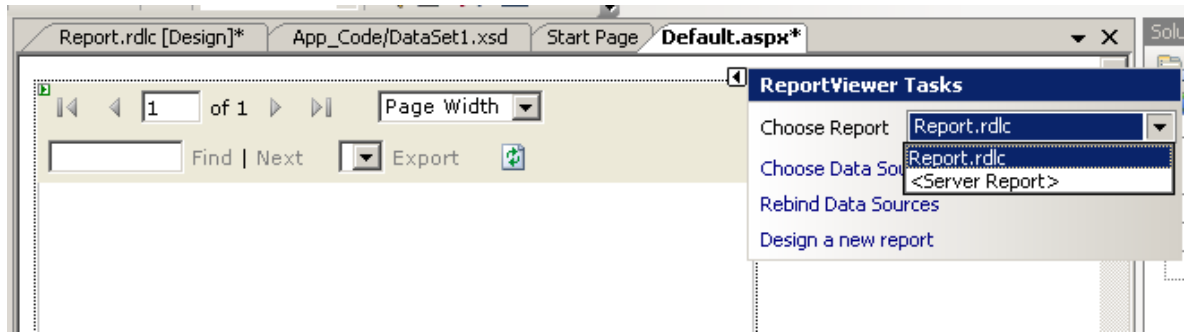
Şekil 4.32

5) Kolonlardan Ad, Soyad ve Adres kolonlarını sürükleyerek tasarım alanına koyunuz (Şekil 4.33).



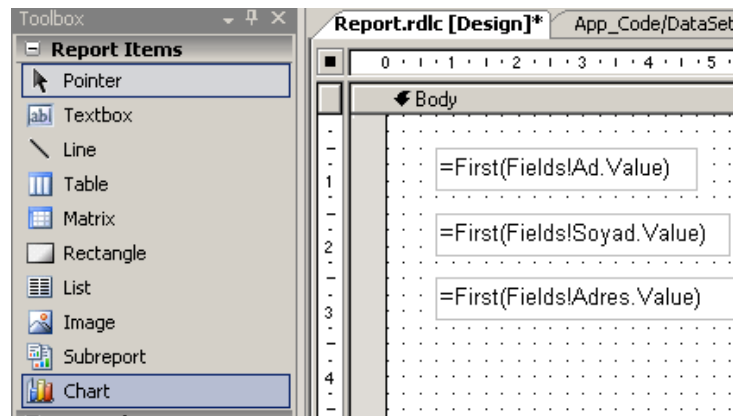
Şekil 4.33

6) Default.aspx sayfasına ReportViewer kontrolü ekleyiniz. Chose Report kısmından Report.rdlc dosyanızı seçiniz (Şekil 4.34).



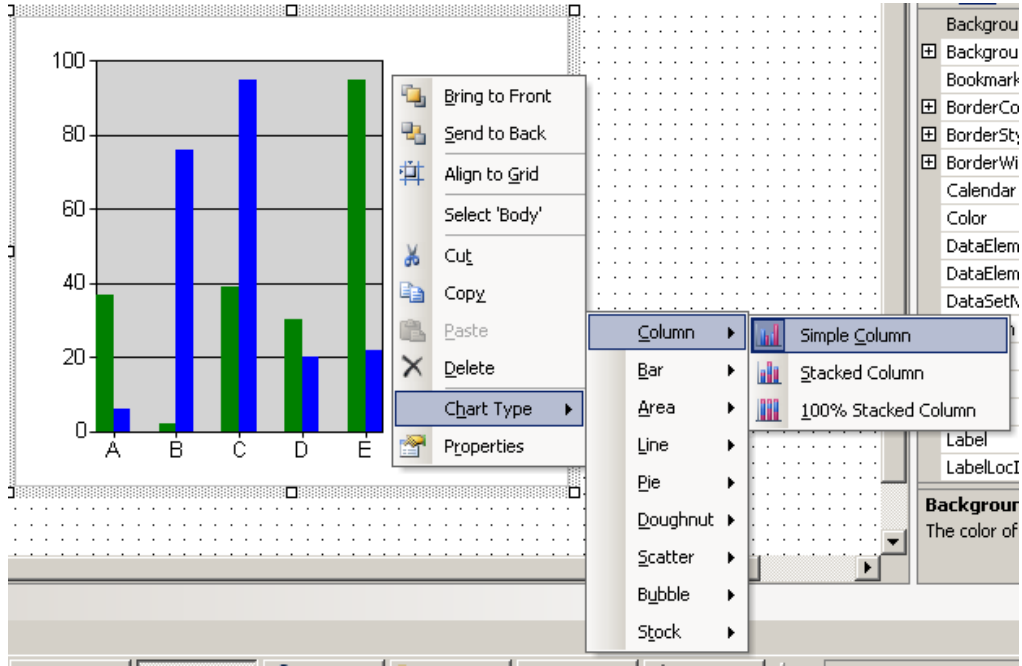
Şekil 4.34

7) Kodumuzu çalıştırdığımızda kaydın geldiğini görebiliriz. Report tasarım alanına gelerek tasarım alanına chart (grafik) ekleyelim (Şekil 4.35).



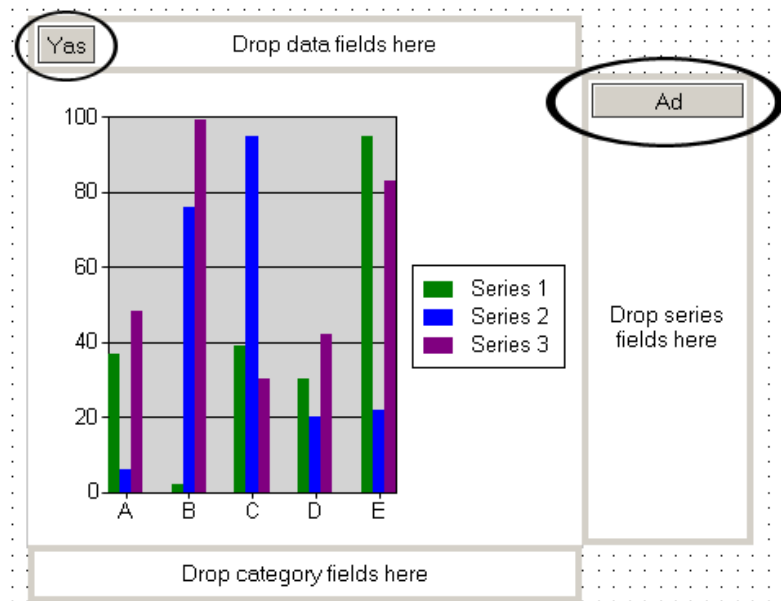
Şekil 4.35

8) Grafik tipini değiştirerek istediğiniz görüntüyü elde edebilirsiniz. Tasarım alanına eklenin grafik'i sağ tıklayarak Chart type'ı belirleyiniz (Şekil 4.36). Biz örneğimizde Simple Column seçeneğinden göstereceğiz.



Şekil 4.36

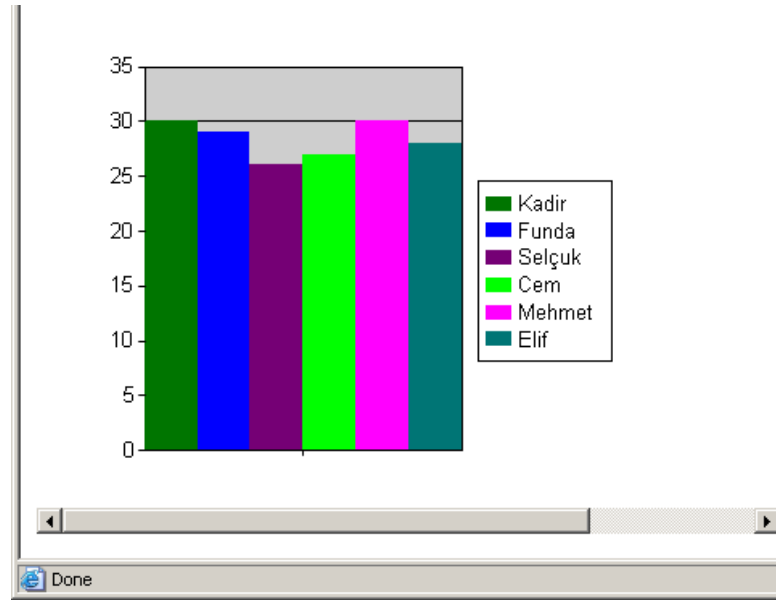
9) Grafiğimizi bir kez tıkladığımızda grafik'in kenarlarında alanlar oluşur. Bu alanlar üst tarafta bulunan alan önemlidir. Burası gelen datalarımızın grafik şeklinde oluşmasını sağlar. Sağ taraf ise hangi rengin kime it olduğunu gösterecek. Ad ve yas data kolonlarını sürükleyip bırakınız. (Şekil 4.37).



Şekil 4.37

10) Default.aspx sayfamızı çalıştırarak grafiğimizin oluşması sağlanacaktır (Şekil 4.38).





Şekil 4.38

Böylelikle basit bir düzeyde raporlamada ReportViewer'ı kullanabiliriz. Daha profesyonel raporlamalar için CrystalReport kullanılacaktır. Crystal Report'u sonraki konularda anlatacağım.

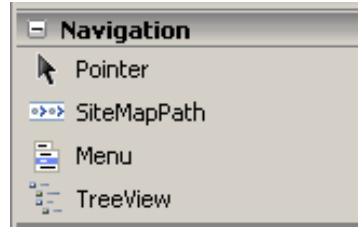
# KONU 5

# NAVIGATION



## NAVIGATION KONTROLLER

Asp.Net 2005’de yeni kontrollerden olan navigation kontroller web ortamında kullanılan birçok işlemlerimizi daha rahat yapmamızı sağlamaktadır (Şekil 5.1).



Şekil 5.1

## SITEMAPPATH

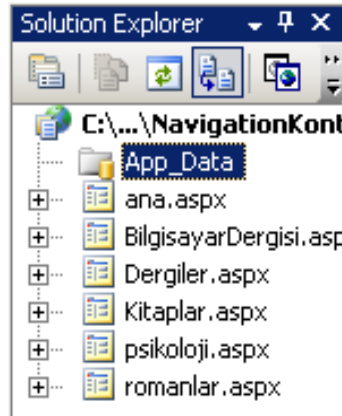
Bir web sayfamıza ait bilgilerin hiyerarşik bir düzen göstermemiz gerektiğinde kullanılır (Şekil 5.2).

[Root Node](#) > [Parent Node](#) > Current Node

Şekil 5.2

## UYGULAMA 1

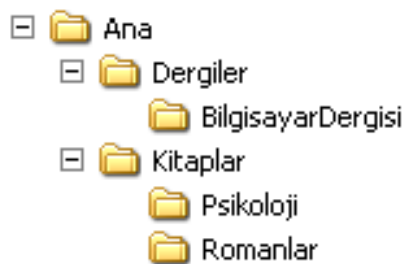
1) Bir web sayfası açalım ve Add New Item dan diğer sayfalarımızı hazırlayalım. Sayfalarımızın adları ana.aspx, Dergiler.aspx, Kitaplar.aspx, bilgisayarDergisi.aspx, romanlar.aspx, psikoloji.aspx olarak oluşturalım (Şekil 5.3).



Şekil 5.3

2) Dosyalarımızı oluşturduktan sonra Project > Add New Item diyerek Site Map nesnesini seçerek Web.sitemap dosyasını oluşturunuz.

3) Web.sitemap dosyamızı aşağıdaki gibi düzeltmeden önce isterseniz nasıl bir yapımızın olduğunu klasör mantığı ile görelim (Şekil 5.4).



Şekil 5.4

4) Birimlerin şematik hali anlaşıldığına göre Web.sitemap dosyamızın kod kısımlarını değiştirelim.

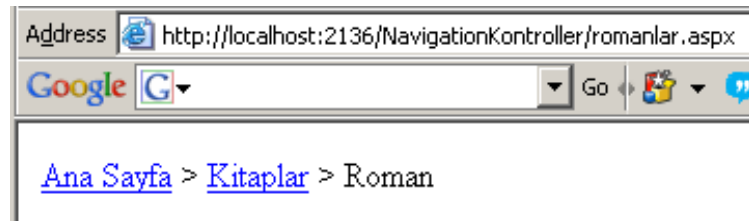
```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="Ana.aspx" title="Ana Sayfa">

    <siteMapNode url="Dergiler.aspx" title="Dergiler">
      <siteMapNode url="Bilgisayardergisi.aspx" title="Bilgisayar"/>
    </siteMapNode>

    <siteMapNode url="Kitaplar.aspx" title="Kitaplar">
      <siteMapNode url="romanlar.aspx" title="Roman"/>
      <siteMapNode url="psikoloji.aspx" title="Psikoloji"/>
    </siteMapNode>

  </siteMapNode>
</siteMap>
```

5) Kodumuzu yazdıktan sonra her sayfaya SiteMapPath kontrolü ekleyiniz. Otomatik olarak sayfa isimleri eklenecektir. Böylece yaptığımız ağaç sistemi oluşacaktır (Şekil 5.5).



Şekil 5.5

## MENU

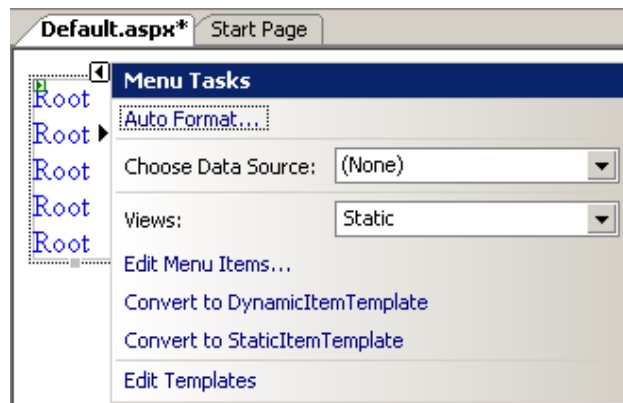
Web sayfalarında sıkça kullanılan açılır menu tanıımıyla bildiğimiz menu artık her web sayfasında yer almaktadır. Asp.Net 2.0 bize menu nesnesini kullanmamızı sağlamıştır (Şekil 5.6).



Şekil 5.6

## UYGULAMA 1

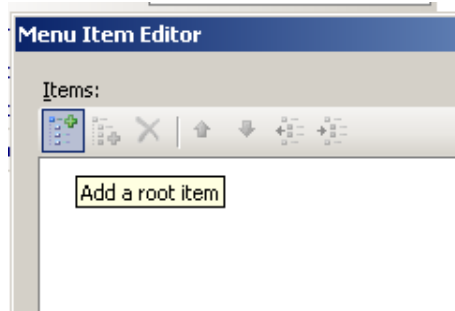
1) Yeni bir web sayfası açalım. Web sayfamıza bir adet Menu kontrolü ekleyelim (Şekil 5.7).



Şekil 5.7

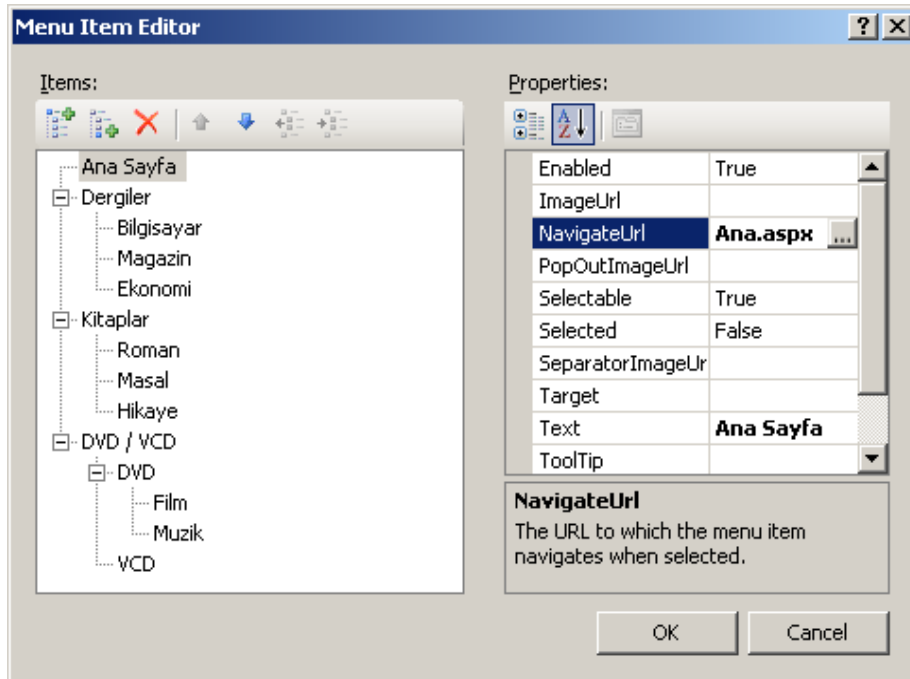
2) Menu ekledikten sonra karşımıza gelen Menu Tasks menüsünden Edit Menu Items seçeneğini tıkladığınızda Menu Item Editor açılacaktır.

- 3) Açılan pencereden Add a Root Item düğmesini tıklayarak menu hazırlamaya başlayabiliriz (Şekil 5.8).



Şekil 5.8

- 4) Düğmeye tıkladıktan sonra menu için ana birim ve alt birimleri oluşturalım (Şekil 5.9).



Şekil 5.9

- 5) Her bir menünün özellikleri yer almaktadır. Menülerden birisini tıklayıp properties penceresinden ayarlamalar yapabiliriz.

| Özellik Adı | Tanımı  |
|-------------|---|
| Text        | Menu Yazısı                                   |
| ImageUrl    | Menu Yazılarının Sol Tarafına Resim Koyar     |
| NavigateUrl | Link Tıklandığında Gidecek Olan Sayfa / Adres |
| Tool Tip    | Tool Tip Yazısını Ekler                       |
| Target      | Framelerde Kullanılan Hedef Pencere Adı       |
| Enable      | False ise seçilemez alt seçenekleri gelmez    |

- 6) Kodumuzu çalıştırdığımızda menümüzü kullanabiliriz.

KONU 6

RAPORLAMA

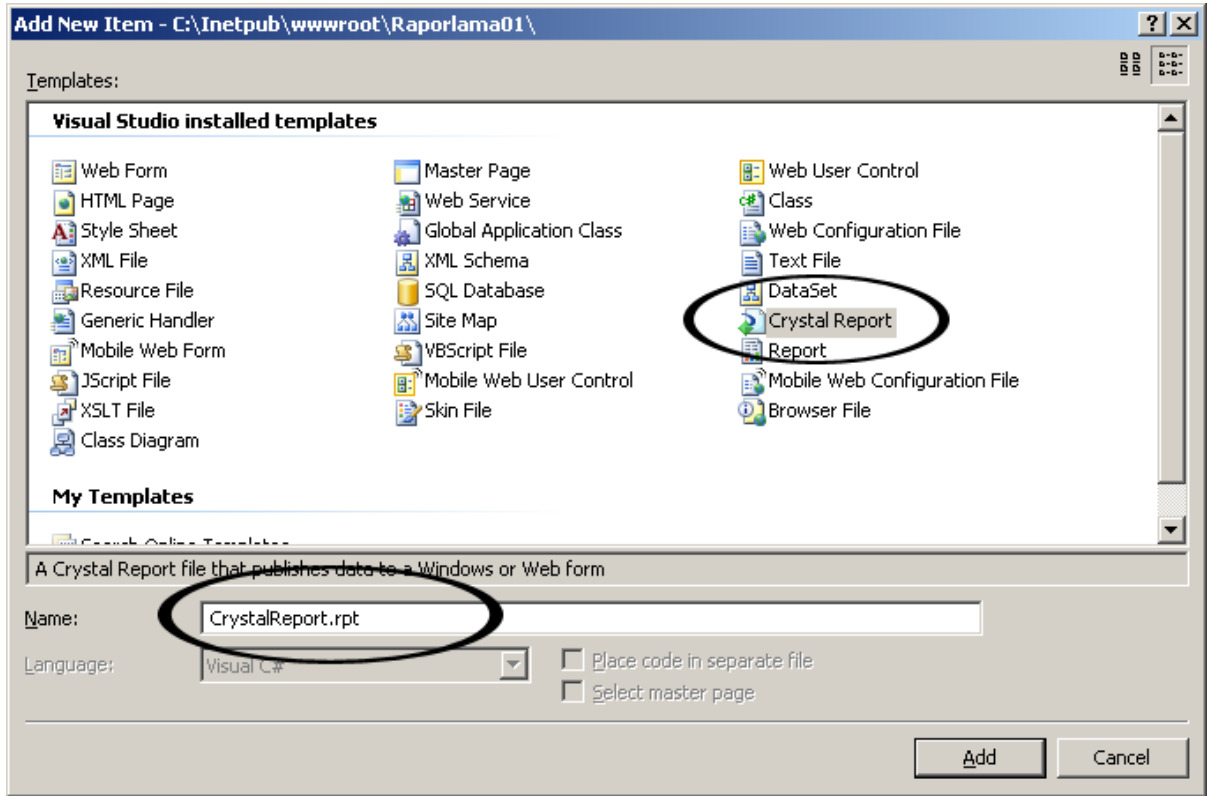
## RAPORLAMA

Raporlama her yazılımda olduğu gibi web ortamındada gerekli olan bir durumdur. Rapor hem kağıt ortamına hemde isterseniz diğer ortamlara taşımak olarak kaydetmektir. Raporlamada en iyi araçlardan biri olan Visual Studio .NET'i kurduktan sonra içerisinde gelen Crystal Report nesnesini anlatacağım.

Yeni bir web sayfası açtıktan sonra Web Site > Add New Item seçeneğinden Crystal Report'u projemize ekleyebiliriz. Crystal Report'un dosya uzantısı rpt türünde görülecektir.

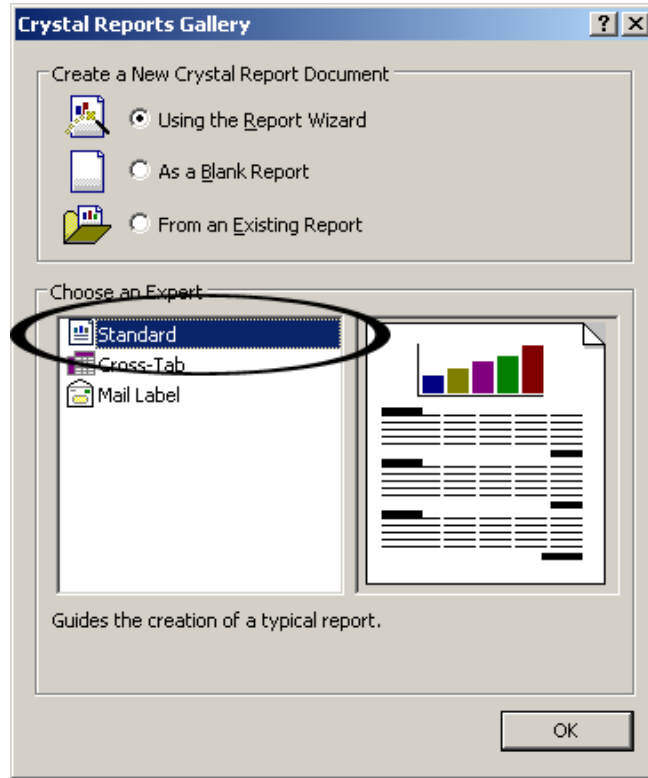
### UYGULAMA 1

1) Bir web sayfası açalım. Daha sonra Web Site > Add New Item seçeneğini tıklayınız. Karşımıza gelen pencereden Crystal Report nesnesini seçtikten sonra Add butonuna basınız (Şekil 6.1).



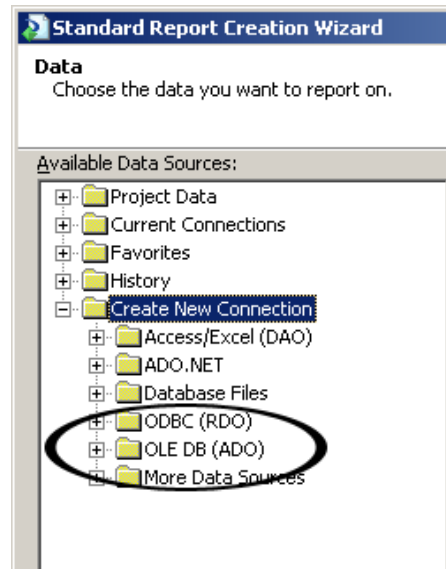
Şekil 6.1

2) Add butonuna bastıktan sonra karşımıza Crystal Report Gallery penceresi gelecektir (Şekil 6.2). Pencereden Standart kağıt düzenini seçerek OK butonuna basınız.



Şekil 6.2

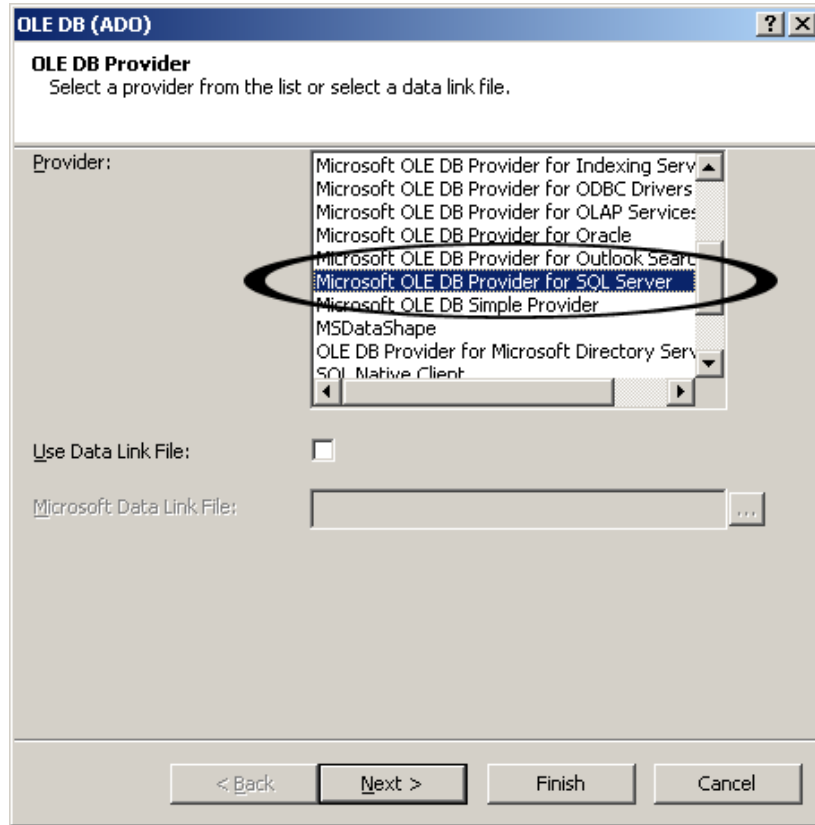
3) Daha sonra karşımıza veritabanımıza bağlantı sağlamak için Standart Report Creation Wizard penceresi gelecektir. Biz uygulamamızda SQL Server 2005 veritabanımıza bağlanacağız. Bunu için açılan pencereden Create New Connection kısmından OLE DB seçeneğini seçiniz (Şekil 6.3).



Şekil 6.3

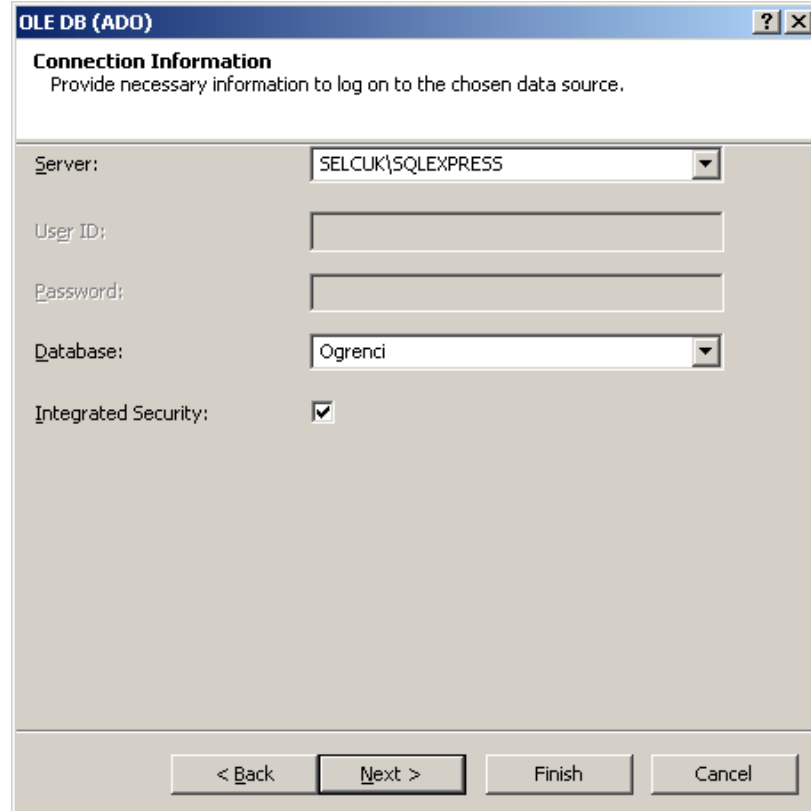
4) OLE DB seçeneği tıkladığımızda karşımıza veritabanı bağlantımız için OLE DB (ADO) penceresi gelecektir. Buradan Microsoft OLE DB Provider for SQL Server seçeneğini seçerek Next düğmesine basınız (Şekil 6.4).





Şekil 6.4

5) Daha sonra karşımıza gelen pencereden SQL Server bağlantımızı gerçekleştirelim. Bu kısımda Sql Server veritabanımızı ve tablomuzu (ogrenci) seçelim (Şekil 6.5).



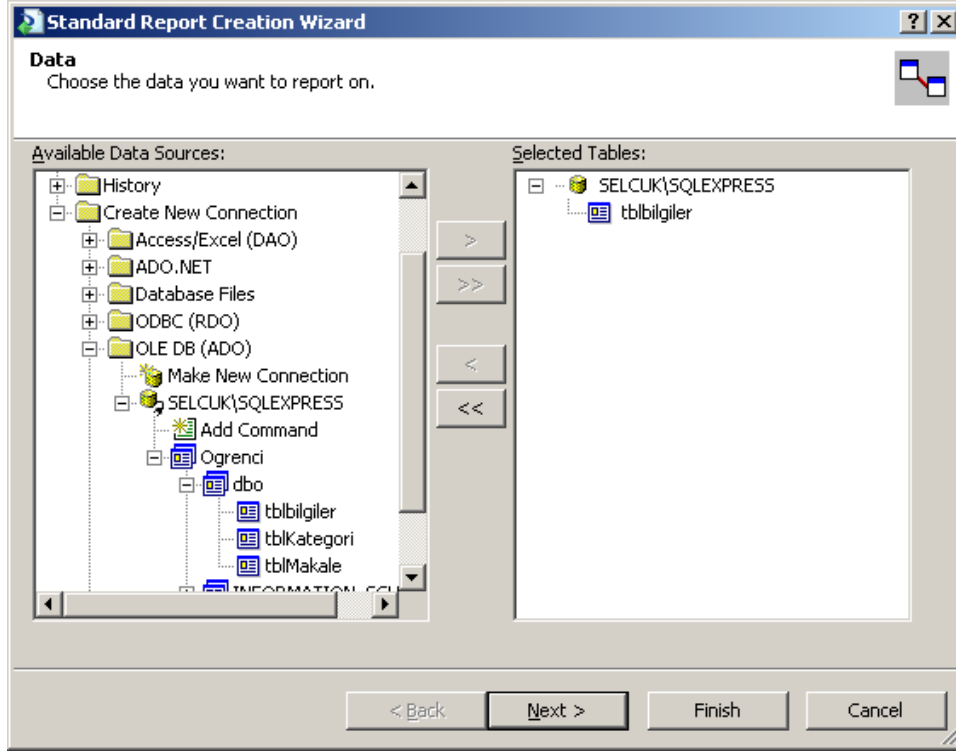
Şekil 6.5

6) Tablomuzu seçtikten sonra next butonuna bastıktan sonra karşımıza seçtiğimiz değerlere göre bilgiler gelecektir Finish butonuna basarak bağlantı işlemimizi bitirelim.



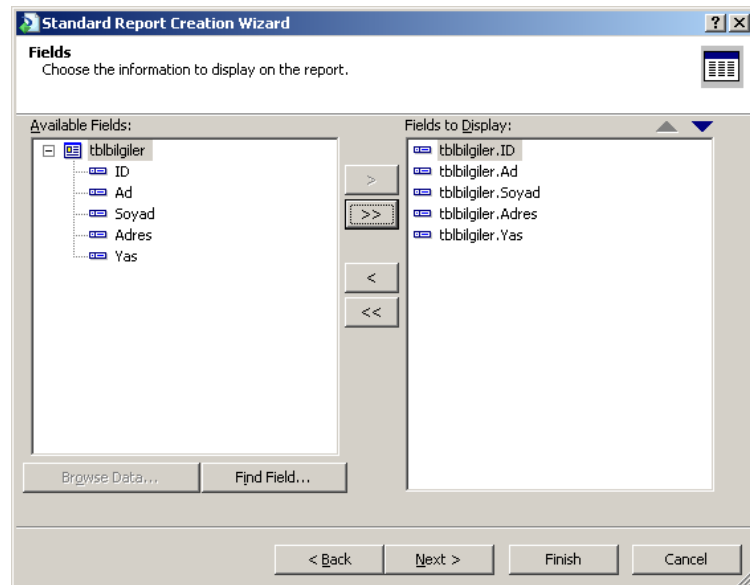
Bazı durumlarda Server'ı seçmek istediğimizde hatalar alabilir veya ismini göremeyebiliriz. Bu durumda direk olarak klavyeden Sql Server'ın tam adını girerek işleminize devam edebilirsiniz.

7) Veritabanımıza bağlandığımız göre artık işlemimize devam edebiliriz. Tekrardan Standart Report Creation Wizard penceresi görünecektir. Bu sefer OLE DB (ADO) nun altında bağlantı yaptığımız veritabanını görebiliriz. Burada Öğrenci veritabanının altında dbo bölümünden tblBilgiler tablosunu seçerek sağ tarafa aktaralım. Sağ tarafa (Selected Tabloes) çift tıklayarak geçirebiliriz (Şekil 6.6).



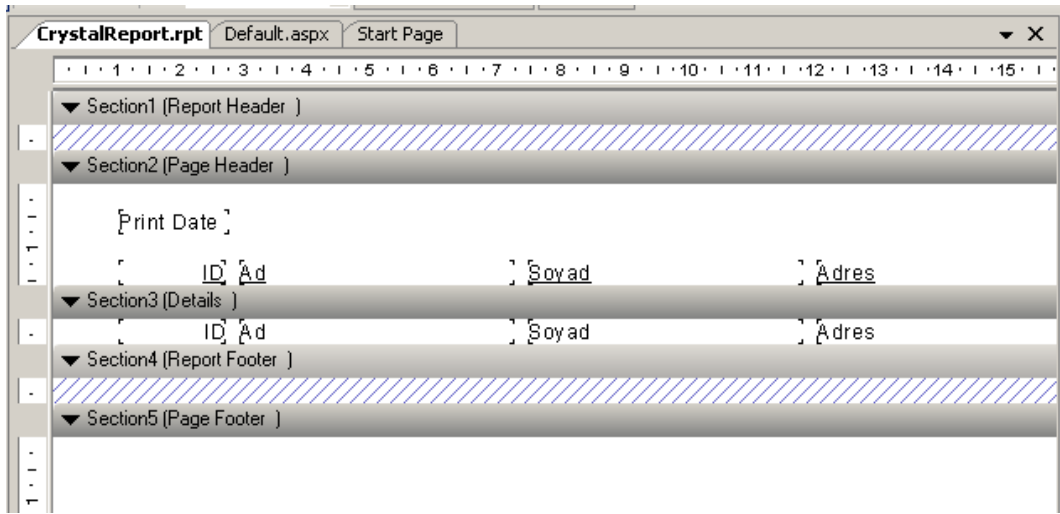
Şekil 6.6

8) Tablomuzu seçtikten sonra Next düğmesine basarak devam edelim. Karşımıza ekranda hangi kolonları görmek istediğimizi soracaktır. Biz tblBilgiler tablosundaki tüm kolonları yine seçerek sağ tarafa geçiriniz (Şekil 6.7).



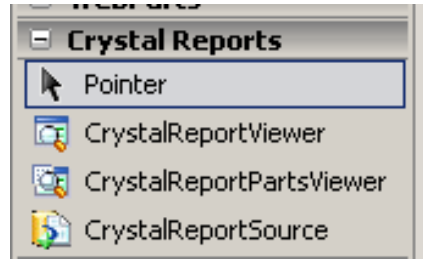
Şekil 6.7

9) Kolonları seçtikten sonra FINISH butonuna basarak işlemimizi sonlandıralım. Karşımıza CrystalReport.rpt dosyası açılacak ve seçtiğimiz kolonlar bir düzen içerisinde yerleşmiş olacaktır (Şekil 6.8).



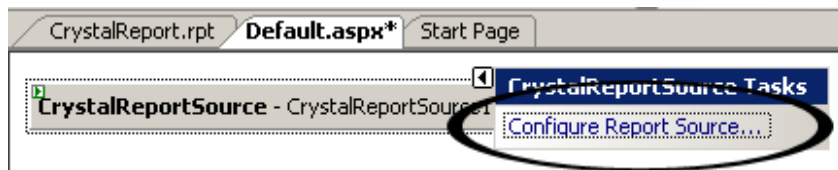
Şekil 6.8

10) Raporumuzu web sayfamızda görüntülemek için web sayfamıza ilk olarak CrystalReportSource nesnesini ekleyiniz (Şekil 6.9).



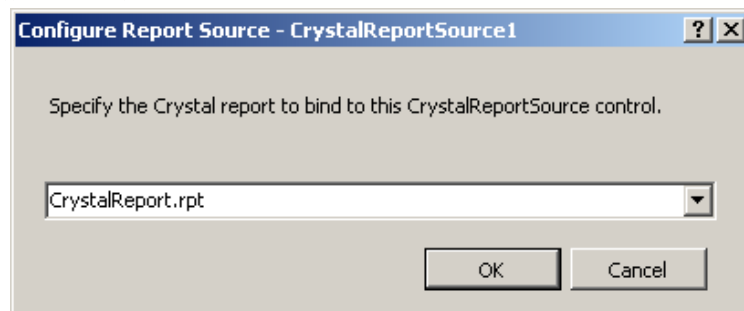
Şekil 6.9

11) Web sayfamıza eklediğimiz CrystalReportSource'u seçiniz. Configure Report Source seçeneğini tıklayınız (Şekil 6.10).



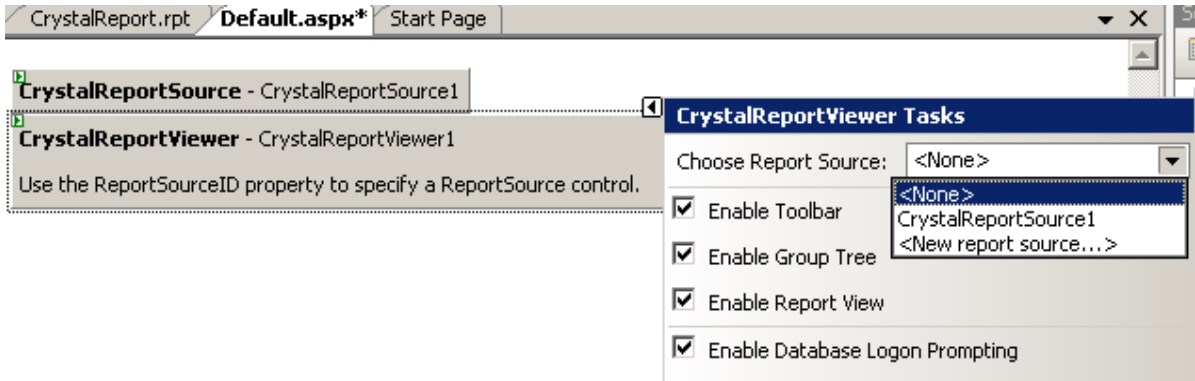
Şekil 6.10

12) Karşımıza gelen Configure Report Source penceresinden CrystalReport.rpt dosyamızı seçtikten sonra OK butonuna basınız (Şekil 6.11).



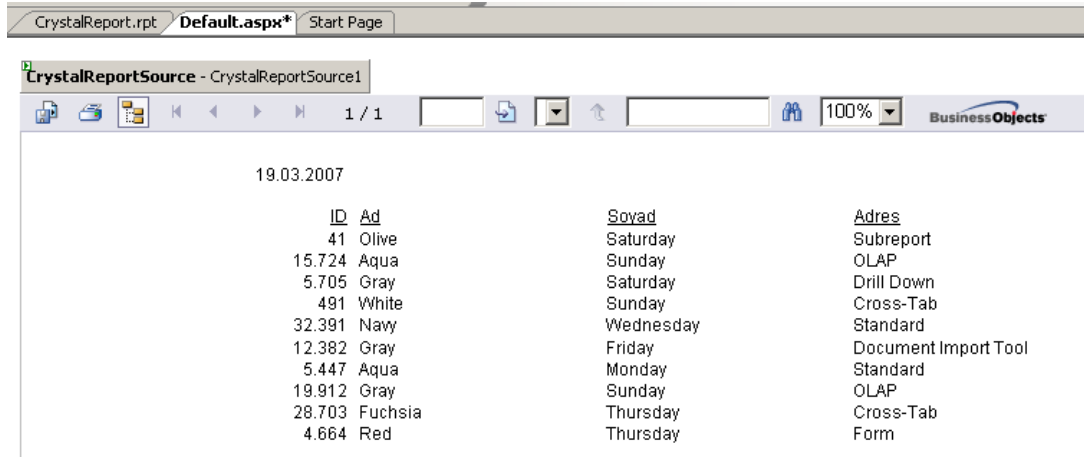
Şekil 6.11

13) Dosyamızı seçtikten sıra geldi içerisindeki bilgileri görüntülemek. Bunu için web sayfamıza CrystalReportViewer ekleyelim ve Choose Report Source seçeneğinden CrystalReportSource1'i seçiniz (Şekil 6.12).



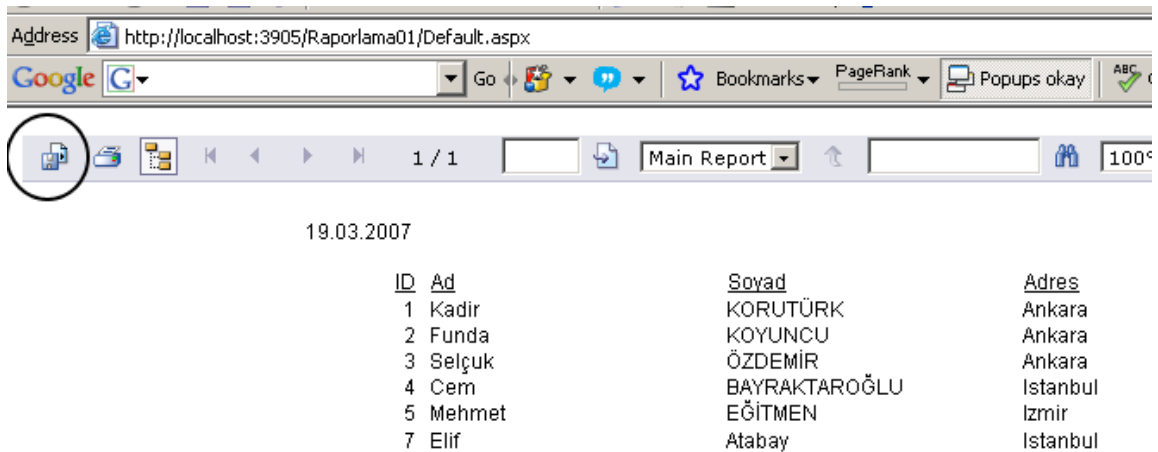
Şekil 6.12

14) Raporumuzu seçtikten sonra CrystalReportViewer'ın görüntüsü değişecektir. Burada bizim web sayfamızı çalıştırdığımızda nasıl bir görüntü olacağına dair bize bilgi vermektedir (Şekil 6.13).



Şekil 6.13

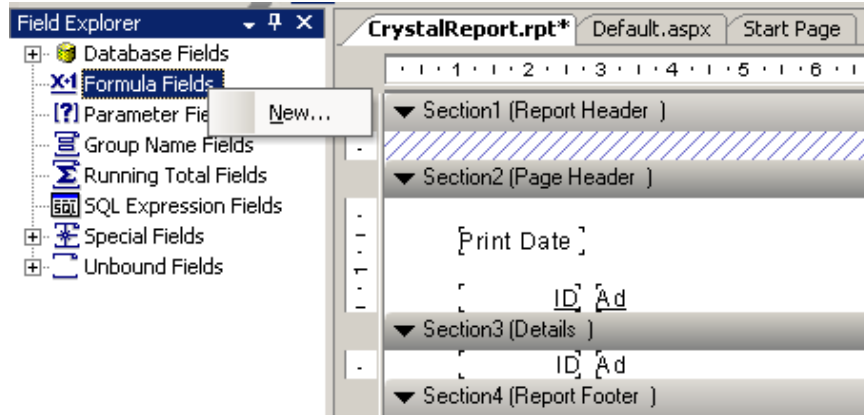
15) Artık web sayfamızı çalıştırarak raporumuzu görebiliriz. Burada printer ikonunu tıkladığımızda bilgiler yazıcıdan çıkacaktır. Buna ek olarak raporumuzun üstteki başlıkta bulunan ilk buton olan EXPORT butonuna basarak excel, word, pdf formatında raporumuzu alabiliriz (6.14).



Şekil 6.14

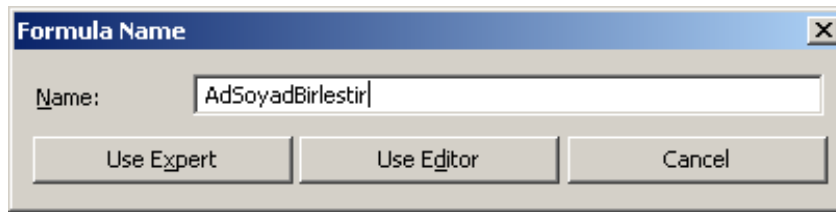
## UYGULAMA 2

1) Bir önceki uygulamanın devamı olacağından önceki raporlama projemizi açalım. Bu uygulamamızda Crystal Report'un bir kaç özelliğinden bahsedeceğim. Bunu için Crystal Report tasarım alanına geçelim. Burada bulunan Field Explorer kısmında Formule Fields kısmını sağ tıklayarak New seçeneğini seçiniz (Şekil 6.15).



Şekil 6.15

2) New dedikten sonra karşımıza Formule Name kısmı açılacaktır. Burada yapacağımız formül ismini istemektir. Bu kısma bir isim yazarak Use Editor butonuna basınız (Şekil 6.16).



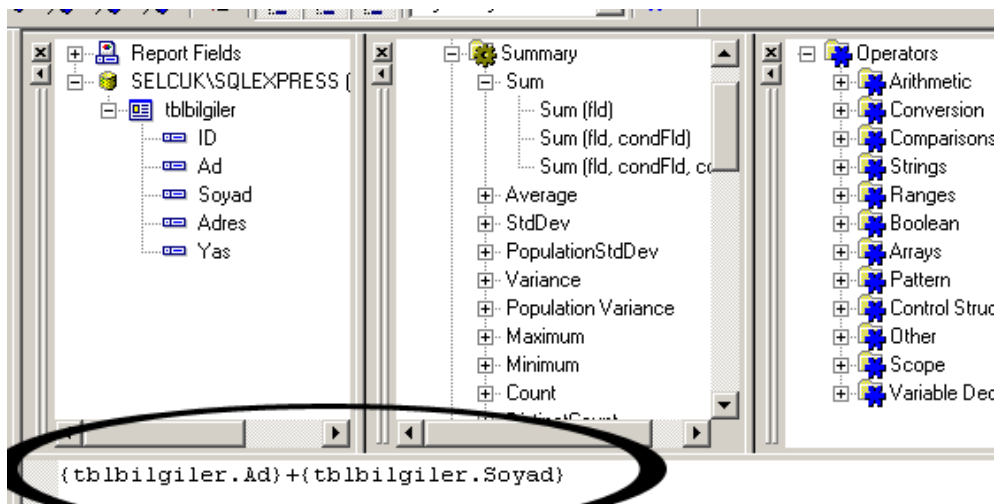
Şekil 6.16

3) Use editor kısmında field ve formulleri görebiliriz. Biz bu formülümüzde Ad ve Soyad kolonlarını birleştirerek tek birim olarak kullanmamızı sağlayacaktır.

4) Biz örneğimizde Ad ve Soyad kolonlarını birleştirerek tek parametrede getireceğiz. Bunu için formül alanına (Şekil 6.17);

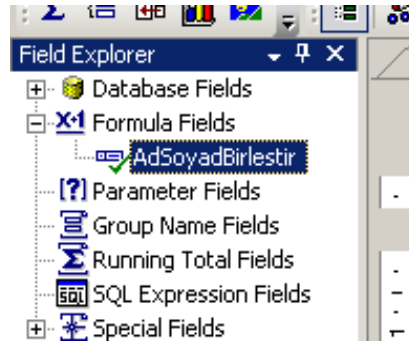
{tblbilgiler.Ad}+{tblbilgiler.Soyad}

Kodunu yazarak penceremizi kapatalım.



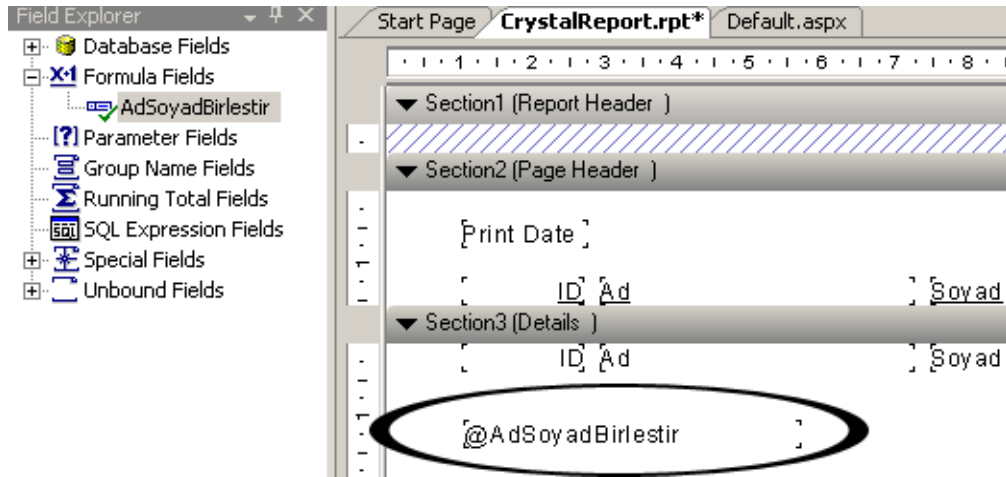
Şekil 6.17

5) Field Explorerda bulunan Formule Fields'ın altında oluşturduğumuz değeri görebiliriz (Şekil 6.18).



Şekil 6.18

6) Oluşturduğumuz formülümüzü rapor tasarım alanına sürükleyiniz (Şekil 6.19).

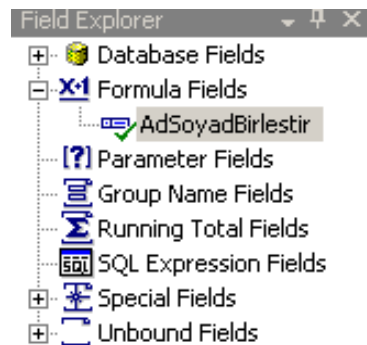


Şekil 6.19

7) Kodumuzu çalıştırdığımızda Ad ve Soyad ekrana gelecektir. Böylelikle tek tek kolonlarlar uğraşmaktanda belli ayrılmayan birimleri bu şekilde sunabiliriz.

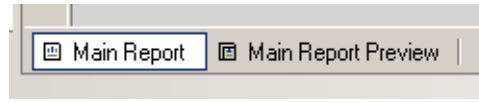
Crystal Report tabikide bu kadar basit işlemlerle kalmıyor. Crystal Repor'un detaylarına girmeden önce ekranımızı tanıyalım. Böylece pencereleri isimleriyle belirttiğimizde nereden bahsettiğimizi anlamış olacağız.

En çok kullanılan penceremiz olan Field Explorer kısmı, bütün işlemlerimizi yapmamızı sağlayan bir yer olarak tanımlanabilir (Şekil 6.20) .



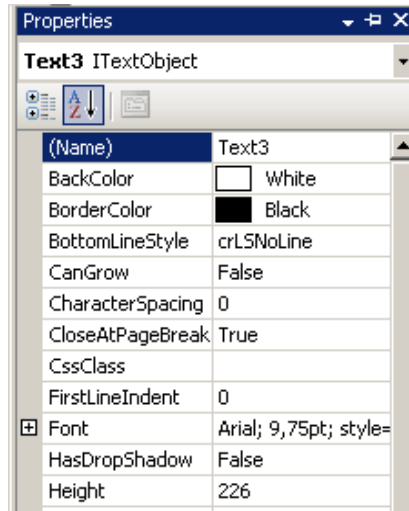
Şekil 6.20

Raporlamamızı hazırladıktan sonra dataların nasıl görüntülendiğini görmek için bir önizleme yapabiliriz. Bu işlemde tasarım alanının altında bulunan Main Report Preview bölümüne girerek görebiliriz (Şekil 6.21).



Şekil 6.21

Böylelikle web sayfalarına bağlamadan son durumunu görebilmekteyiz. Bunların dışında tasarım alanına yerleşen her objenin özelliklerini değiştirebileceğimiz Properties penceresi yer almaktadır. ASP.NET'in girişinde de anlattığımız Properties penceresinden bir farkı yoktur (Şekil 6.22).



Şekil 6.22

Şimdi size Crystal Report'dan biraz bahsetmek istiyorum. Raporlama konusunda birçok durumu burada çözerek işlemlerinizin daha kolay olduğu göreceksiniz.

#### ÖRNEK 1

Field Explorer'dan, Formule Fields kısmından Kosul adında bir formül sistemi oluşturalım. Burada yapacağımız örnek yaş kolonunu kontrol etmek olacaktır. Eğer kişinin yaşı 28 ve daha küçük ise ekranda sadece adı yazacak. Büyük ise adını ve yaşını yazmasını sağlayacağız. Bunun için kod alanına aşağıdaki kodu yazıp Kosul isimli formülümüzü tasarım alanına ekleyelim. Kodumuzu çalıştırdığımızda 28 ve daha küçük yaşı olanların sadece ad kolonları görünecektir (Şekil 6.23).

```
if {tblBilgiler.Yas} <=28 Then
    {tblBilgiler.Ad}
Else
    {tblBilgiler.Ad} & " , " & {tblBilgiler.yas}
```

Kodlama Visual Basic editörü olduğundan C# dan farklı bir kodlama söz konusu olacaktır. Burada eğer tblBilgiler.Yas kolonundaki değer 28 eşit yada küçük ise sadece tblBilgiler.Ad yazılacak. Büyük ise Ad ve Yas kolonları yazılacaktır.

19.03.2007

| Kosul | ID     | Ad            | Soyad |
|-------|--------|---------------|-------|
| 1     | Kadir  | KORUTÜRK      |       |
|       |        | Kadir         | 30,00 |
| 2     | Funda  | KOYUNCU       |       |
|       |        | Funda         | 29,00 |
| 3     | Selçuk | ÖZDEMİR       |       |
|       |        | Selçuk        | 26,00 |
| 4     | Cem    | BAYRAKTAROĞLU |       |
|       |        | Cem           |       |
| 5     | Mehmet | EĞİTMEN       |       |
|       |        | Mehmet        | 30,00 |

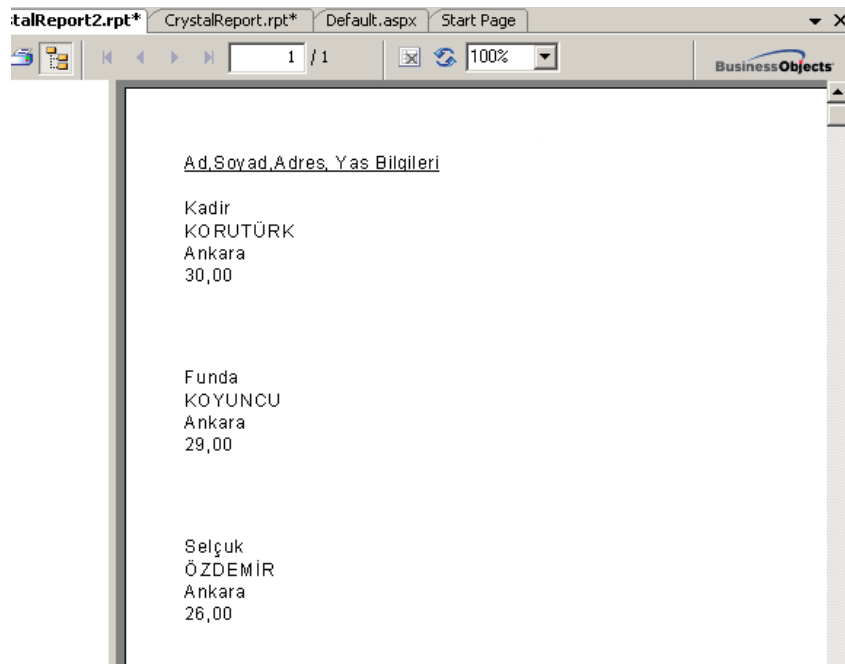
Şekil 6.23

## ÖRNEK 2

Bu örneğimizde tüm kolonlarımızı tek bir formül içerisinde toplayarak raporlamada daha düzgün bir görüntü sağlayabiliriz. Birleştire adında bir formül yaratalım ve kodumuzu yazıp çalıştıralım.

```
{tblbilgiler.Ad} & chr(13) &
{tblbilgiler.Soyad} & chr(13) &
{tblbilgiler.Adres} & chr(13) &
{tblbilgiler.Yas}
```

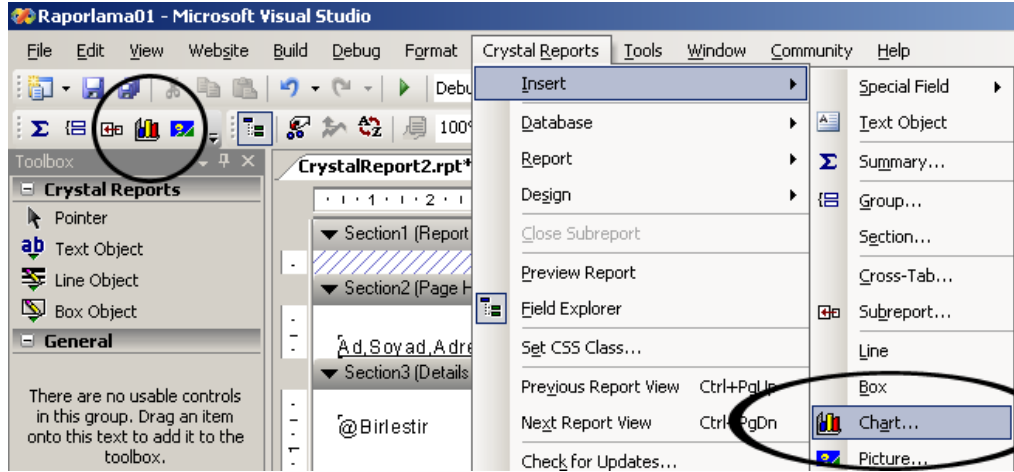
Kodumuzu çalıştırdığımızda tüm bilgiler gelecektir. Kodlamada chr(13) kısmı ascii karakterde Enter yani alt satıra geçmek için kullanılır (Şekil 6.24).



Şekil 6.24

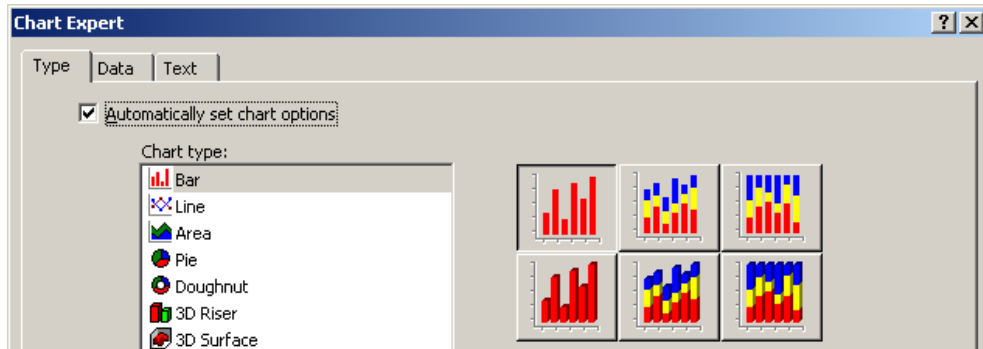


Raporlamada en büyük ihtiyaçlarımızdan biriside raporlama esnasında grafik kullanmaktır. Değerlerimize göre kağıda grafik aktarmak için Chart objesini kullanmamız yeterlidir. Bunu için bir yaş grafiği çıkartalım. Chart objesini üst menüden Cystal Reports > Insert > Chart seçeneğini tıklayabilir yada araç çubuğundan Chart objesine tıklayabiliriz (Şekil 6.25).



Şekil 6.25

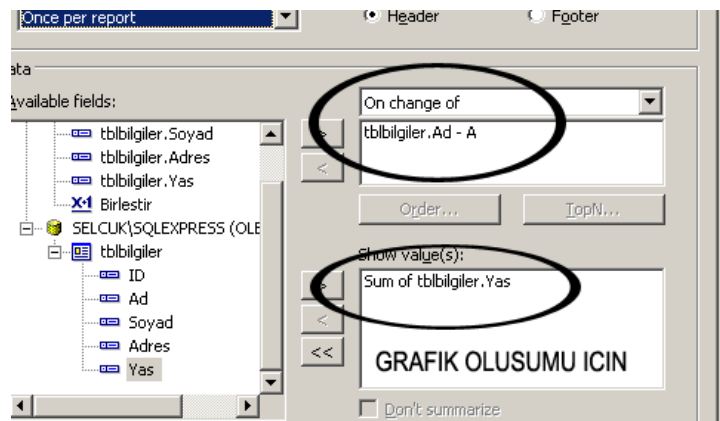
Chart objesini tıkladığımızda karşımıza Chart Expert penceresi gelecektir. Burada grafik türünü veritipimize uygun olarak seçebiliriz (Şekil 6.26). Biz örneğimizde Bar türünü kullanacağız.



Şekil 6.26

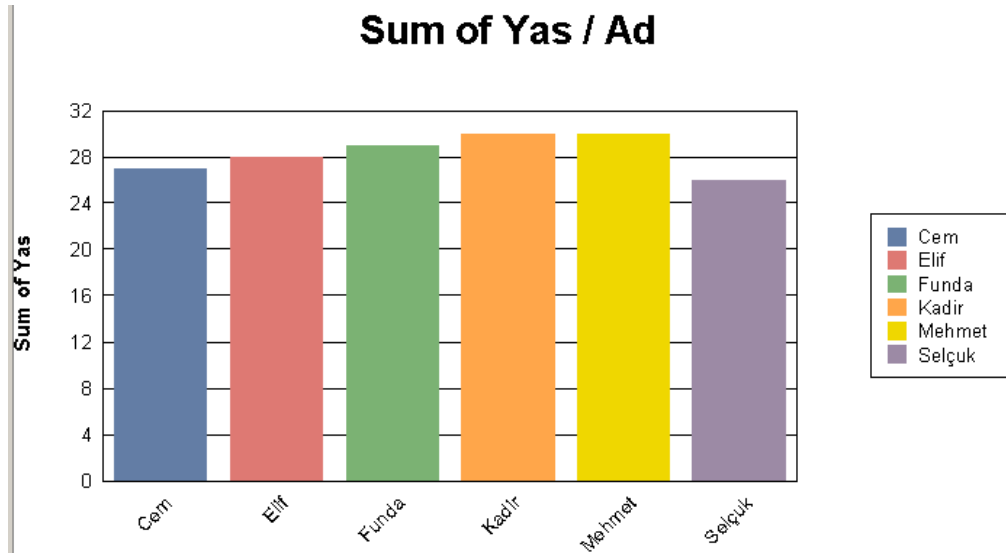
### ÖRNEK 3

Bar türünü seçtikten sonra Data tab alanına girerek kolonlarımızı seçeceğiz. Data kısımda Show Values kısmına Yas kolonunu aktaralım (Şekil 6.27).



Şekil 6.27

Yas kolonuna göre grafiğimi oluşturmuştur. OK butonuna basarak grafik işlemimizi sonlandıralım. Böylelikle kişilerin yaş grafiği çıkmıştır (Şekil 6.28).



Şekil 6.28

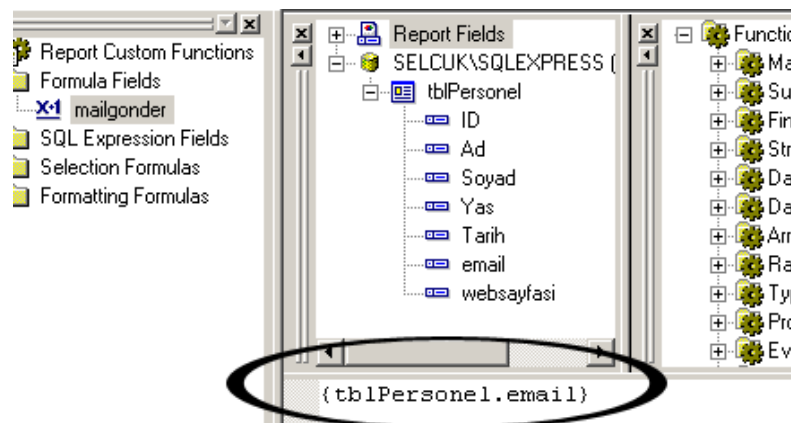
Şimdi bazı rapor detaylarımızı anlatabilmemiz için Öğrenci veritabanımıza tblPersonel adında bir bir tablo oluşturacağız .

| Kolon Adı  | Veritipi       | Açıklama (Örnek Değer) |
|------------|----------------|------------------------|
| ID         | İnt            | PK                     |
| Ad         | nvarchar (50)  | Selçuk                 |
| Soyad      | nvarchar (50)  | Özdemir                |
| Yas        | İnt            | 26                     |
| Tarih      | DateTime       | 05.07.1981             |
| Email      | nvarchar (100) | selcukmct@gmail.com    |
| Websayfasi | nvarchar (50)  | www.google.com         |

Tablomuzu oluşturduktan sonra içerisine bilgiler giriniz. Buradaki örneğimiz kişinin mailine html işlemi gibi üstüne tıkladığımızda outlook programının açılmasını ve Gönder (to) kısmında o kişinin mail adresinin yazılmasını sağlamak veya web adresine gitmemiz gerekiyorsa bu işlemi kolayca yapabiliriz.

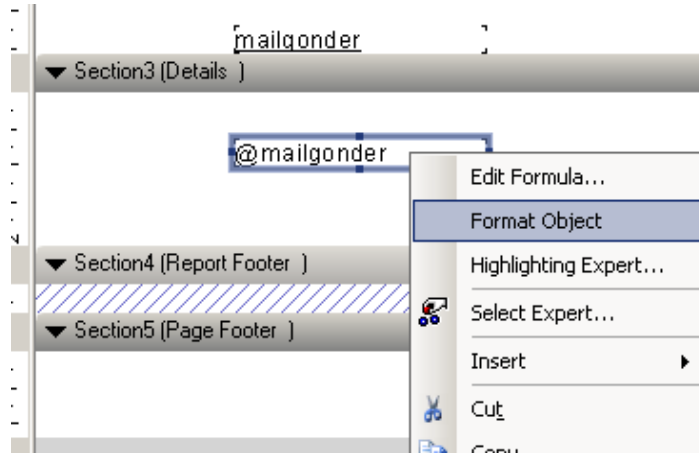
#### Örnek 4

Projemize yeni bir rapor dosyası ekleyiniz ve rapor dosyanızı yeni oluşturduğunuz tblPersonel tablosuna bağlayınız. Burada mailgonder adında yeni bir Formule Fields ekleyiniz. Buraya email kolonunu ekleyiniz (Şekil 6.29).



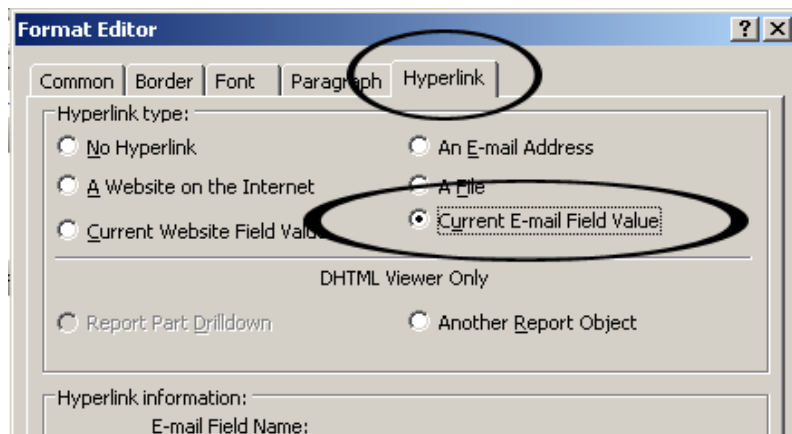
Şekil 6.29

Save and Close düğmesine bastıktan sonra pencremizi kapatalım. Daha sonra mailgonder formülünü rapor tasarım alanına koyalım. Eklenen formülün üstünü sağ tıklayarak Format Object alanına girelim (Şekil 6.30).



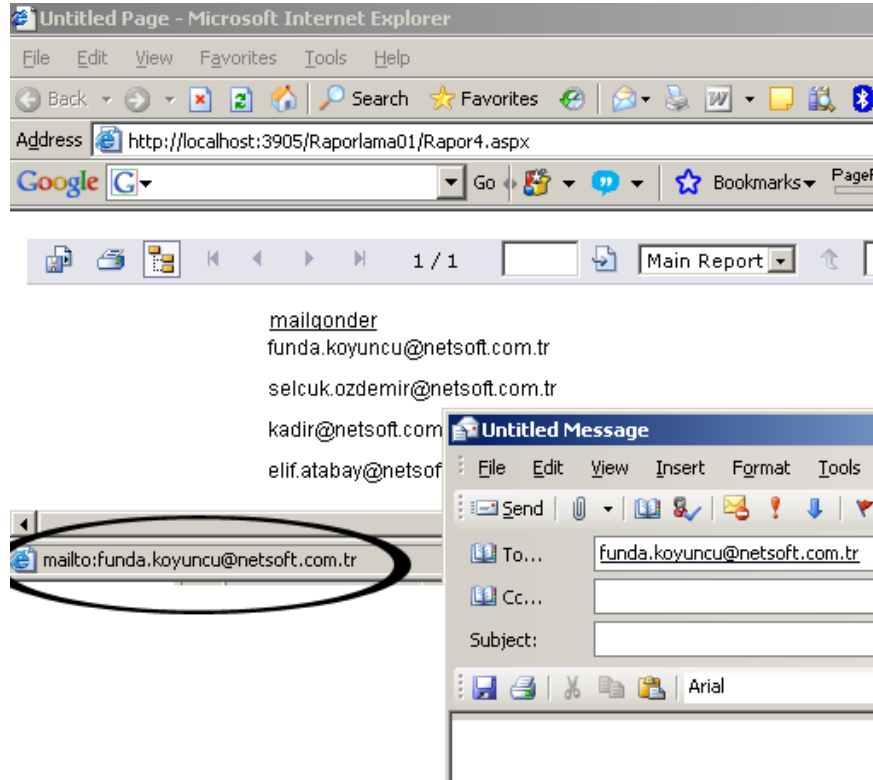
Şekil 6.30

Format Editor penceresi açılacaktır. Buradan üst tab seçeneklerden Hyperlink kısmına girelim. Mail adresi olacağından Current E-Mail Field Value seçeneğini seçerek atanan data kolonunun otomatik olarak mailto kavramını alacağını söyleyebiliriz (Şekil 6.31). OK butonuna basarak pencremizi kapatalım.



Şekil 6.31

Raporumuzu web sayfamıza atayarak çalıştıralım. Karşımıza datadan gelen mail adresleri tıkladığımızda Outlook programı açılarak Gönder (to) kısmında mail adresi yazılmış olacaktır (Şekil 6.32).



Şekil 6.32

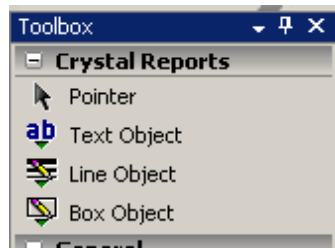


mailto:funda.koyuncu@netsoft.com.tr

Bu kod satırı html ortamında çalışan ve internet explorerların tanıdığı bir kod yorumudur. Yukarıda anlattığımız durumda html mantığı ile çalışması gerektiğinden mailto görünmektedir.

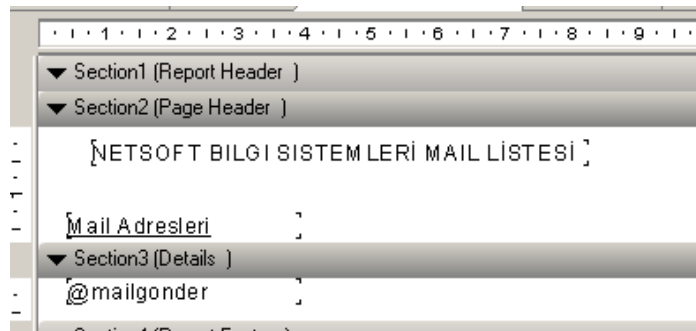
#### CRYSTAL REPORT OBJELERİ

Raporlarımızı düzenleme aşamasında oldukça kullanışlı objeler raporlarımızı daha iyi anlamamızı sağlayacaktır. Objelerimize raporumuzun tasarım alanında ToolBox penceresinden ulaşabiliriz (6.33).



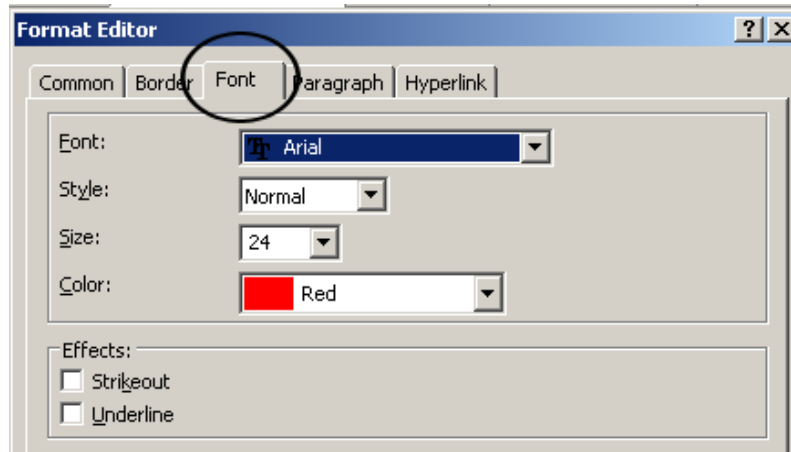
Şekil 6.33

Text Object raporumuza yazacağımız sabit bir yazı için kullanılabilir. Her raporun bir başlığı vardır. Bunun için Text Object'ı Page Header alanına sürükleyiniz ve içerisine rapor başlığımızı yazalım (Şekil 6.34) .



Şekil 6.34

Text Object nesnemizin font ayarı, rengi gibi değişiklikleri yapmak için Text Object'ı sağ tıklayarak Format Object bölümüne giriniz. Karşımıza gelen Format Editor penceresinden Font tabını seçiniz (Şekil 6.35).



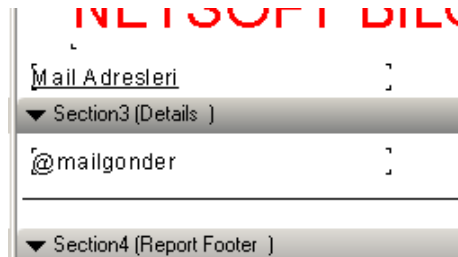
Şekil 6.35

Kişilere uyarıcı bildirimler yapmak için bu tarz font ayarlamaları yararlı olacaktır (Şekil 6.36). Eğer Text Object yazısını düzenlemek istiyorsanız yine objemizin üstünde sağ tıklayarak Edit Text Object seçeneğini kullanabilirsiniz.



Şekil 6.36

Line Object ve Box Object ler tasarım alanında çizgi ve kare çizmek için kullanılır. Kullanımları daha çok değerleri birbirinden ayırmak için kullanılır. Line Object'ı seçerek Section kısmında görünen mailgonder formülünün altına çizgimizi çizelim (Şekil 6.37).



Şekil 6.37

Böylece her mailimizin altında bir çizgi çizmiş olacağız. Mail adresleri daha düzgün görünmesinde Line Object rol oynayacaktır (Şekil 6.38). Çizgi üzerinde kalınlık, renk, tip olarak ayarlamalar yapmak için çizgi üzerinde sağ tıklayarak Format Objet seçeneğini seçmeniz yeterli olacaktır.



Şekil 6.38

Sizlere yardımcı olduğunu düşündüğüm bazı formülleri ve sonuçlarını anlatayım.

### **DateTimeToDate**

Veritabanından veritipini alır. Gelen değeri GG.AA.YYYY formatında yazar.

`DateTimeToDate({tblPersonel.Tarih})`

Sonuc: 10.10.2006

Direk olarak tarih kolonunu eklediğinizde sonu 10.10.2006 00:00:00 olarak görünecektir.

### **Average**

Belirttiğimiz bir kolonun veya belirleyeceğimiz değerlerin ortalamasını almak için kullanılır.

`Average({tblPersonel.Yas})`

`Average ([10,20,30,34,67])`

### **Maximum**

Seçtiğimi kolon yada değerler içerisinde en büyük olanını getirir

`Maximum ({tblPersonel.Yas})`

### **DistinctCount**

Gelen değerler içerisinde aynı olanları döndürmez toplam değeri verir.

`DistinctCount(["Funda","Kadir","Funda","Selçuk","Kadir","Elif"])`

Sonuc : Değer 4 çıkar, Funda ve Kadir isminde 2 tane olmasına rağmen 1 tane sayar.

### **Sum**

Verilen kolonun veya değerlerin toplamını verir.

`Sum({tblPersonel.Yas})`

`Sum([10,20,30,40,50,60])`

Sonuc: Yas kolonunun toplamını verecektir. Diğeri ise sayıların toplamını verecektir.

### UpperCase

Datamızdan nasıl gelirse gelsin raporlamada esnasından harfler büyük yazar.

UpperCase ({tblPersonel.Ad})

Sonuc: Ad kolonu büyük harflerle gelecektir.

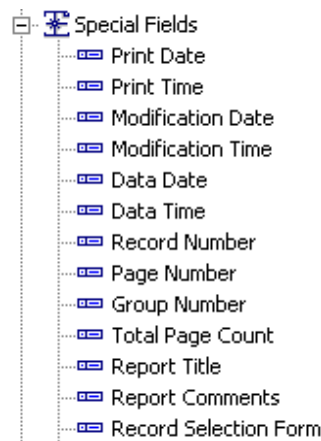
### LowerCase

Datamızdan nasıl gelirse gelsin raporlamada esnasından harfler küçük yazar.

LowerCase ({tblPersonel.Ad})

Sonuc: Ad kolonu küçük harflerle gelecektir.

Kendimizi yazdıklarının dışından Crystal Report'un bize sunduğu hazır fonksiyonlar bulunmaktadır. Special Fields altında bunları bulabilirsiniz. Sayfa numarası, kayıt numarası gibi hazır fonksiyonları da kullanabiliriz (Şekil 6.39).



Şekil 6.39

Hazır fonksiyonları da kullanarak raporlamanızı faydalı bir çıktı görüntüsü olarka yapılandırabilirsiniz.

# KONU 7

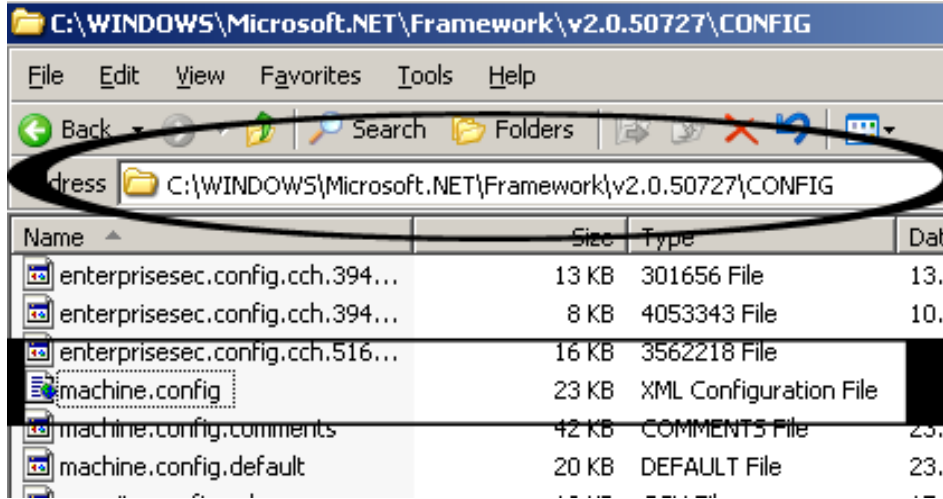
## WEB.CONFIG



## WEB.CONFIG

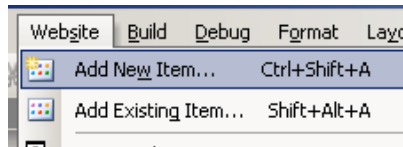
Asp.Net'in yeniliklerinden olan web.config dosyamız aslında bir ayar dosyasıdır. Ayarlarımızı bir XML formatında oluşturur ve ona göre işlemlerimizi gerçekleştirebiliriz. Web.config dosyası Asp.Net çalışmalarında mutlaka olacaktır kuralı yoktur. Web.Config olmadığı durumlarda daha üst bir seviyede olan machine.config dosyasıdır. Web.Config olmadığı durumlarda bu dosyadan ayarlar okunur. Machine.Config dosyasını incelemek istiyorsanız (Şekil 7.1);

C:\Windows\Microsoft.Net\Framework\[Framework Version]\CONFIG\machine.config



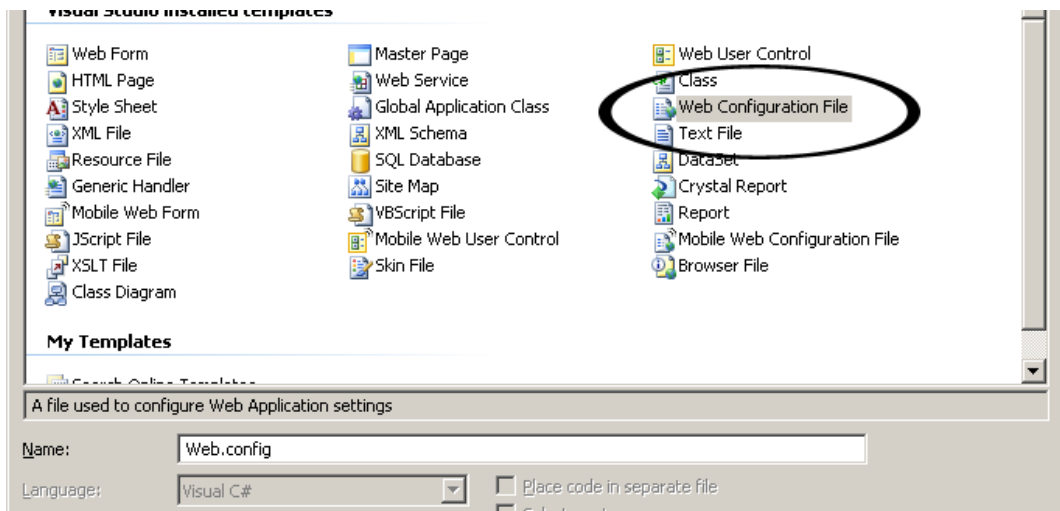
Şekil 7.1

Asp.Net projemizi açtığımızda Solution Explorer kısmında Web.Config dosyasını göremeyiz. Bunu için çalışma sayfamıza eklememiz gerekmektedir. Web Site > Add New Item seçeneğini tıklayınız (Şekil 7.2).



Şekil 7.2

Karşımıza gelen Add New Item penceresinden Web Configuration File seçeneğini seçerek Add butonuna tıklayınız (Şekil 7.3).



Şekil 7.3

Soution Explorer da web.config dosyamızın eklendiğini görebiliriz. Web.Config dosyamızın birimlerini inceleysek...

| Birim Adı        | Tanımı   |
|------------------|--|
| <trace>          | Asp.Net trace ayarlarının yapılmasını sağlar.  |
| <globalization>  | Uygulamanızın dil, sayı, para biçimleri gibi bölgesel ayarlarını yapmanızı sağlar.   |
| <customErrors>   | Uygulamanızda herhangi bir hata verdiğinde ne yapılması gerektiği belirlenebilir. Hata sayfasına git, mesaj iletisi ver gibi işlemler yapılabilir. |
| <authentication> | Üyelik sistemlerinde kullanıcıların geçerliliğini sağlamak için kullanılır.  |
| <sessionState>   | Session State ayarlarını bulundurur.   |
| <webServices>    | Web Sevisleri ile ilgili ayarları tutar.   |
| <securityPolicy> | Kullanabileceğiniz güvenlik ayarlarını içerir.   |

Web.Config nasıl kodlama yapılır dersiniz bunu örneklerle anlatalım. İlk olarak aklımıza gelen SQL Server bağlantı kısmını web.config den yapmak nasıl oluyor onu gösterelim.

#### UYGULAMA 1

- 1) Yeni bir web sayfası açalım. Web sayfamıza bir adet button ve GrdView nesneleri yerleştirelim.
- 2) Web.Config dosyamızda statik bir okuma sağlanır.

```
string deger= ConfigurationManager.ConnectionStrings["SqlBaglan"];
satırı ile okunması sağlanmaktadır.
```

- 3) Web.Config dosyamıza gelerek aşağıdaki kodu yazalım.

```
<connectionStrings>

<add name="SQLBaglan" connectionString="Data Source=Selcuk\SqlExpress;initial
Catalog=Ogrenici;integrated security=sspi;"
providerName="System.Data.SqlClient"/>

</connectionStrings>
```



Web.Config dosyasının kod alanına baktığınızda <connectionStrings/> olarak yazılarak kapalı olduğunu görürsünüz. Bu satırı sildikten sonra kodlamayı yapınız. Aksi takdirde hata alacaksınız.

- 4) Web.config dosyamızı yazdıktan sonra web sayfamıza gelerek Button1 üstüne çift tıklayarak kod alanına geçelim. Button1\_Click olayına yazdığımız SqlBaglanti isimli anahtarı çağıralım.

```
using System.Data.SqlClient; // Eklemeyi unutmayın
```

```
protected void Button1_Click(object sender, EventArgs e)
{
    SqlConnection baglan = new SqlConnection();
```

```

baglan.ConnectionString =
ConfigurationManager.ConnectionStrings["SqlBaglan"].ToString();

string sorgu = "select * from tblBilgiler";
SqlDataAdapter sadp = new SqlDataAdapter(sorgu,baglan);

DataSet ds = new DataSet();
sadp.Fill(ds);

this.GridView1.DataSource=ds.Tables[0];
this.DataBind();
}

```

5) Kodumuzu yazıp çalıştırdıktan sonra tüm sayfalarda artık aynı bağlantı satırını kullanabiliriz. Böylelikle veritabanı yolumuzda bir değişiklik olduğunda web.config dosyasından değiştirmemiz yeterli olacaktır. Derleme işlemi gerekmediğinden web.config dosyamızı düzenlemek için Visual Studio .NET platformuna gerek olmadan her işletim sisteminde bulunan not defteri (notepad) editöründe açarak düzeltebiliriz.

6) Kodumuzu çalıştırdığımızda tblBilgiler tablosu GridView1'e gelecektir (Şekil 7.4).



| ID | Ad     | Soyad         | Adres    | Yas |
|----|--------|---------------|----------|-----|
| 1  | Kadir  | KORUTÜRK      | Ankara   | 30  |
| 2  | Funda  | KOYUNCU       | Ankara   | 29  |
| 3  | Selçuk | ÖZDEMİR       | Ankara   | 26  |
| 4  | Cem    | BAYRAKTAROĞLU | Istanbul | 27  |
| 5  | Mehmet | EĞİTMEN       | Izmir    | 30  |
| 7  | Elif   | Atabay        | Istanbul | 28  |

Şekil 7.4

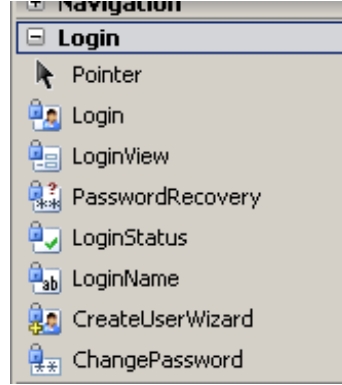
Web.Config dosyamıza hiçbirşey yazmasakta servera (FTP) gönderdiğimizde hata alma durumunuz söz konusudur. Bu hata birkaç güvenlik sorunundan kaynaklanmaktadır.

# KONU 8

## ÜYELİK SİSTEMİ

## LOGIN KONTROLLER

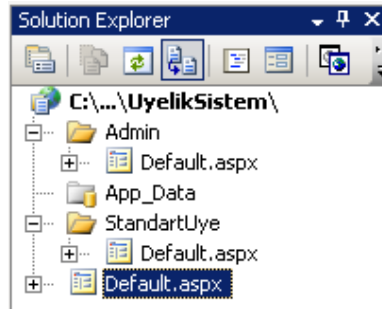
Hemen her sitede gördüğümüz üyelik alanlarını yazmak eskiden kodlarımızı yazmamız gerekirdi. Asp.Net 2.0 ile birlikte gelen Login nesnelerin neredeyse hiç kod yazmadan işlemlerimizi yapabiliriz. Toolbox penceresinden Login kısmına baktığımızda Login, Login View, Password Recovery gibi üyelik nesnelerini görebiliriz (Şekil 8.1). Login nesnelerinin çalışması için veritabanını oluşması ve yetkilerin atanması gibi bazı güvenlik düzeylerinin yapılması gerekmektedir.



Şekil 8.1

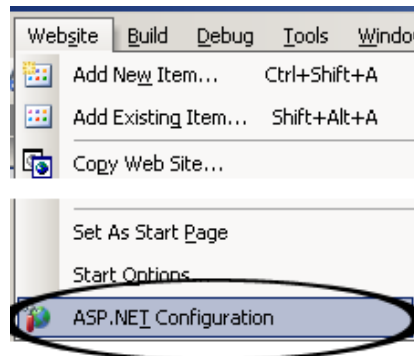
## UYGULAMA 1

1) Yeni bir web projesi açalım. Solution Explorerda proje isminin üstünü sağ tıklayarak New Folder seçeneğini seçelim ve StandartUye, Admin adından iki tane klasör oluşturalım (Şekil 8.2).



Şekil 8.2

2) Daha sonra üst menüden WebSite > ASP.NET Configuration seçeneğini tıklayalım (Şekil 8.3).



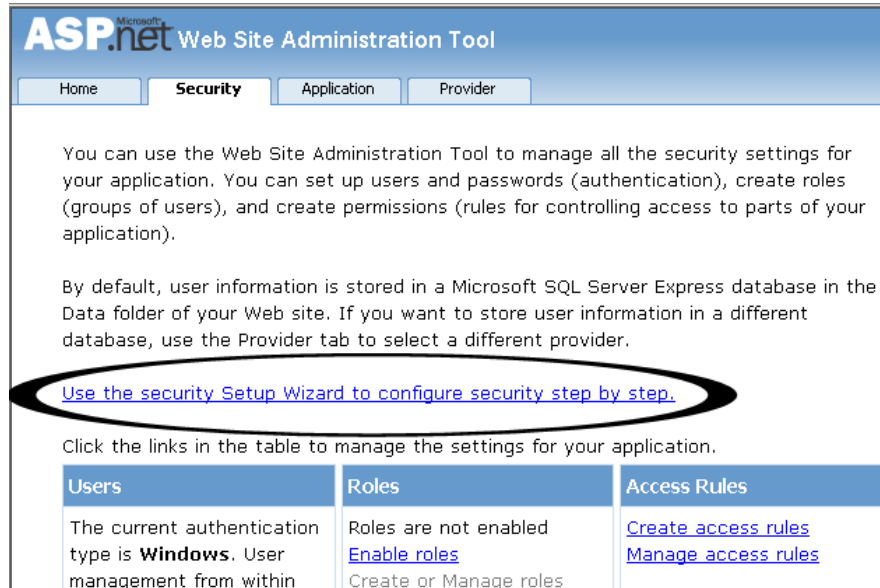
Şekil 8.3

3) Karşımıza Asp.Net Web Site Administration Tool web sayfası açılacaktır. Gelen pencereden Security linkini tıklayınız (Şekil 8.4) .



Şekil 8.4

4) Security bölümünü tıkladıktan sonra securiy tabında adım adım ayar yapacağımızdan **Use the security Setup Wizard to configure security step by step** linkini tıklayınız (Şekil 8.5). Karşımıza ayarşarı



Şekil 8.5

5) Karşımıza gelen Security Setup Wizard'ın ilk penceresini Next butonuna basarak diğer adıma geçiniz.  
6) Karşımıza iki seçenek gelmiştir. From the Internet ve From a local area network seçenekleri gelecektir. Biz uygulamamızı internete taşıyacağımızdan dolayı From the Internet seçeneğini seçerek Next butonuna basalım (Şekil 8.6).

☒ **From the internet**

Your application is a public site available to anyone on the Internet. Users can log in to your application by entering their user name and password in a login page you create.

☐ **From a local area network**

Your application runs in a private local area network (intranet) only. Users are identified by their Windows domain and user name and do not have to log in to your application explicitly.



Şekil 8.6

- 7) Karşımıza Advanced Provider Settings kısmına geçeceğimizi bildiren bir ileti gelecektir. Next diyerek bir sonraki aşamaya geçelim.
- 8) Daha sonra Roller tanımlamam gerektiğini belirten bir seçim gelecektir. Enable roles for this Web site seçeneğini işaretleyip Next düğmesine tıklayarak devam edelim.
- 9) Daha sonra karşımıza Role ismi isteyecektir. Buraya GenelKullanici ve Yonetici adından iki tane Rol tanımlayalım. Rol ismini yazdıktan sonra Add Role butonuna basalım (Şekil 8.7).

You have enabled roles for this web site.

### Create New Role

New Role Name:

### Existing Roles

|                |                        |
|----------------|------------------------|
| GenelKullanici | <a href="#">Delete</a> |
| Yonetici       | <a href="#">Delete</a> |

Şekil 8.7

- 10) Roller ekledikten sonra Next düğmesini tıklayarak devam edelim. Daha sonra karşımıza admin kullanıcıları oluşturmak için bir kayıt alanı gelecektir. Bilgileri girdikten sonra Create User butonuna tıklayınız (Şekil 8.8) .

Add a user by entering the user's ID, password, and e-mail on this page. You can also specify a question with an answer that the user must give when resetting a password or requesting a forgotten password.

### Create User

Sign Up for Your New Account

User Name:

Password:

Confirm Password:

E-mail:

Security Question:

Security Answer:

☒ Active User

Şekil 8.8

- 11) Kullanıcı oluşturduktan sonra karşımıza kullanıcının başarılı şekilde oluştuğunu belirten bir yazı gelecektir (Şekil 8.9). Eğer başka bir kullanıcı yaratmak istiyorsanız Continue butonuna tıklayınız. İşlerimiz bittiği için biz Next düğmesine tıklayarak işlemimize devam edelim.

Add a user by entering the user's ID, password, and e-mail on this page. You can also specify a question with an answer that the user must give when resetting a password or requesting a forgotten password.

Şekil 8.9

12) Daha sonrasında karşımıza Roller atayacağımız sayfa gelecektir. Burada GenelKullaniciya üye olan kişilerin daha önceden oluşturduğumuz Admin klasörüne girmemesini sağlayacağız. Bunu için sol taraftan Admin klasörünü seçiniz. Daha sonra Permission kısmından Deny seçeneğini seçerek Add This Rule butonuna tıklayınız. Fakat StandartUye klasöründen Allow diyerek giriş izni vermemiz gerekecektir. (Şekil 8.10).

Rules that appear dimmed are inherited from the parent and cannot be changed at this level.

| Permission | Users And Roles | Delete                 |
|------------|-----------------|------------------------|
| Deny       | GenelKullanici  | <a href="#">Delete</a> |
| Allow      | [all]           | <a href="#">Delete</a> |

< Back

Next >

Şekil 8.10

13) Rule tanımlamalarımızı yaptıktan sonra Next düğmesine basarak işlerimize devam edelim. Bu işlemden sonra yetkilendirme işlemlerimiz bitmiştir. Finish düğmesine tıklayarak işlemimizi bitirelim.

14) Şimdi açık olan visual studio .net ortamına gelelim. Buraya login nesnemizi web sayfamıza ekleyelim (Şekil 8.11).

Şekil 8.11



15) Hiçbir kod yazmadan sayfamızı çalıştıralım. Biraz önce kullanıcı oluşturmada kullandığımız bilgilerden kullanıcı adı ve şifresini girip Log In butonuna basalım. Eğer kullanıcı bilgilerini yanlış yazarsanız bilgilerinizin yanlış olduğuna dair bir ileti alacaksınız.

Login nesnesinin özellikleri ise...

| Özellik Adı          | Tanımı  |
|----------------------|---|
| FailureText          | Giriş hatalı olduğunda çıkacak olan yazı                        |
| LoginButtonText      | Login butonun üstündeki yazı                                    |
| LoginButtonType      | Giriş butonunun tipini belirler                                 |
| Orientation          | Login ekranın yatay veya dikey görünmesini sağlar               |
| PasswordLabelText    | Password ifadesinin yazısı                                      |
| RememberMeText       | Remember Me Next Time (Beni hatırla) yazısı                     |
| ToolTip              | ToolTip yazısı  |
| UserNameLabelText    | User Name ifadesinin yazısı                                     |
| TitleText            | Başlıktaki yazı   |
| DestinationPageUrl   | Bilgiler doğru ise yönlendirilmesi gereken web sayfası          |
| CreateUserText       | Üye olmak isteyenlerin ekran göreceği yazı                      |
| CreateUserUrl        | Üyelik olmak isteyenlerin yazıyı tıkladıklarında açılacak sayfa |
| PasswordRecoveryText | Şifremi unuttum yazısı  |
| PasswordRecoveryUrl  | Şifremi unuttum yazısını tıklayınca açılacak sayfa              |

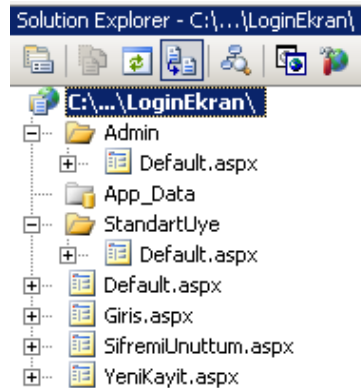


ASP.NET Configuration kısmında kullanıcı, rol veya rulları oluştururken Sql Server hataları alma durumunuz olabilir. Bunun nedenlerinden bazıları, Framework de sorun olması veya açık olan Configuration sayfasının fazla süre açık kalmasından dolayı zaman aşımına uğramış olabilir. Bundan farklı olarak da Asp.Net user yetki hatası verebilir. Burada Commanda satırına gelerek (cmd.exe)

**C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet\_regiis -i**  
Komutunu çalıştırmanız gerekecektir.

## UYGULAMA 2

1) Yeni bir web sayfası açalım. Asp.Net Configuration ayarlamalarını yapalım. Klasör ve dosya düzenlemenizi yapınız (Şekil 8.12).

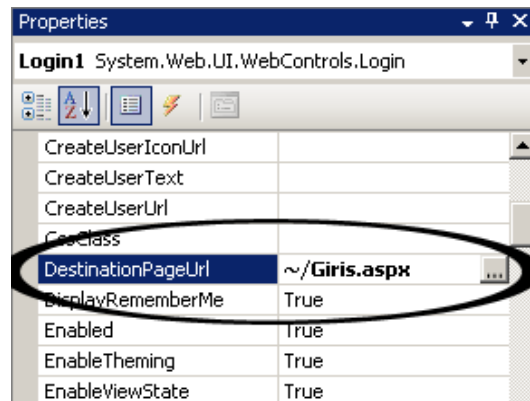


Şekil 8.12

2) Rootdaki Default.aspx dosyamızda Login nesnesini yerleştirelim (Şekil 8.13).

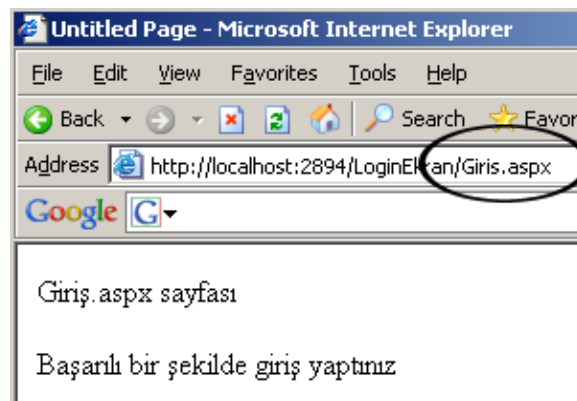
Şekil 8.13

3) Login nesnesinin properties penceresinden DestinationPageUrl özelliğine Giriş.aspx dosyasını atayınız (Şekil 8.14).



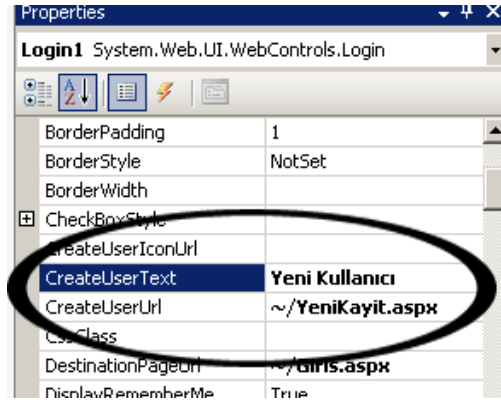
Şekil 8.14

4) Default.aspx dosyamızı çalıştıralım. Asp.Net configuration kısmında girdiğimiz kullanıcı adı ve şifresini kullanarak giriş yapalım. Girdiğimiz bilgiler doğru ise Giriş.aspx sayfamıza yönlendirilecektir (Şekil 8.15). Aksi takdirde hata mesajı verecektir.



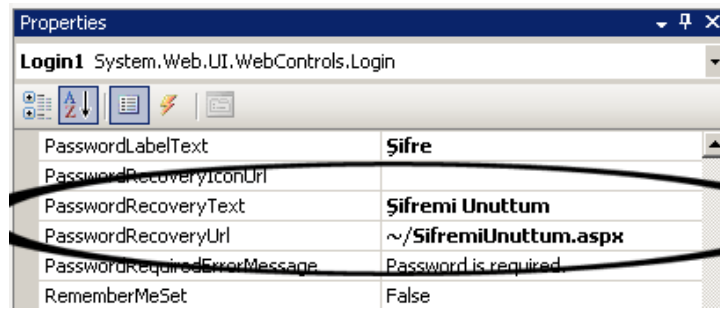
Şekil 8.15

5) Tekrardan şimdi projemizde bulunan Login penceresinin bulunduğu Default.aspx sayfamıza gelelim. CreateUserText özelliğine “Yeni Kullanıcı” yazısını yazalım, CreateUserUrl bölümüne ise YeniKayit.aspx dosyamızı yönlendirelim (Şekil 8.16).



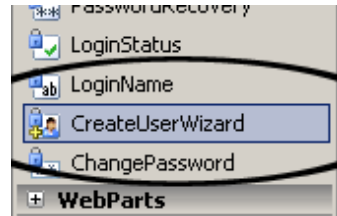
Şekil 8.16

6) Properties penceresinden PasswordRecoveryText özelliğine “Şifremi Unuttum” yazısını, PasswordRecoveryUrl kısmına ise SifremiUnuttum.aspx dosyamızı atayalım (Şekil 8.17).



Şekil 8.17

7) YeniKayit.aspx dosyamızı açalım. Design alanını açalım ve CreateUserWizard nesnemizi ekleyelim (Şekil 8.18).



Şekil 8.18

8) CreateUserWizard seçeneğini sayfamıza eklediğimizde kayıt için gerekli bir form gelecektir. Bu formu properties penceresinden değerlerimizi türkçeleştirebiliriz (Şekil 8.19).

Address: http://localhost:2894/LoginEkran/YeniKayit.aspx

Google

Sign Up for Your New Account

Kullanıcı Adı:

Şifre:

Tekrar şifre:

E-mail:

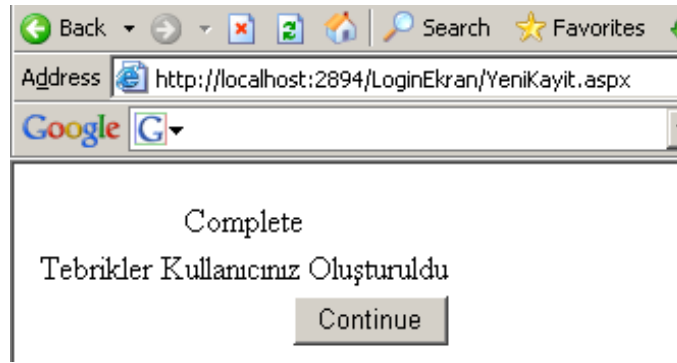
Güvenlik Sorusu:

Cevabı:

Kullanıcı Oluştur

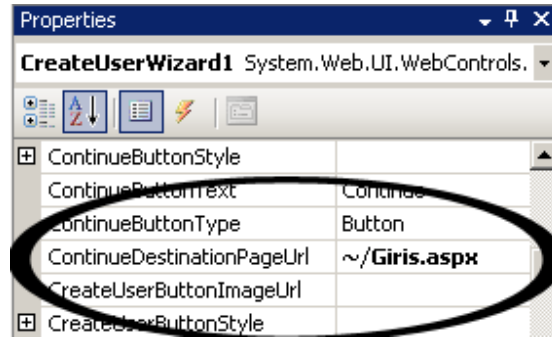
Şekil 8.19

9) YeniKayit.aspx dosyamızı çalıştıralım. Formdaki gerekli bilgilerimizi girdikten sonra Kullanıcı Oluştur butonumuzu tıklayalım. Bilgilerimizi doğru girdikten sonra kullanıcı oluşacaktır (Şekil 8.20).



Şekil 8.20

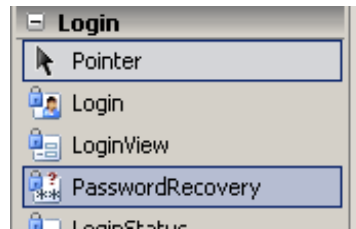
10) Continue butonuna bastığımızı göndermemiz gereken bir sayfa var ise properties penceresinden ContinueDestinationPageUrl özelliğine dosyanızı atamanız yeterlidir (Şekil 8.21).



Şekil 8.21

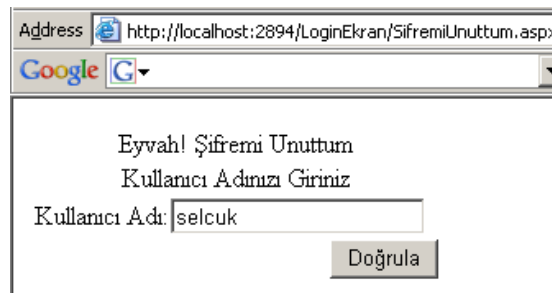
11) Böylece artık default.aspx sayfamızdan yeni kayıt olduğumuz kullanıcılar giriş yapabilirler.

12) Şimdi ise SifremiUnuttum.aspx dosyamızın design kısmına geçelim. Açılan web sayfamıza Login bölümünden PasswordRecovery nesnemizi yerleştirelim (Şekil 8.22).



Şekil 8.22

13) PasswordRecovery nesnemizi properties penceresinden yine üstündeki yazılarımızı değiştirebiliriz (Şekil 8.23).



Şekil 8.23

14) Kullanıcı adımızı girdikten sonra Doğrula butonumuzu tıkladığımızda Güvenlik Soru gelecektir. Bu soruyuda doğru cevaplamanız gerekmektedir (Şekil 8.24).

Identity Confirmation  
Answer the following question to receive your password.  
Kullanıcı Adı: selcuk  
Cevap: Sirket Adi  
Soru:

Şekil 8.24

15) Bundan sonrasında localde deniyorsanız SMTP hatası alacaksınız. Bu hatayı görmeden önce kimden mail gideceğine dair bir kaç ayar yapmanız gerekmektedir. PasswordRecovery nesnesinin özelliğine geldiğimizde MailDefinition kısmında From seçeneğine kimden mail gidecekse onun mail adresi yazılması gerekiyor (8.25).

| Properties                                   |                          |
|--|--------------------------|
| PasswordRecovery1 System.Web.UI.WebControls. |                          |
| LabelStyle                                   |                          |
| MailDefinition                               |                          |
| BodyFileName                                 |                          |
| CC   |                          |
| EmbeddedObjects                              | (Collection)             |
| From   | selcuk.ozdemir@netsoft.c |
| IsBodyHtml                                   | False                    |
| Priority                                     | Normal                   |
| Subject                                      | Sifre Hatırlatma         |

Şekil 8.25

16) SMTP hatasını aşmak için bu kadarı yeterli değil. Tabikide sisteminizde SMTPServer yapısını kurulu olması gerekmektedir. Bu ayarlamadan sonra Kişiyi “Şifre Hatırlatma” başlığında From kısmındaki mail üzerinden kişiye sifresi gönderilecektir.

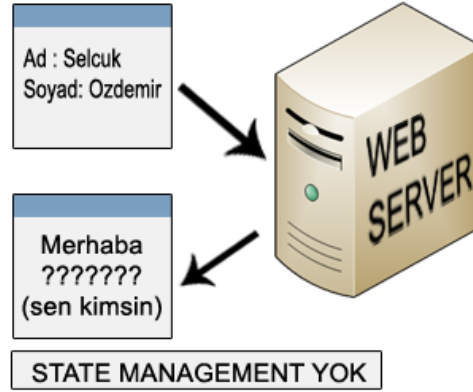
KONU 9

STATE / CACHE

## STATE MANAGEMENT

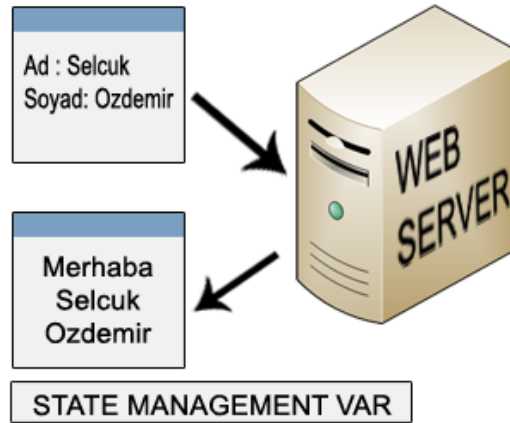
Web sayfalarında en çok kullanılan yöntemlerden biriside bir bilginin sayfalar arasın geçiş yapmasıdır. Bunun en iyi örneği üye girişi yaptığımız web sitesinde dolaşabilmemizdir. Asp.Net teknolojisi sayfalar arasında kullanıcı bilgilerini tutmayı (state) sağlar. Bu tarz uygulamalar windows uygulamalarda rahatlıkla yapılırken, her isteğimizi sürekli olarak server'a post edilmesinden dolayı Asp.Net uygulamalarından biraz farklıdır.

State management yok ise gönderilen bilgileri diğer sayfalar algılayamayacaktır (Şekil 9.1).



Şekil 9.1

Fakat state management var ise kullanıcı web sitemizi gezdikçe tanıyacaktır (Şekil 9.2).



Şekil 9.2

Kullanıcı bilgilerini saklamak ve kullanmak istiyorsanız kullanabileceğiniz yöntemler vardır.

- 1) View State
- 2) Session State
- 3) Application State

View State kontrolü kendisine verilen bilgiyi gizli bir alanda tutar. Bu bilgi hafızaya Base64 formatında bir karakter dizisi olarak yazılır ve her geri postalamada(postback) sunucuya getirilir ancak viewstate özelliği sunucu kontrolleriyle sınırlıdır. Yüklenen bilgi ya sizin tarafınızdan oluşturulan bir custom object veya basit bir bilgi türü olmak zorundadır.

Session State kontrolü HttpSessionState sınıfını kullanır. Kullanıcıların bilgilerini içeren kod bilgileri, 120-bit'lik Session ID'de tutulur. Bilgileri tespit etmek için Session ID yardımı ile olur.

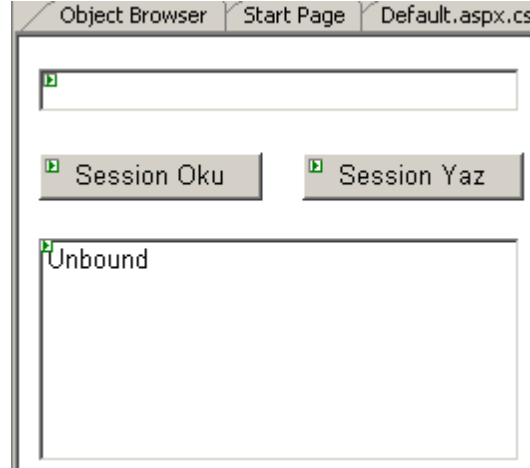
Application State web uygulaması içindeki tüm sayfalar için global bir sistemdir. Ziyaretçi sayısını tutulması Application üzerinden kullanılır.

## SESSION STATE

Bir bilgimiz kontrol ederek o bilgiler doğrultusunda değerleri aktarmakda diyebiliriz. Session state'i uygulamalar ile daha net anlayacağız.

### UYGULAMA 1

1) Bir web sayfası açalım birer adet textbox, listbox ve iki adette buton yerleştirelim (Şekil 9.3).



Şekil 9.3

2) Aşağıdaki kodu yazarak çalıştıralım. Çalıştırma durumunda textbox'a bir değer yazını ve Session Yaz butonuna tıklayınız. Daha sonra Session Oku butonuna basarak işlemin gerçekleştiğini görebilirsiniz (Şekil 9.4).



Şekil 9.4

3) Kodumuzu aşağıdaki gibi yazalım

```
protected void btnSessionYaz_Click(object sender, EventArgs e)
{
    Session["state"] = this.TextBox1.Text;
    this.ListBox1.Items.Add("");
    this.ListBox1.Items.Add("Session Değeri Yenilendi");
    this.ListBox1.Items.Add("Session'ın yeni değeri :" +
        Session["state"].ToString());
}
protected void btnSessionOku_Click(object sender, EventArgs e)
{
    if (Session["state"] == null)
    {
```



```

        this.ListBox1.Items.Add("");
        this.ListBox1.Items.Add("Session Boş");
        this.ListBox1.Items.Add("Bir Değer Giriniz");
    }
    else
    {
        this.ListBox1.Items.Add("");
        this.ListBox1.Items.Add("Deger" + Session["state"].ToString());
    }
}

```

4) Burada değerimiz state adında session nesnesine aktarılıyor. Aktarılan değer internet explorer üzerinden tutularak değerın okunmasını sağlayacaktır.

## UYGULAMA 2

1) Yeni bir web sayfası açalım. Daha sonra Web Site > Add New Item seçeneğinden yeni bir web form ekleyelim. Default.aspx sayfamıza bir textbox ve bir button ekleyelim. Daha sonra button'u çift tıklayarak kod alanına geçelim ve Button1\_Click olayına kodumuzu yazalım

```

protected void Button1_Click1(object sender, EventArgs e)
{
    Session["i"] = this.TextBox1.Text;

    Response.Redirect("Default2.aspx");
}

```

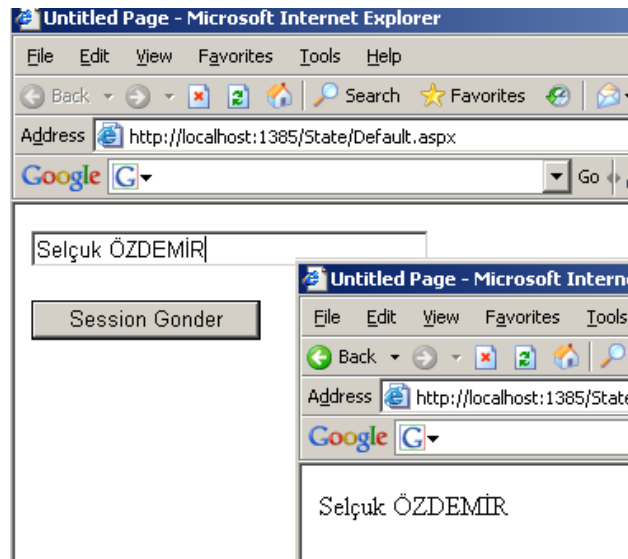
2) Default2.aspx sayfamıza geçelim ve bir adet label nesnesi yerleştirelim. Form\_Load olayına kodumuzu yazalım.

```

protected void Page_Load(object sender, EventArgs e)
{
    Session["deger"] = Session["i"];
    Label1.Text = Session["deger"].ToString();
}

```

3) Kodumuzu çalıştırdığımızda Default.aspx'de bulunan textbox'a bir değer yazarak button'a bastığımızda diğer sayfaya aktarılacaktır (Şekil 9.5).

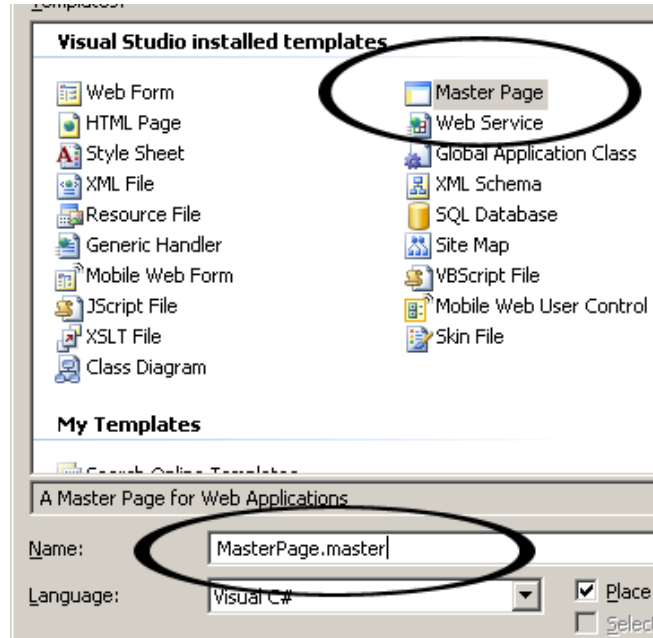


Şekil 9.5

## MASTER PAGE

Normal web formumuza benzer. Kodlaması ile tamamen aspx sayfalarda yaptığımız işlemleri yapabiliriz. Direk çalıştırmak dışında aspx de yaptığımız tasarımları da yapabiliriz. Master page'i anlamak için söyle anlatılabilir. Bir web sayfamız var ve sitemizin solmenu, ust banner gibi sayfanın belli kalıpları sabittir. Bu sabitleri bir kere tasarlayarak yada kodlayarak tüm sayfalarda kullanmamızı sağlayacaktır. Size önerim öncelikle masterpage'i tasarlayıp daha sonrasında kodlamaya geçmeniz olacaktır.

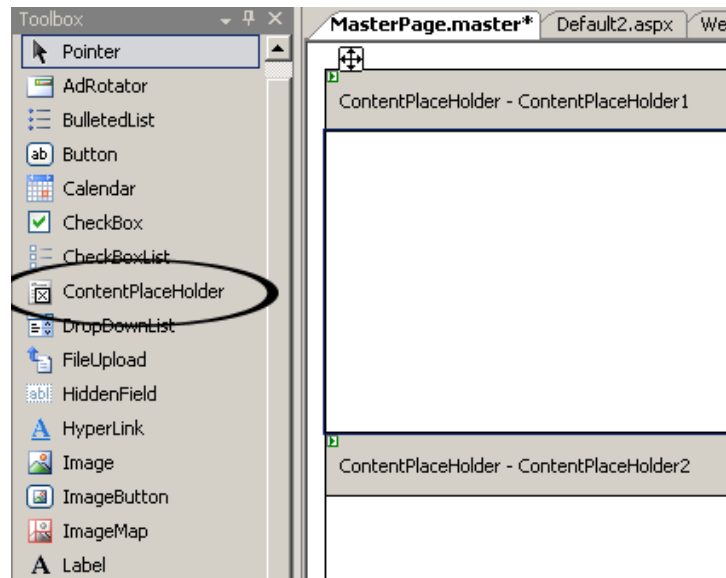
Masterpage oluşturmak için Web Site > Add New Item seçeneğinden Masterpage eklenebilir (Şekil 10.1).



Şekil 10.1

Dosyamızın uzantısı master türündedir. Master page'de düzenleme yaparken sizlere tavsiyem table objelerinin içerisinde çalışmaktır. Böylece sayfada kayma yada biçim bozukluğu olmayacaktır. Tabloların içerisine ContentPlaceHolder nesnesi eklememiz gerekmektedir. Öncelikle işlemlerin düzgün olması için bir tasarım yapılması gerekmektedir.

Masterpage sayfanısı açtığımızda design kısmında bir tane ContentPlaceHolder gelmiş olacaktır. Biz başka bir tane eklemek istediğimizde ContentPlaceHolder'ı toolbox penresinden tasarım alanına sürüklemiz yeterli olacaktır (Şekil 10.2).

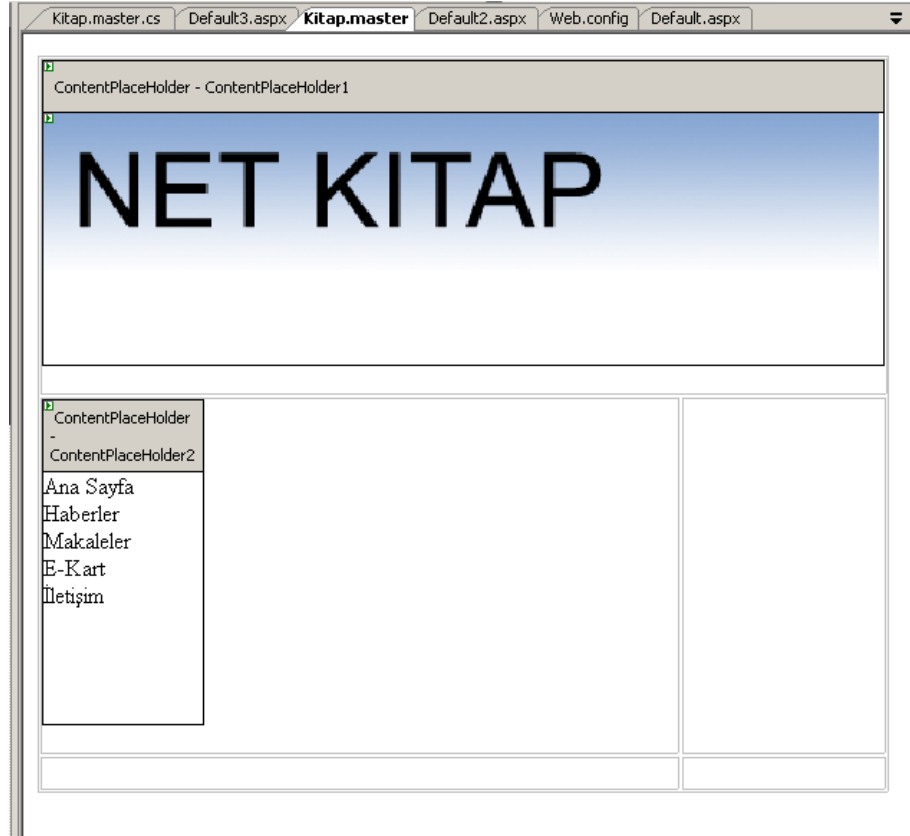


Şekil 10.2

Tablo oluşturduktan sonra ContentPlaceholder'ları tablonun içerisine sürüklememiz yeterli olacaktır.

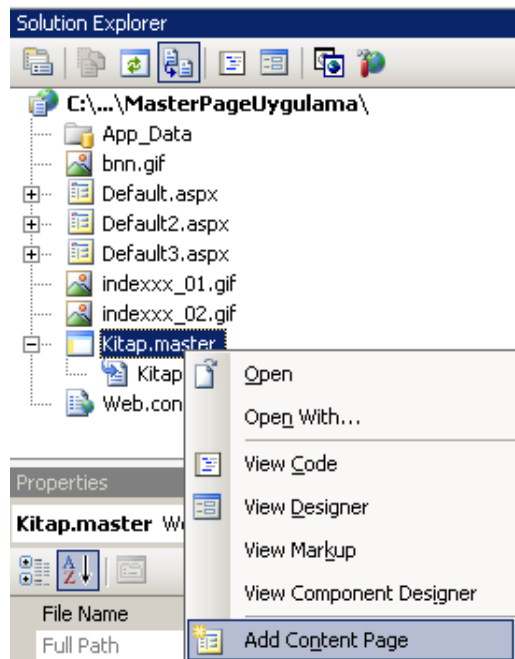
#### UYGULAMA 1

- 1) Bir proje açalım ve projemize Kitap.master adında bir master.page ekleyelim.
- 2) MasterPage içerisine tasarım için gerekli tablo düzeneğini oluşturunuz (Şekil 10.3).



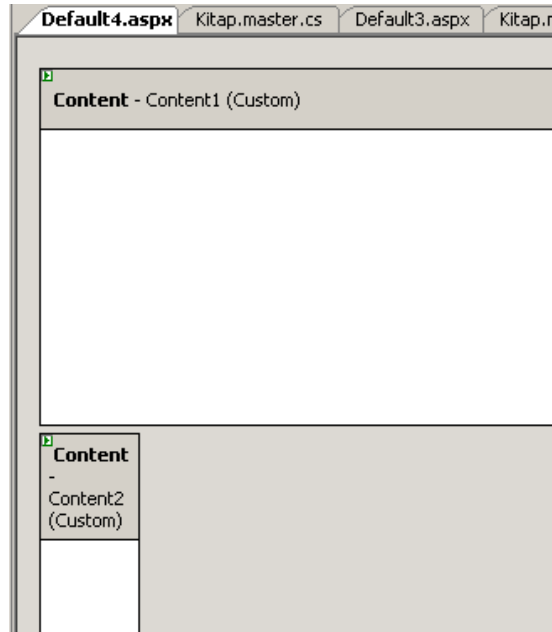
Şekil 10.3

- 3) Masterpage'i yaptıktan sonra buna bağlı sayfalar yapmak için Solution Explorerda Kitap.Master dosyamızı sağ tıklayalım, gelen menüden Add Content Page seçeneğini seçelim (Şekil 10.4).



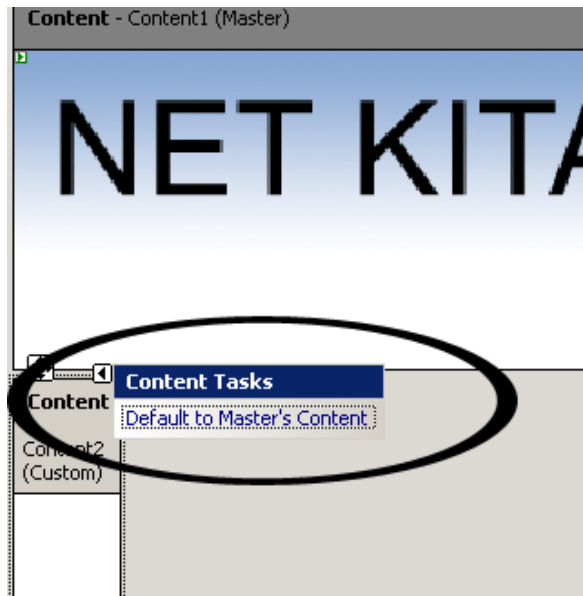
Şekil 10.4

4) Add Content Page seçeneğini tıkladığınızda otomatik olarak bir web sayfası oluşacaktır. Oluşturduğumuz web sayfasının Design kısmına geçtiğimizde Kitap.Master sayfasının kalıbının geldiği göreceksiniz (Şekil 10.5).



Şekil 10.5

5) Sayfamızı çalıştırdığımızda web sayfası boş gelecektir. Sayfanın görüntünü gelmesi için aspx de bulunan Content'i tıklayarak Default to Master's Content seçeneğini tıklayınız (Şekil 10.6).



Şekil 10.6

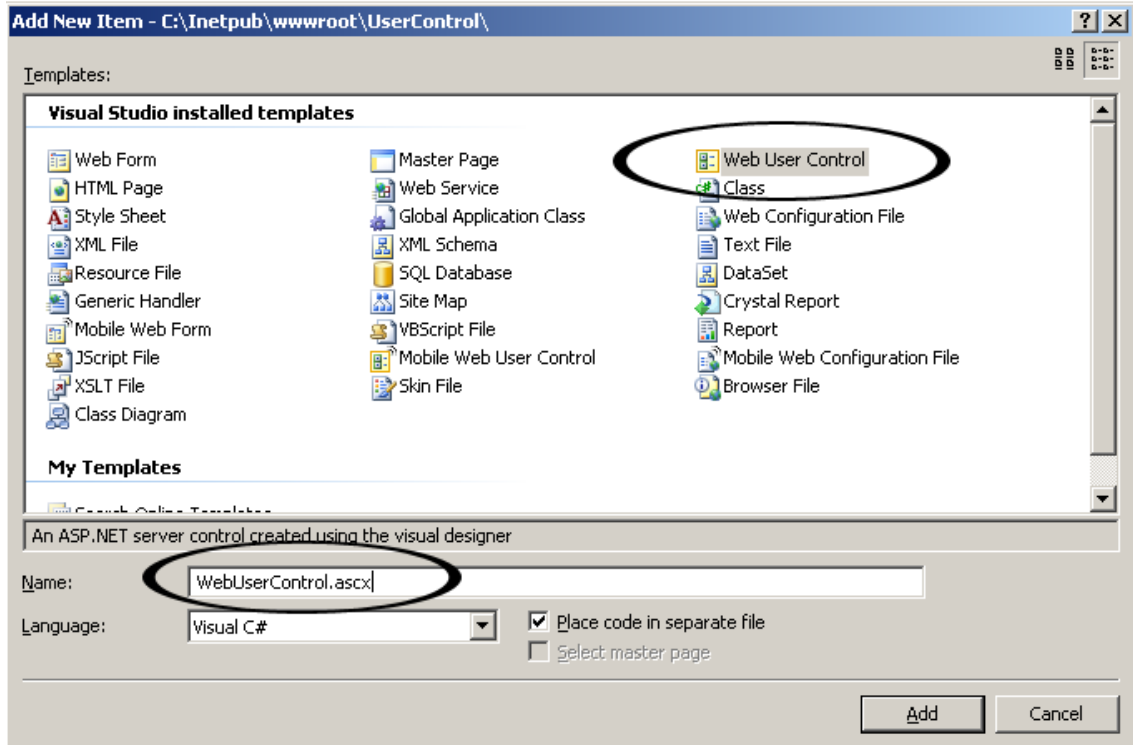
6) Artık masterpage leri kullanarak tüm sitemizde bir standart site görüntüsü yapabiliriz.

## USER CONTROL

Aslında Masterpage çıkmadan önce kullanılan bir teknik olarak nitelendirebiliriz. User Controller masterpageden biraz farklı çalışır. MasterPage de sayfa teması hazırlarken User Controller tek amacı yazıp sayfaya yerleştirmek için kullanılır. Yine bir sol menu düşünelim. Sol menüyü user control olarak yaptığımızda her sayfaya (aspx) tek tek taşımamız gerekmektedir.

### UYGULAMA 1

- 1) Yeni bir web projesi açalım. Add New Item seçeneğinden ek olarak 2 sayfa daha ekleyelim. Böylece projemizde Default.aspx, Default2.aspx, Default3.aspx adından toplam 3 tane sayfamız olacaktır.
- 2) Bunlara ek olarak Add New Item'dan Web User Control ekleyelim (Şekil 10.7).



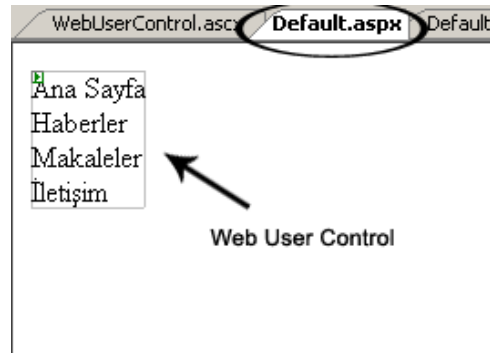
Şekil 10.7

- 3) Web User Control dosyamızın uzatısı ascx olarak tanımlanır. Normal web sayfaların farklı yoktur. Tasarım ve kodlamalar aynen Web User Controlde yapılabilir. Tek farklılık ascx dosyalar tek olarak çalıştırılmaması. Projemize eklemiş olduğumuz WebUserControl.ascx dosyamızı açalım ve içerisine bilgiler girelim (Şekil 10.8).



Şekil 10.8

4) Oluşturduğumuz Web User Control artık tüm sayfalarımızda çalışabilir. Bunu için Default.aspx, Default2.aspx, Default3.aspx dosyalarına webusercontrol.ascx dosyasını sürükleyiniz (Şekil 10.9).



Şekil 10.9

5) Artık tüm sayfalarımızda menumuzu kullanabiliriz. Web User Controlde yaptığımız tüm değişikliklerde diğer sayfalarda etkilenecektir.

KONU 11

XML

## XML NEDİR?

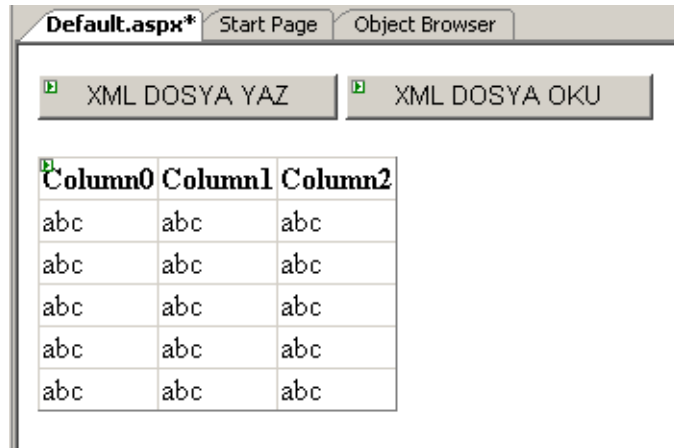
XML (Extensible Markup Language) genişletilebilir işaretleme dilidir. XML, elektronik ticaret, elektronik veri değişimi, terdarik zinciri bütünleştirmesi, veri yönetimi, akıllıarama makinaları gibi bir çok alanda stratejik bir araç olarak kullanılacak basit ve esnek metin biçimi teknolojisidir. XML'in özellikleri veri yapılarını, içeriklerini ve kavramlarınıplatform, şirket ve dilden bağımsız bir yapıda temsiline imkan vermektedir. XML uygulamalarımıza özel kendi işaretleme dilleri tanımlamamızı sağlayan bir meta dildir.

Genel olarak görüntüsü HTML diline benzesede aslında farklılıklar fazlasıyla mevcuttur. HTML sabit bir bilgiyi aktarırken XML data bilgilerini yorumlayarak bilgilerin güncel olmasını sağlamaktadır. XML'de içeriği kendimiz tanımlayabiliriz.

Dataset'deki bilgileri XML dosyasına aktararak verilere ulaşmada yüksek performans için kullanılır. Dataset'deki bilgileri tablo, sütuni, veritipi, constraint içerebilir. Bunun içinde XML Schmema (xsd) kullanılır.

## UYGULAMA 1

1) Yeni bir web projesi acalım. Çalışma sayfamıza iki buton ve bir gridview koyalım (Şekil 11.1).



Şekil 11.1

2) Daha sonra XML DOSYA YAZ butonunu çift tıklayarak kod alanına geçelim ve kodumuzu yazalım.

```
protected void btnXmlYaz_Click(object sender, EventArgs e)
{
    string baglan = "Data Source=salcuk\\sqlexpress;"
    +"initial Catalog=Ogrenci;integrated security=sspi";

    SqlConnection conn = new SqlConnection(baglan);
    string sorgu = "select * from tblBilgiler";
    SqlDataAdapter sadp = new SqlDataAdapter(sorgu, conn);
    DataSet ds = new DataSet();
    sadp.Fill(ds);

    // XML DOSYA OLUSTUR
    ds.WriteXml(@"c:\XmlDosya.xml");
    Response.Write("Dosya Oluşturuldu");
}
```

3) Kodumuzu çalıştırıp XML DOSYA YAZ butonuna bastığımızda C:\ rootda XmlDosya.xml adında bir dosya oluştuğunu görebiliriz. Burada XML dosyamıza Dataset üzerinden içerisindeki bilgiler ds.WriteXML metoduyla aktarılacaktır.

4) Oluşturduğumuz dosyayı çift tıklayarak açalım. Karşımıza internet explorer açılır ve ağaç şeklinde Xml dosyamız görünecektir (Şekil 11.2).





Şekil 11.2

5) XML dosyamızı oluşturduğumuza göre tekrardan asp.net ortamına geçerek XML DOSYA OKU butonunu çift tıklayınız ve kodumuzu yazalım.

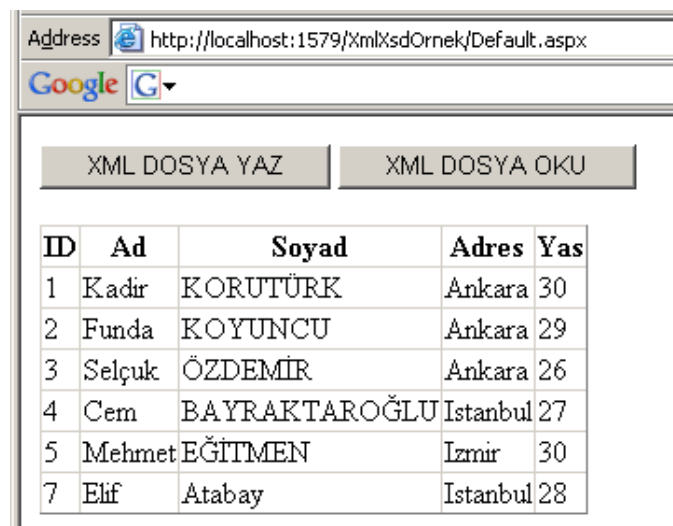
```

protected void btnXmlOku_Click(object sender, EventArgs e)
{
    DataSet verioku = new DataSet();
    verioku.ReadXml(@"c:\XmlDosya.xml");
    this.GridView1.DataSource = verioku.Tables[0];
    this.GridView1.DataBind();
}

```

6) Buradaki olay ise ReadXml metoduyla rootda bulunan xml dosyamızı okuyarak GridView'e aktarmak olacaktır.

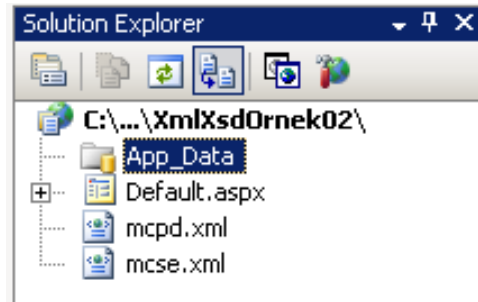
7) Kodumuzu çalıştırdıktan sonra XML DOSYA OKU butonumuzu tıkladığımızda datalar gridview'e aktarılacaktır (Şekil 11.3).



Şekil 11.3

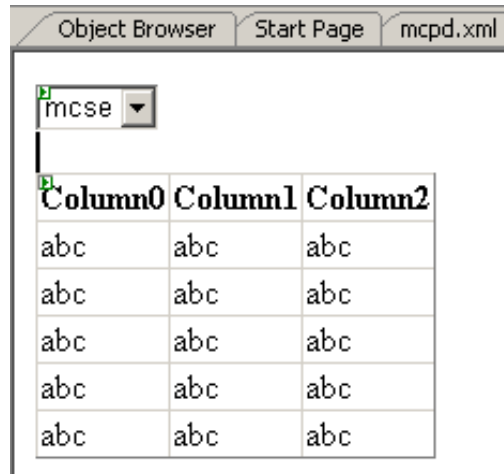
## UYGULAMA 2

1) Yeni bir web projesi açalım. Projemize Add New Item seçeneğinden mcpd.xml, mcse.xml adında iki adet dosya oluşturalım (Şekil 11.4).



Şekil 11.4

2) Default.aspx dosyamıza bir adet DropDownList ve GridView ekleyelim). DropDownListi tıklayarak properties penceresinden Item özelliğini kullanarak içerisine mcse,mcpd değerlerini girelim (Şekil 11.5).

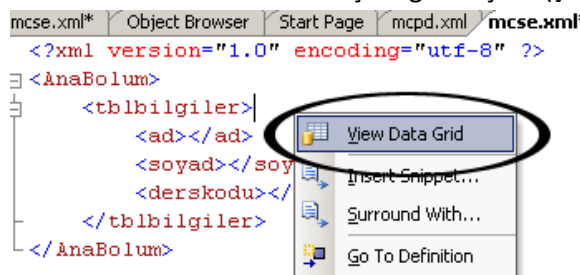


Şekil 11.5

3) Daha sonra mcse.xml dosyamızı .net ortamında açalım. Mcse.xml dosyamızı kolonlarımızı yazalım.

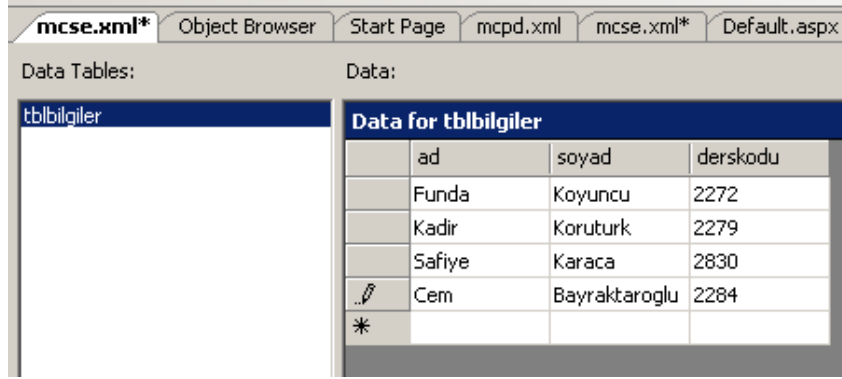
```
<AnaBolum>
  <tblbilgiler>
    <ad></ad>
    <soyad></soyad>
    <derskodu></derskodu>
  </tblbilgiler>
</AnaBolum>
```

4) Kolonlarımızı tablo şeklinde görmek ve içerisine rahat bilgi girişi yapmak için kod alanımızda mouse'u sağ tıklayınız. Karşımıza gelen menüden View Data Grid seçeneğini seçiniz (Şekil 11.6).



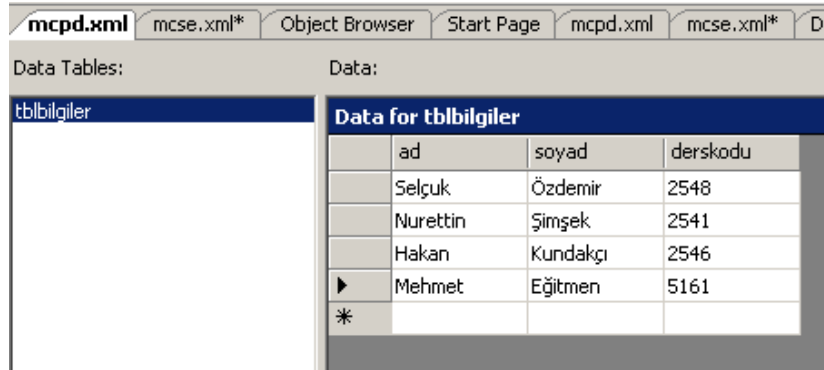
Şekil 11.6

5) Karşımıza gelen tabloya bilgilerimizi girelim (Şekil 11.7).



Şekil 11.7

6) Aynı işlemleri mcpd.xml dosyamız için gerçekleştirelim. İçerisine girediğimiz dataların farklı olmasına dikkat edin (Şekil 11.8).

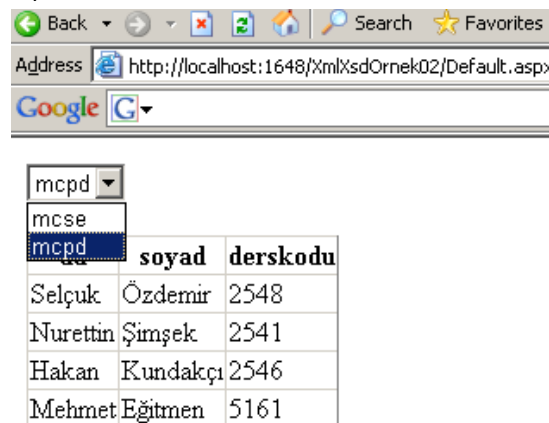


Şekil 11.8

7) Data girişlerimiz bittikten sonra default.aspx dosyamıza gelerek DropDownList nesnemize çift tıklayarak kod alanına geçelim.

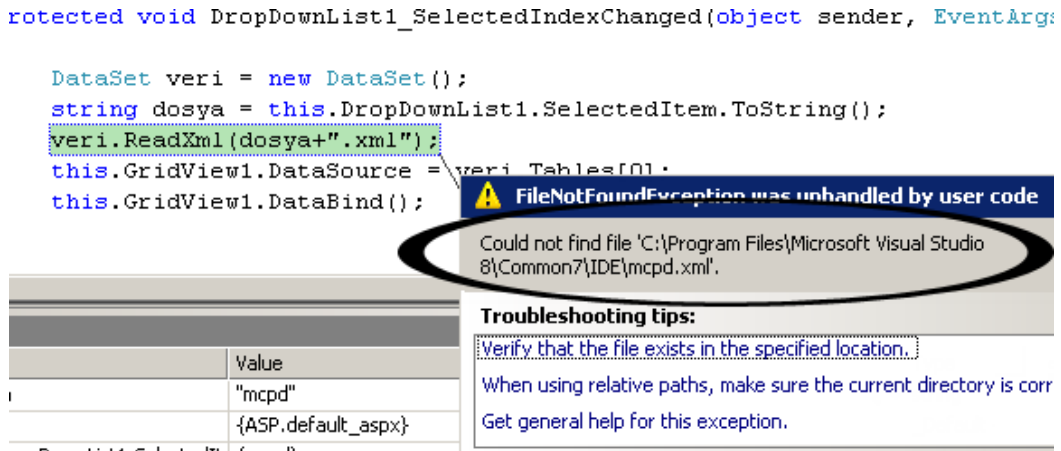
```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    DataSet veri = new DataSet();
    string dosya = this.DropDownList1.SelectedItem.ToString();
    veri.ReadXml(Server.MapPath(dosya+".xml"));
    this.GridView1.DataSource = veri.Tables[0];
    this.GridView1.DataBind();
}
```

8) Kodumuzu çalıştıralım ve DropDownList'den hangi alanı seçersek o dosya adına bağlı olarak xml dosyası yüklenecektir (Şekil 11.9).



Şekil 11.9

9) Burada Server.MapPath dosyalarımızın aynı yerde olduğunu belirlemektedir. Eğer Server.MapPath yazılmasaydı dosyalarımızı C:\Program Files\Microsoft Visual Studio 8\Common7\IDE klasöründe arayacak ve hata verecektir (Şekil 11.10).



Şekil 11.10

10) Bunun dışında DropDownList'den isim seçmemize rağmen gelmiyor ise DropDownList'i seçerek AutoPostBack özelliğini True yapmanız gerekmektedir.



Eğer sayfa yüklenince datanın GridView'e gelmesini istiyorsak. Form\_Load olayına kodumuzu yazabiliriz.

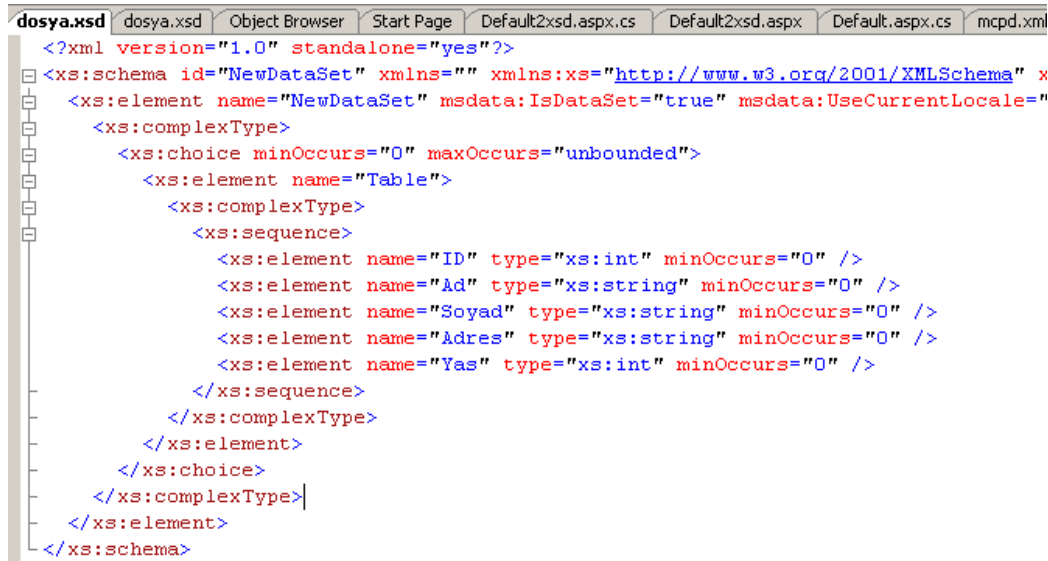
### XSD NEDİR?

XML Schema (XSD), XML döküman içerisindeki elementler sırası, sayısı, alt elementleri, veritinin tipi gibi tanımlamalarımızı yapacağımız standartları bulunan bir dil olarak tanımlanabilir. XSD'yi bir veritabanı oluşturmak gibi düşünebiliriz. Nasıl orada veritipi, kısıtlama gibi işlemler gerçekleştirebiliyorsak XSD'de de bu işlem gerçekleşir.

Dataset kullanarak XSD oluşturmak XML oluşturmaktan pek farklı yoktur.

ds.WriteXmlSchema(@"c:\sayfa.xsd");

olarak yazdığınızda rootda XSD dosyası oluşacaktır (Şekil 11.11).



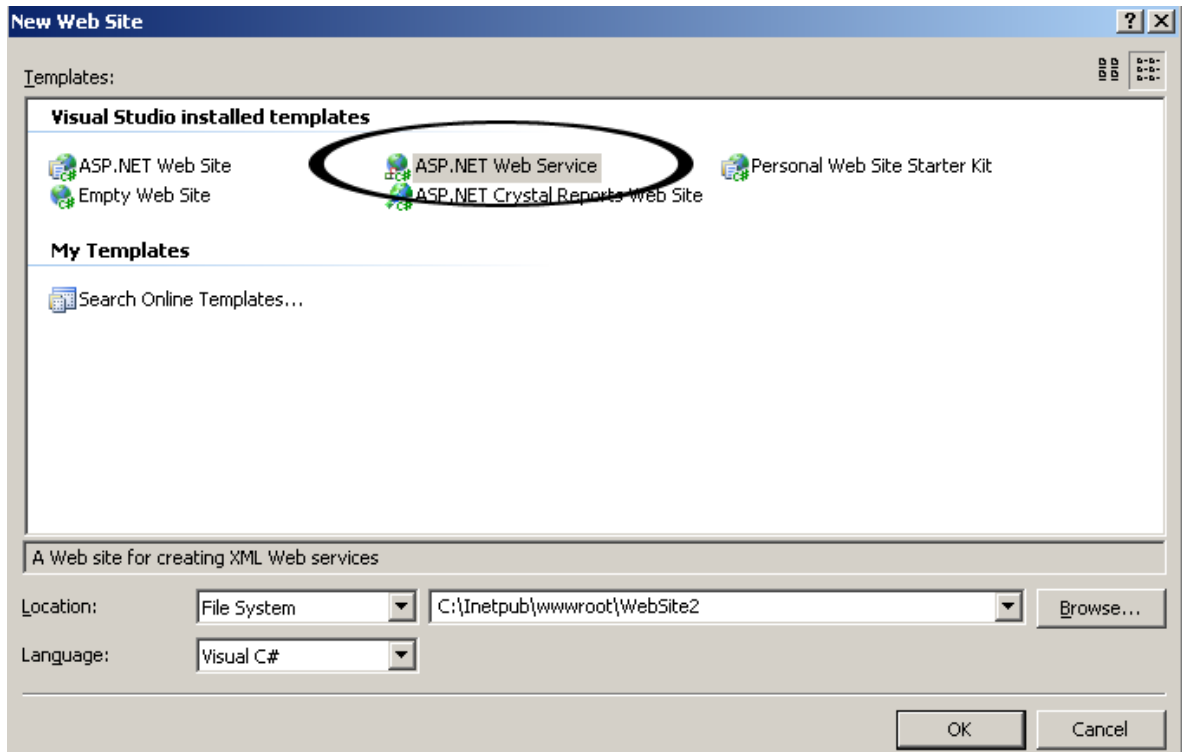
Şekil 11.11

## WEB SERVİSLER

Xml Web Servisleri yazılan program kodunu çalıştıran bileşenlerdir. Sıradan bileşenlerden farkı bu işlevi farklı bilgisayarlara da sunmasıdır. XML Web Servislerine farklı bir bilgisayar ve farklı bir platformdan istemci olunabilir. Yani XML Web Servisleri farklı platformlar arasında ortak olarak kullanılan bileşenlerdir. XML Web Servisleri ile farklı platformlar arasında veri aktarımında belli standartlar sayesinde oldukça kolaylaşmaktadır. XML Web Servisleri standart olan protokoller üzerinde çalışır. Bunlar, HTTP, XML, SOAP standartlarıdır. XML Web Servisleri adından da anlaşıldığı gibi mesaj alış verişi için XML standardını kullanır. XML' in bir standart olması ve herhangi bir platforma bağlı olmaması sayesinde XML Web Servisleri farklı platformların konuşmasını, veri alış verişini sağlar. Bu sayede uygulam geliştirme aşamasında programcılar farklı sistemleri kullanan uygulamalar yazarken standartları kullanarak işlerini daha kolay bir şekilde yapabilirler. XML Web Servisleri HTTP protokolü üzerinden mesaj alış verişin yaptığı için internet üzerinden bir XML Web Servisine erişim yapılabilmektedir. Ek olarak bir intranetde de XML Web Servisi uygulaması yapılabilir. Önemli olan XML Web Servisinin HTTP protokolü ile yani bir web sunucusu ile sunuluyor olmasıdır. Tabiki .NET Framework üzerinde programlanmış olan bir XML Web Servisinin sunulabilmesi için Internet Information Services (IIS) ve .NET Framework gerekmektedir.

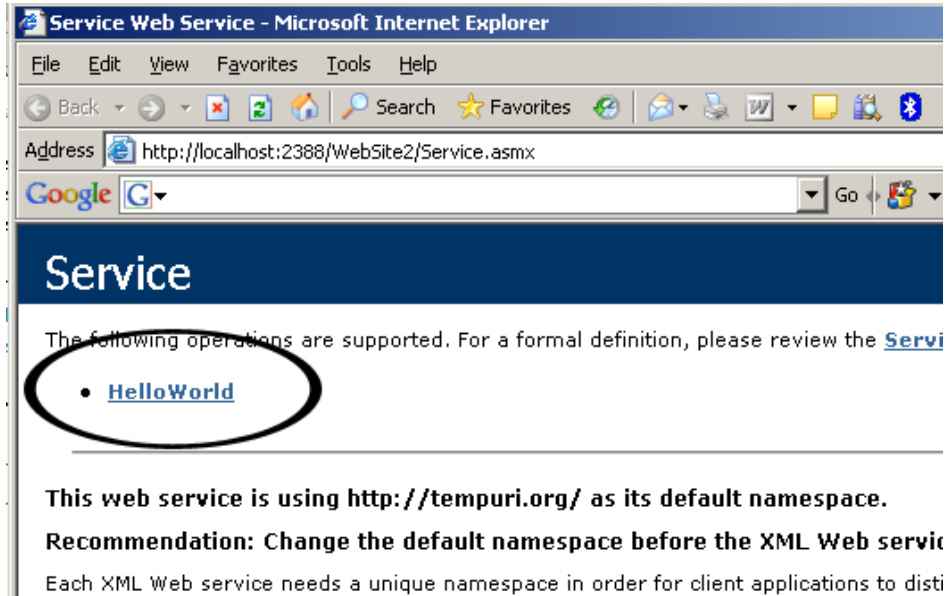
## UYGULAMA 3

- 1) Bu uygulamamızda öncelikle nasıl web servis kuracağımızı anlatacağım.
- 2) File>New>Web Site seçeneğini tıkladığımızda karşımıza gelen pencereden ASP.NET Web Service seçeneğini tıklayalım. OK butonuna bastıktan sonra Web Servisimize başlayalım (Şekil 11.12).



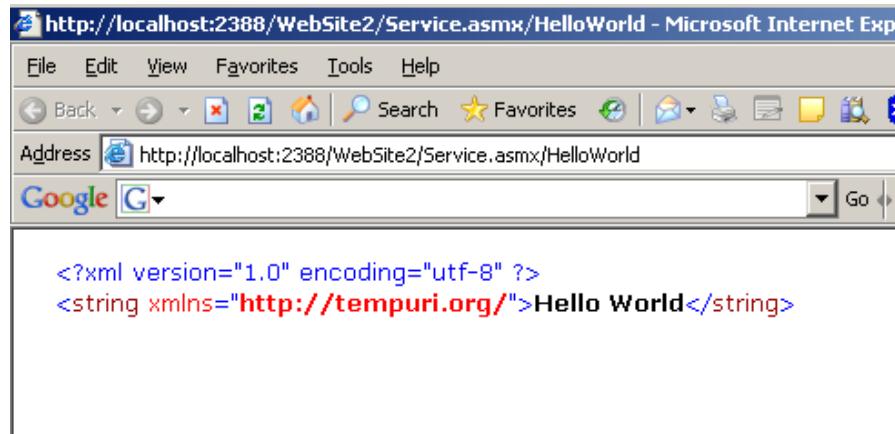
Şekil 11.12

- 3) Web servisi ekledikten sonra karşımıza Service.cs dosyası gelecektir. Burada [WebMethod] ile başlayan kısımlar bizi ilgilendiren servis adının yer aldığı kısımdır. Açtığımız projemize kod yazmadan F5 tuşuna basarak çalıştıralım (Şekil 11.13).



Şekil 11.13

4) Burada HelloWorld linkini tıkladığımızda karşımızda o servisin sayfası gelecektir. Invoke düğmesine tıkladığımızda XML formatında bilgisi görünecektir.



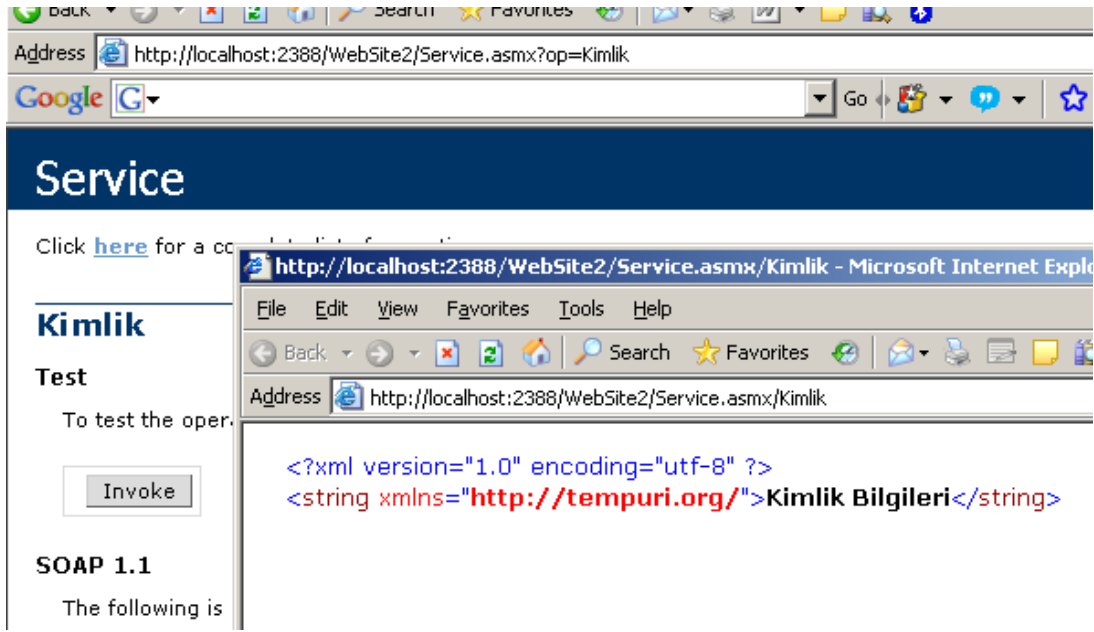
Şekil 11.14

5) Burada farklı bir method yazarak ekranda birden fazla servis adı görebiliriz (Şekil 11.15).



Şekil 11.15

6) Girmiş olduğumuz parcaları çalıştırarak deneyebiliriz. Kimlik'e tıkladığımızda karşımıza return değerinde yazan yazımız XML formatında gelecektir (Şekil 11.16).



Şekil 11.16

Basit düzeyde bir web servis yazılımını görmüş olduk. Şimdi ise girilen parametreye göre değerin gelmesini sağlayabiliriz.

## UYGULAMA 2

1) Bir önceki web servis kod alanına geçelim ve aşağıdaki kodu ekleyelim (Şekil 11.17).

```
[WebMethod]
public string Egitmen(string egitim)
{
    if (egitim == "mcse")
    {
        return "Mehmet EĞİTMEN";
    }
    else if (egitim == "mcpd")
    {
        return "Selçuk ÖZDEMİR";
    }
    else
    {
        return "Funda KOYUNCU";
    }
}
```

Hangi değeri girersek ona göre sonuç döndürecektir.

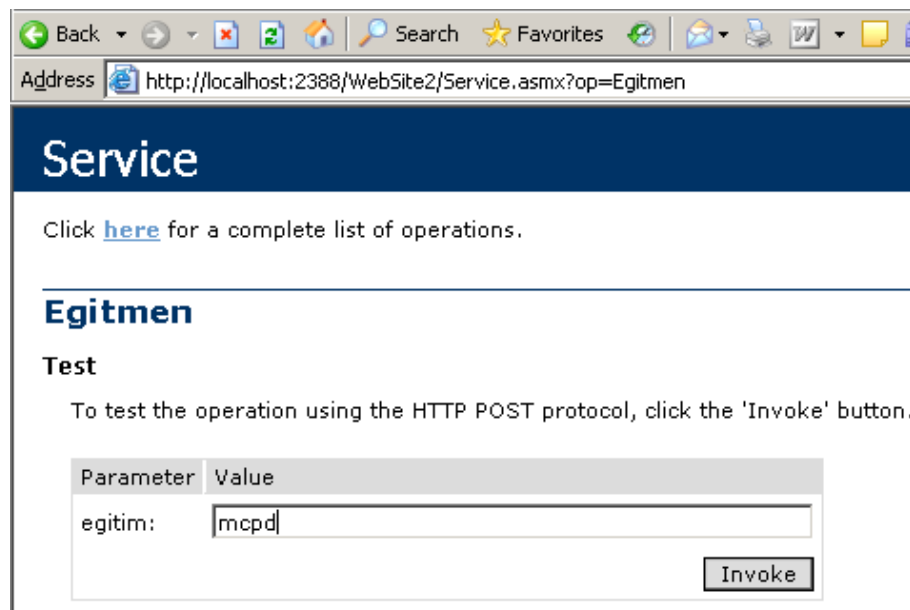
```

    }
    [WebMethod]
    public string Kimlik()
    {
        return "Kimlik Bilgileri";
    }
    [WebMethod]
    public string Egitmen(string egitim)
    {
        if (egitim == "mcse")
        {
            return "Mehmet EĞİTMEN";
        }
        else if (egitim == "mcpd")
        {
            return "Selçuk ÖZDEMİR";
        }
        else
        {
            return "Funda KOYUNCU";
        }
    }
}

```

Şekil 11.17

2) Kodumuzu çalıştırdığımızda karşımıza gelen Egitmen adında web servis ismi göreceksiniz. Egitmen linkini tıklayınız. Karşımıza bir text alanı gelecektir. Bu alana mcpd yazarak Invoke butonuna basınız (Şekil 11.18).



Şekil 11.18

3) Karşımıza XML dosyasında mcpd karşılığı gelecektir (Şekil 11.19).



## Egitmen

### Test

To test the operation using the H

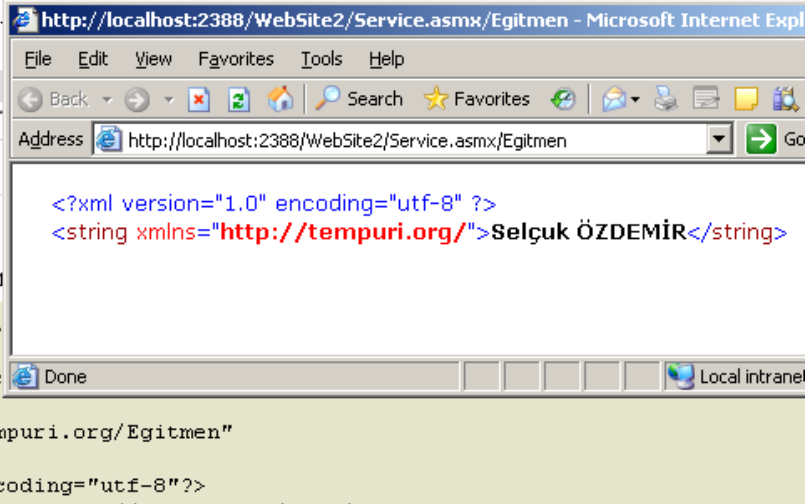
| Parameter | Value |
|-----------|-------|
| egitim:   | mcpd  |

### SOAP 1.1

The following is a sample SOAP 1

```
POST /WebSite2/Service.
Host: localhost
Content-Type: text/xml;
Content-Length: length
SOAPAction: "http://tempuri.org/Egitmen"

<?xml version="1.0" encoding="utf-8"?>
```



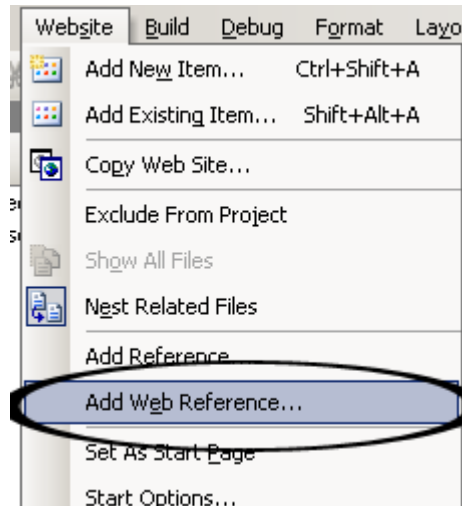
Şekil 1.19

Text alanımıza girdiğimiz mcpd değerinin karşılığı olan "Selçuk ÖZDEMİR" değerini döndürecektir.

Bu uygulamanın sonucunda bu yazdıklarımızı web sayfası (aspx) dosyasından nasıl okuyabileceğimizi görelim.

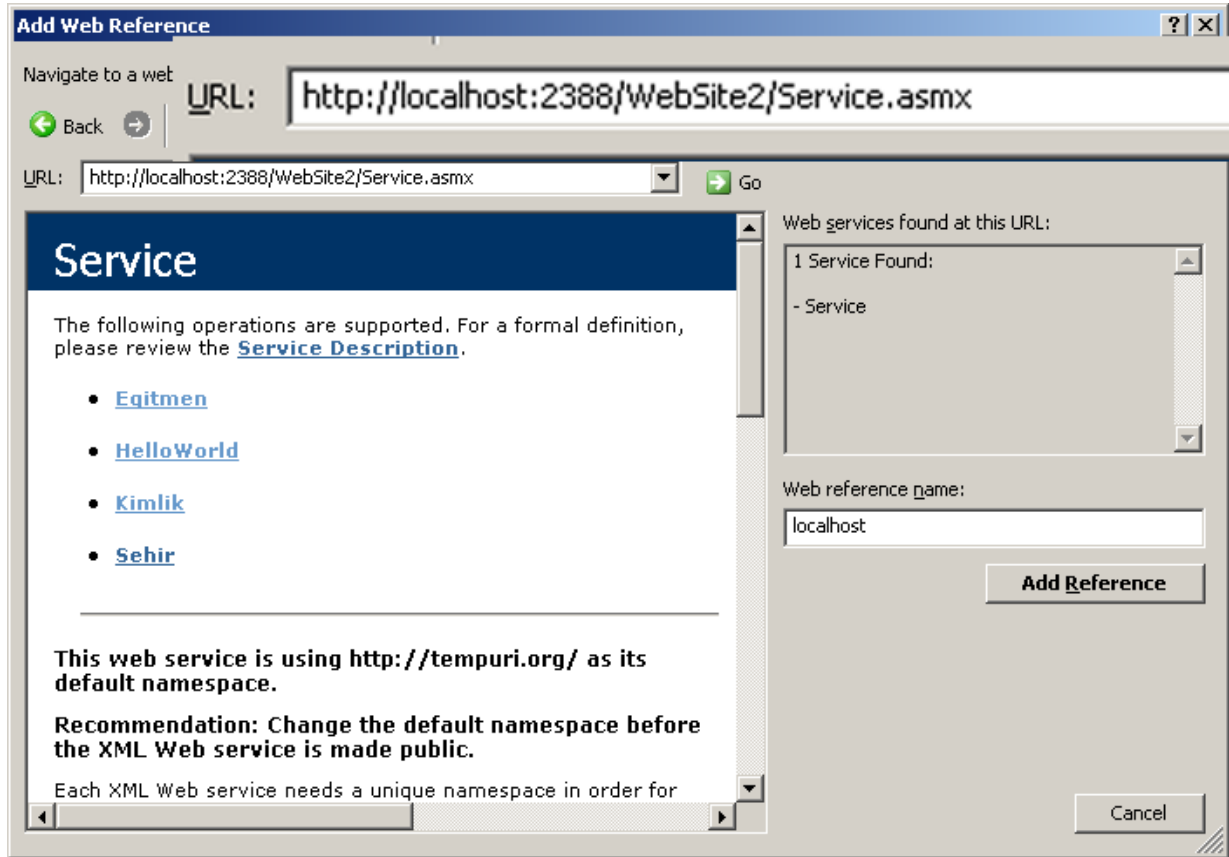
### UYGULAMA 4

- 1) Yeni bir web projesi açalım. Bu projede bir önceki uygulamada yer alan web servisi kullanacağımız gerekecektir.
- 2) Açılan web projemize Web Site>Add Web Reference seçeneğini tıklayalım (Şekil 11.20).



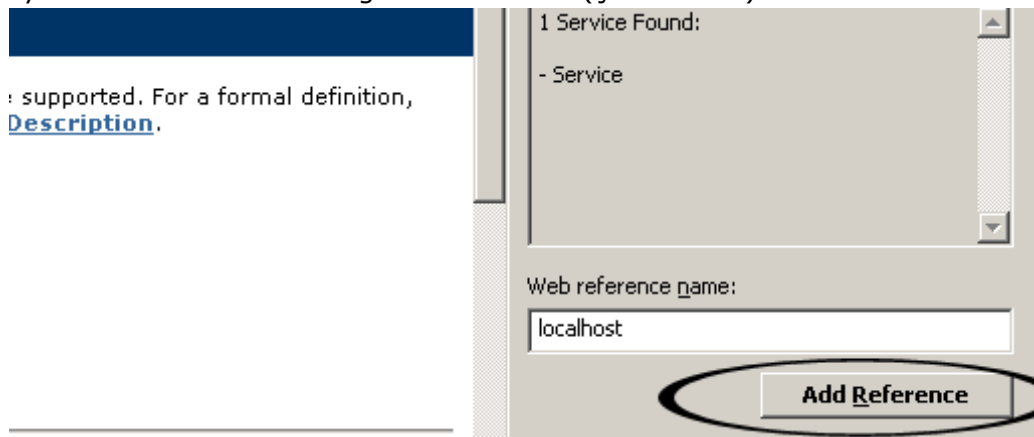
Şekil 11.20

3) Karşımıza Add Web Reference penceresi gelecektir. Burada bulunan URL kısmına bizim bir önceki uygulamada kullandığımız asmx uzantılı web servisinin yolunu yazınız ve GO butonuna basınız (Şekil 11.21). Yalnız burada adreste dikkat edeceğimiz husus asmx dosyasını bilgisayarada bulunan yolunu değilde çalıştırdığımızda internet explorerda oluşan adresi yazınız.



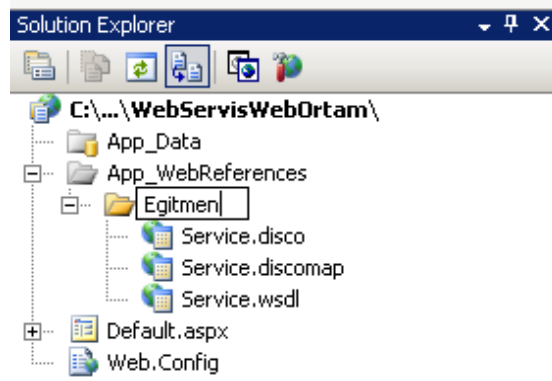
Şekil 11.21

4) GO simgesini tıkladıktan sonra web servis ekrana gelecektir. Bu servisi projemize dahil etmek istiyorsak Add Reference düğmesine basınız (Şekil 11.22).



Şekil 11.22

5) Projemize web servisi eklediğimi Solution Explorer'dan görebiliriz. Burada ismini değiştirmediyse localhost olarak geçmiş olacaktır. Localhost adındaki klasörün adını sonradan değiştirebilirsiniz. İsmi Egitim olarak değiştirelim (Şekil 11.23).



Şekil 11.23

6) Web sayfamıza (Default.aspx) bir buton ve textbox nesnesi koyalım (Şekil 11.24).

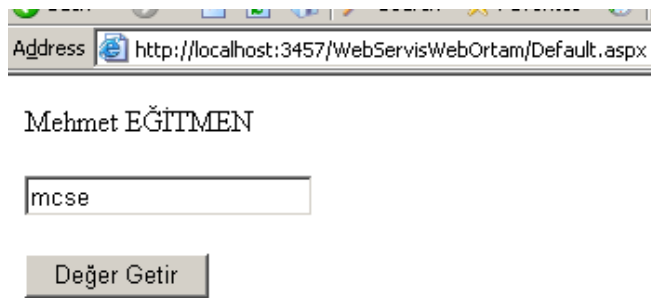


Şekil 11.24

7) Değer Getir butonumuzu çift tıklayarak kod alanımıza geçelim. Kod alanımıza aşağıdaki kodu yazınız.

```
protected void btnDegerGetir_Click(object sender, EventArgs e)
{
    Egitmen.Service egitmenler = new Egitmen.Service();
    Response.Write(egitmenler.Egitmen(this.TextBox1.Text));
}
```

8) Kodumuzu yazıp çalıştırdıktan sonra textbox'ın içerisine mcse değeri yazıp butona basalım. Ekrana mcse karşılığı olan "Mehmet EĞİTMEN" değeri gelecektir (Şekil 11.25).



Şekil 11.25

Burada service içerisinde bulunan Egitmen (egitmenler.Egitmen) biriminin değerini alabiliyoruz.

Web servisimizi veritabanına bağlayarak işlemlerimiz daha dinamik hale getirebiliriz. Bunun için bir döviz bürosuna ait bir webservis işlemi gerçekleştirelim. Veritabanımız Şekil 11.26'da yer almaktadır.

| Table - dbo.tblDoviz |    |                  |       |
|----------------------|----|------------------|-------|
|                      | ID | Para             | Deger |
| 1                    | 1  | Dolar            | 144   |
| 2                    | 2  | Euro             | 189   |
| 3                    | 3  | İngiliz Sterlini | 280   |
| 4                    | 4  | Japon Yeni       | 25    |

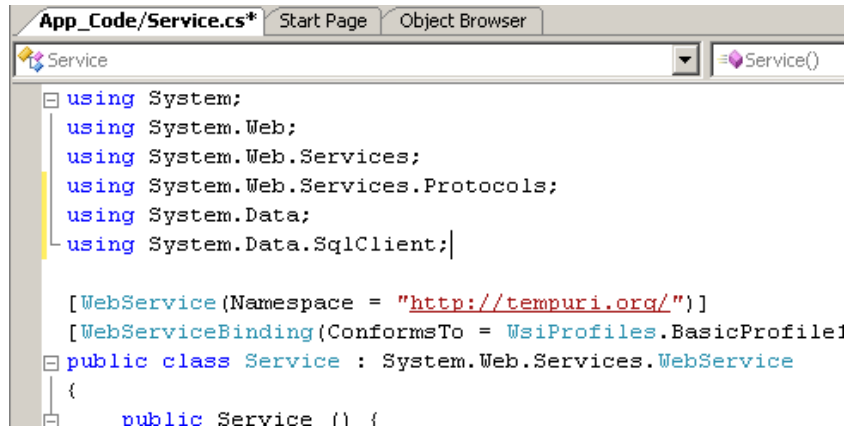
  

| Table - dbo.tblDoviz |             |              |                                     |
|----------------------|-------------|--------------|-------------------------------------|
|                      | Column Name | Data Type    | Allow Nulls                         |
| 1                    | ID          | int          | <input type="checkbox"/>            |
| 2                    | Para        | nvarchar(50) | <input checked="" type="checkbox"/> |
| 3                    | Deger       | int          | <input checked="" type="checkbox"/> |
| 4                    |             |              | <input type="checkbox"/>            |

Şekil 11.26

## UYGULAMA 5

- 1) Yeni bir web servis uygulaması başlatalım.
- 2) Database'e bağlanmak için System.Data ve System.Data.SqlClient namespaces'leri tanımlayalım (Şekil 11.27).



Şekil 11.27

- 3) Kod kısmına yazarak kodumuzu yazalım.

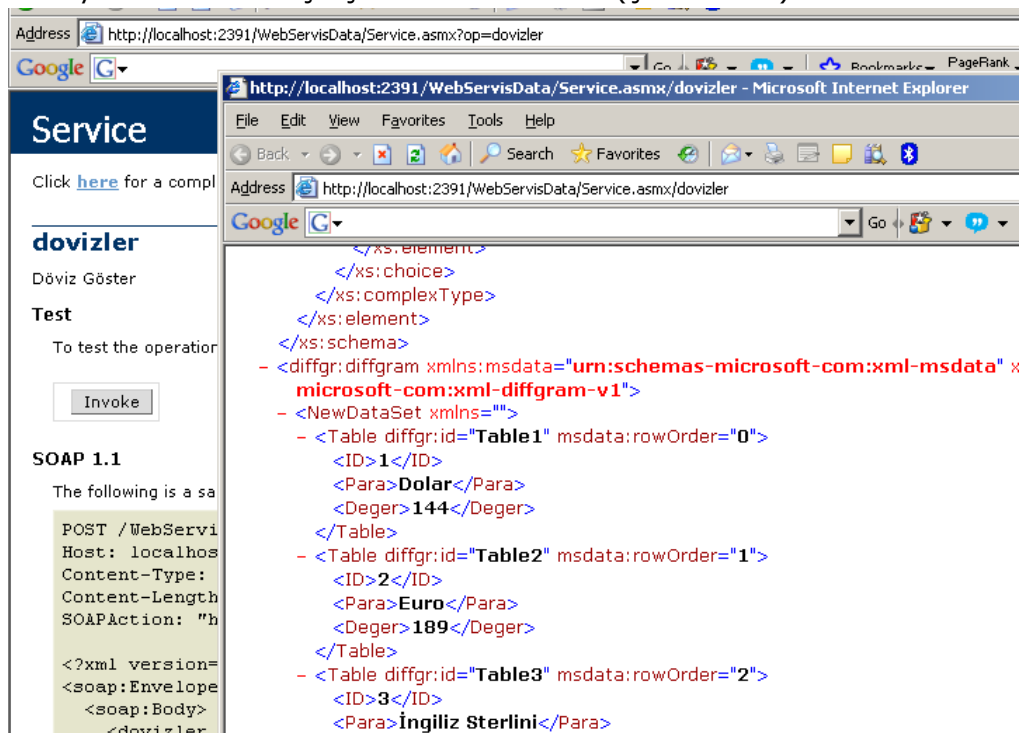
```

[WebMethod(Description="Döviz Göster")]
public DataSet dovizler() {
    string yol=@"Data Source=Selcuk\sqlExpress;"
    +"initial Catalog=ogrenci;integrated security=sspi";
    SqlConnection conn = new SqlConnection(yol);

    string sorgu = "select * from tblDoviz";
    SqlDataAdapter sadp = new SqlDataAdapter(sorgu,conn);
    DataSet ds = new DataSet();
    sadp.Fill(ds);
    return ds;
}

```

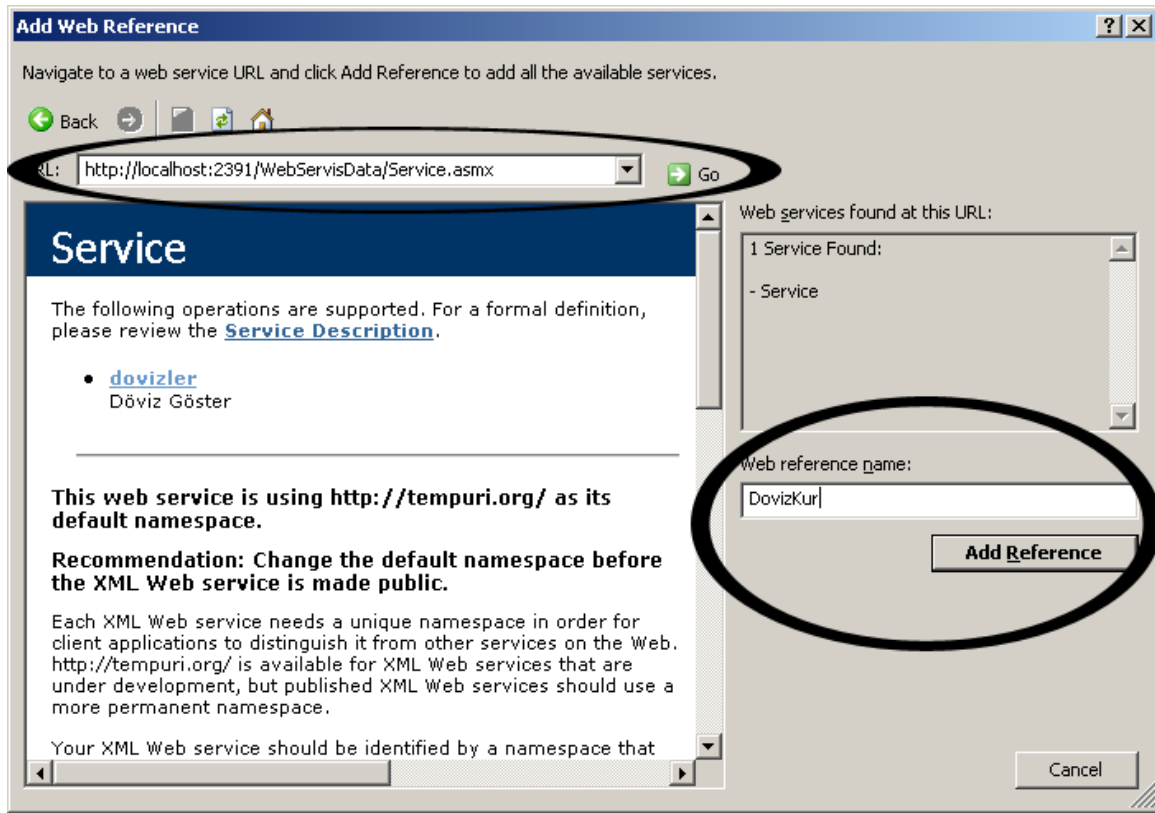
- 4) Kodumuzu yaztıktan sonra çalıştırarak test edelim (Şekil 11.28).



Şekil 11.28

5) Bu bilgileri artık web sayfamızdan bu bilgiyi alabiliriz. Bunun için yeni bir web sayfası başlatalım.

6) Web Site > Add Web Reference seçeneğini tıklayarak web servisimizi ekleyelim (Şekil 11.29).



Şekil 11.29

7) Web sayfamıza bir GridView nesnesi ekleyelim. Form\_Load olayına kodumuzu yazalım.

```
protected void Page_Load(object sender, EventArgs e)
{
    DovizKur.Service kur = new DovizKur.Service();
    DataSet ds = new DataSet();
    ds = kur.dovizler();
    this.GridView1.DataSource = ds.Tables[0];
    this.GridView1.DataBind();
}
```

8) Kodumuzu yazıp çalıştırdıktan sonra GridView nesnemize bilgiler gelecektir (Şekil 11.30).



| ID | Para             | Deger |
|----|------------------|-------|
| 1  | Dolar            | 144   |
| 2  | Euro             | 189   |
| 3  | İngiliz Sterlini | 280   |
| 4  | Japon Yeni       | 25    |

Şekil 11.30

## UYGULAMA 6

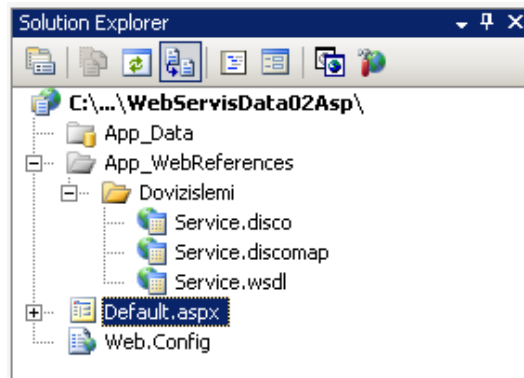
- 1) Bu uygulamamızda isteğimiz para biriminin sonucunu getirelim.
- 2) Bunun için yeni bir web servis projesi başlatalım ve web servisin kod alanına yazalım.

```
[WebMethod(Description="Döviz Türü Seçerek Getir")]
public DataSet dovizsec(string sec) {
    string yol = @"Data Source=Selcuk\SqlExpress;"
    + "initial Catalog=Ogrenci;integrated security=sspi";

    SqlConnection conn = new SqlConnection(yol);
    string sorgu = "select * from tbldoviz where para='"+sec+"'";
    SqlDataAdapter sadp = new SqlDataAdapter(sorgu, conn);
    DataSet ds = new DataSet();

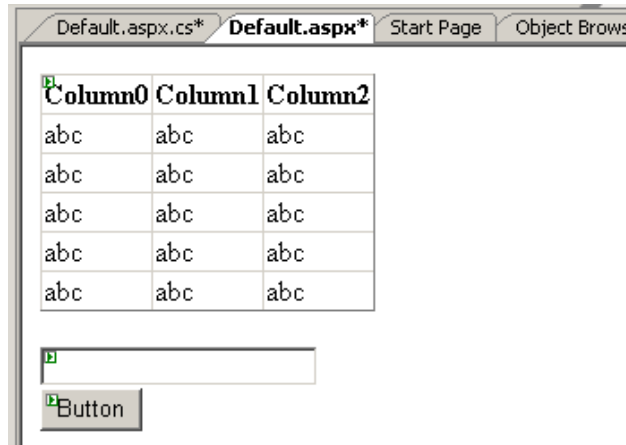
    sadp.Fill(ds);
    return ds;
}
```

- 3) Yeni bir web sayfası açarak Web Servisi referans ediniz (Şekil 11.31).



Şekil 11.31

- 4) Web sayfamıza bir buton, textbox ve gridview ekleyiniz (Şekil 11.32).



Şekil 11.32

- 5) Butonu çift tıklayarak kod alanına geçelim ve kodumuzu yazalım.

```
protected void Button1_Click(object sender, EventArgs e)
{
    Dovizislemi.Service Kurislem = new Dovizislemi.Service();
    DataSet ds = new DataSet();
    ds = Kurislem.dovizsec(this.TextBox1.Text);
    this.GridView1.DataSource=ds.Tables[0];
    this.GridView1.DataBind();
}
```

6) Kodumuzu çalıştırdıktan sonra textbox alanına "Dolar" değerini yazarak butona basınız (Şekil 11.33).



Şekil 11.33

Asmx dosyalarını web servisi hizmeti veren başka bir siteden yolunu yazarakta ulaşabilir ve kodlamanızı yapabilirsiniz. Eğer datalarınızı GridView'e değilde textbox nesnesine aktarmak istiyorsanız;

```
this.TextBox1.Text=System.Convert.ToString
(DataBinder.Eval(ds,"Tables[0].DefaultView.[0].Ad"));
this.DataBind();
```

kodunu yazmanız yeterlidir.

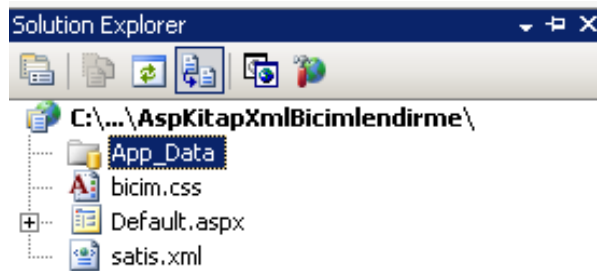
#### XML DOKUMANLARINI BİÇİMLENDİRME

Xml dosyalarımızı baktığımızda ağaç seklinde görüntülendiğini biliyoruz. XML dosyalarımızı biçimlendirerek HTML gibi web sayfası gibi görüntülenmesini sağlayacağız. Bunun için CSS ve XSL kullanabiliriz.

#### CSS İLE XML SAYFALARINI BİÇİMLENDİRME

##### UYGULAMA 7

1) Bir web projesi açalım. Projemize Add New Item seçeneğini kullanarak satis.xml adında bir xml dosya ve bicim.css adında dosyalarımızı ekleyelim (11.34).



Şekil 11.34

2) Satis.xml dosyamızı açalım ve aşağıdaki kodu yazarak dosyamızı oluşturalım.

```
<?xml version="1.0" encoding="utf-8" ?>
<magaza>
  <satis>
    <DVD>Doogal Magic Roundabout</DVD>
    <TURU>Animasyon</TURU>
    <UCRET>35 YTL</UCRET>
  </satis>
</satis>
```

```

        <DVD>Shallow Hal</DVD>
        <TURU>Komedi</TURU>
        <UCRET>40 YTL</UCRET>
    </satis>
</satis>
    <DVD>Planet Earth </DVD>
    <TURU>Belgesel</TURU>
    <UCRET>23 YTL</UCRET>
</satis>
</magaza>

```

### 3) XML kodumuzu yazdıktan sonra CSS dosyamızıda biçimlendirelim

```

satis
{
    display:block;
    margin:10px;
}
dvd
{
    display:inline;
    font-family:"Comic Sans MS";
    color:Green;
}
turu
{
    display:inline;
    font-size:30px;
}
ucret
{
    display:inline;
    color:Red;
    font-size:30px;
}

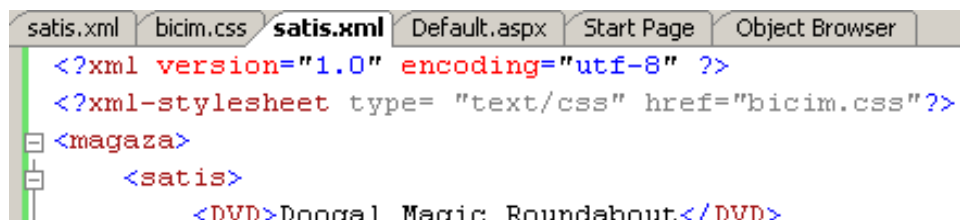
```

Burada yer alan isimlere dikkat ederseniz CSS dosyasında yer alan değişken isimleri XML taglarının aynısı olması gerekmektedir. Böylelikle satis kolonunu ayarladığımız görüntü ve biçimleri ayarlanacaktır.

4) Kodumuzu yazdığımızda ilk olarak çalışmayacaktır.Nedeni ise CSS dosyamızı XML dosyamıza tanıtmamamızdan kaynaklanmaktadır.

5) Bu aşamada XML dosyamızın üst kısmına kodumuzu yazalım (Şekil 11.35).

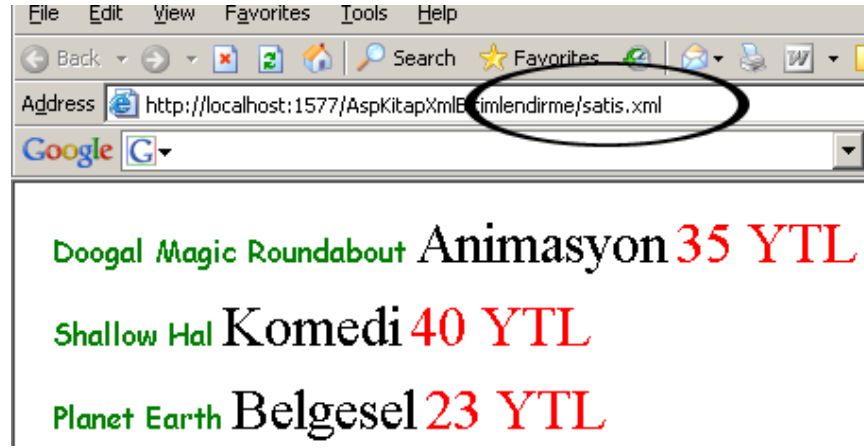
**<?xml-stylesheet type= "text/css" href= "bicim.css"?>**



Şekil 11.35

6) Sonuç olarak artık XML dosyamızın görüntüsü değişecektir. XML dosyamızı çalıştırdığımızda Html gibi biçimlenmiş görüntüsü olacaktır (Şekil 11.36).





Şekil 11.36

### XSL İLE XML DOSYALARIMIZI BİÇİMLENDİRME

CSS de XML dosyalarımızın görüntü, renk gibi işlemleri yapılır. XML biçimlendirilmesinde XSL tercih edilmelidir. CSS de yapılamayan sıralama, yendiden düzenleme, döngüler ve script desteği XSL'in avatajları olarak değerlendirebiliriz. XSL dosyasını XML dosyamıza çağırmak için aşağıdaki satırı yazmamız yeterlidir.

```
<?xml-stylesheet type="text/xsl" href="bizimdosya.xml"?>
```

XSL dosyalarımız temel bir sisteme sahiptir.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

Uygulanacak kalıp

```
</xsl:stylesheet>
```

Burada CSS gibi istediğimiz gibi elementlere biçim uygulaması yapılmaktadır.

Eğer XML dökümanında Root elementi kastediyorsak.

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    .....
  </xsl:template>
</xsl:stylesheet>
```

Burada Root elementini match="/" satırı işletmektedir. Eğer tüm kaynak bilgileri istiyorsak \* işareti kullanmamız yeterli olacaktır.

```
<xsl:template match="*">
  .....
</xsl:template>
```

Belli bir elementin altında bulunan tüm alt elementler için "/" işareti kullanılır.

```
<xsl:template match="//kitap">
  .....
</xsl:template>
```

Belli bir elementin altında bulunan tüm belli bir alt element ve onun altındaki tüm diğer elementler için;

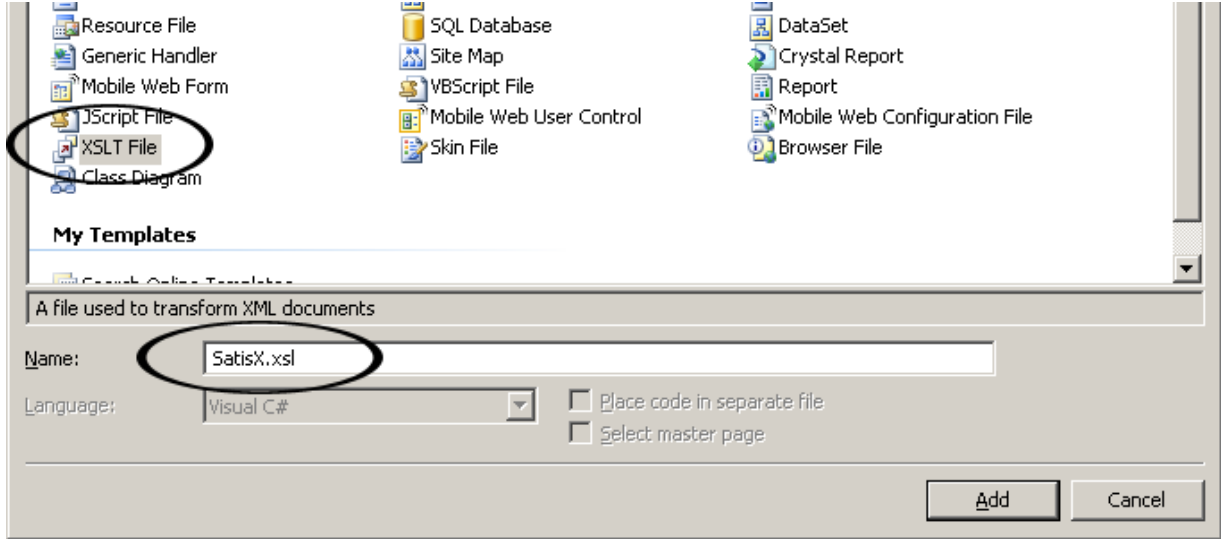
```
<xsl:template match="kitap//ad">
  .....
</xsl:template>
```

Direkt alt element'e uygulamak için;

```
<xsl:template match="kitap/ad">
    .....
</xsl:template>
```

#### UYGULAMA 8

Bu uygulamamızda daha önceki uygulamamız üzerinde devam edeceğiz. XML dosyamızda sıralama yapacağız. Projemize Add New Item seçeneğinden SatisX.xsl adında bir dosya ekleyelim (Şekil 11.37).

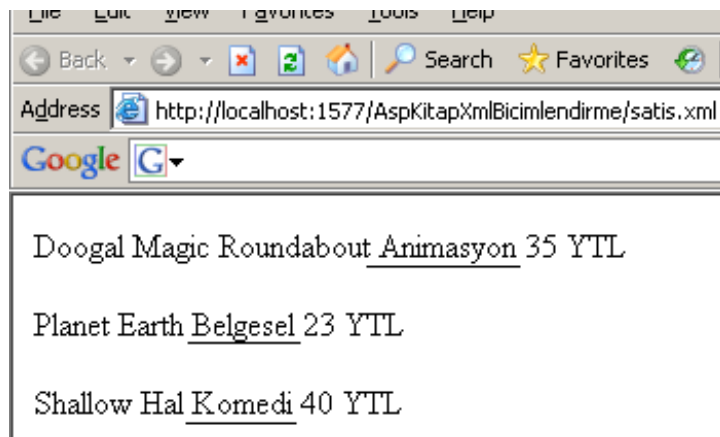


Şekil 11.37

XSL dosyamızı açarak kodumuzu yazalım.

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
    <xsl:apply-templates select="//satis" order-by="TURU" />
</xsl:template>
    <xsl:template match="satis">
        <p>
            <xsl:value-of />
        </p>
    </xsl:template>
</xsl:stylesheet>
```

Buradaki olay TURU kolonuna göre A – Z'ye sıralama gerçekleştirilecektir. Türlerine baktığımızda sıralamanın olduğunu göreceksiniz (Şekil 11.38).



Şekil 11.38



Burada yazarken dikkat etmemiz gerekmektedir. Örneğin: `<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/TR/WD-xsl">` Kısmında Tr şeklinde yazdığında ekranda yazılar gelmeyecektir. Büyük küçük harf duyumunu dikkat ediniz.

## UYGULAMA 9

Bir önceki projemize devam edeceğiz. Xmldeger.xml adında yeni bir XML dosyası oluşturalım.

```
<?xml version="1.0" encoding="utf-8"?>
<Netsoft>
  <Egitmenler>
    <ad>Mehmet</ad>
    <soyad>EĞİTMEN</soyad>
    <sehir>Ankara</sehir>
    <ucret>1500</ucret>
    <tarih>15/10/2006</tarih>
  </Egitmenler>
  <Egitmenler>
    <ad>Selçuk</ad>
    <soyad>ÖZDEMİR</soyad>
    <sehir>Istanbul</sehir>
    <ucret>1450</ucret>
    <tarih>12/09/2005</tarih>
  </Egitmenler>
  <Egitmenler>
    <ad>Cem</ad>
    <soyad>BAYRAKTAROĞLU</soyad>
    <sehir>Ankara</sehir>
    <ucret>1550</ucret>
    <tarih>11/12/2007</tarih>
  </Egitmenler>
  <Egitmenler>
    <ad>Funda</ad>
    <soyad>KOYUNCU</soyad>
    <sehir>Ankara</sehir>
    <ucret>1800</ucret>
    <tarih>07/03/2007</tarih>
  </Egitmenler>
</Netsoft>
```

Projemize maas.xsl adında dosyamızı oluşturalım.

Maas.xsl dosyamıza aşağıdaki kodumuzu yazarak maaşa göre sıralayalım.

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <xsl:apply-templates select="//Egitmenler" order-by="number(ucret)" />
  </xsl:template>
  <xsl:template match="Egitmenler">
    <p>
      <xsl:value-of />
    </p>
  </xsl:template>
</xsl:stylesheet>
```

Burada number(maas) kısmı yazmamızın amacı sayısal değer olarak almamızı sağlar.

Projemize tablo.xml dosyası oluşturalım. XML dosyamıza tablo.xml'i ekleyerek kodumuzu çalıştıralım.

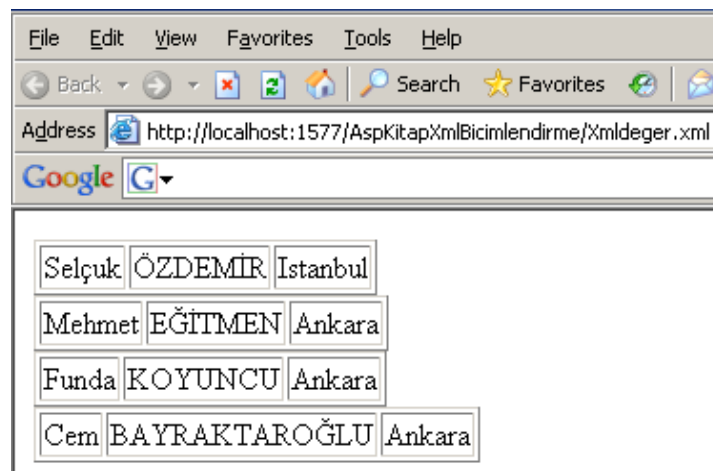
```
<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/TR/WD-xsl">

<xsl:template match="/">
    <xsl:apply-templates select="//Egitmenler" order-by="-ad" />
</xsl:template>

    <xsl:template match="Egitmenler">
        <table border="1">
            <tr>
                <td>
                    <xsl:value-of select="ad"/>
                </td>
                <td>
                    <xsl:value-of select="soyad"/>
                </td>
                <td>
                    <xsl:value-of select="sehir"/>
                </td>
            </tr>
        </table>
    </xsl:template>
</xsl:stylesheet>
```

Sonuç olarak bilgilerimiz "-ad" olarak yazıldığından Z-A'ya sıralanacaktır. Tablo kodlarımızdan dolayıda bilgilerimize tablo içerisinde gelecektir (Şekil 11.39).



Şekil 11.39

## UYGULAMA 10

1) Bu uygulamamızda ise özel bir süzme uygulayalım. XML dosyamızda maaşı 1550 olan kişinin bilgilerini getirelim. Bunu için XSL dosyamızı oluşturalım.

```
<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/TR/WD-xsl">

<xsl:template match="/">
    <xsl:apply-templates select="//Egitmenler" order-by="-ad" />
</xsl:template>

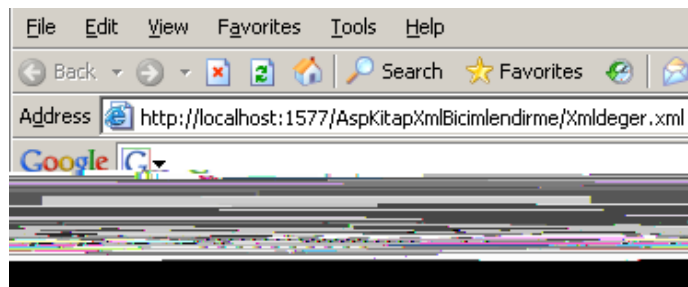
    <xsl:template match="Egitmenler[ucret='1550']">
        <table border="1">
            <tr>
```

```

        <td>
            <xsl:value-of select="ad"/>
        </td>
        <td>
            <xsl:value-of select="soyad"/>
        </td>
        <td>
            <xsl:value-of select="sehir"/>
        </td>
    </tr>
</table>
</xsl:template>
</xsl:stylesheet>

```

Burada önemli olan kısım [ucret= '1550'] kısmında ücreti 1550 olan kişinin bilgileri gelecektir (Şekil 11.40).



Şekil 11.40

## XML BİLGİLERİMİZİ HTML SAYFASINDA GÖSTERMEK

### UYGULAMA 11

1) Projemize sayfa.htm dosyası ekleyelim. HTML dosyamızı açarak kodumuzu yazalım.

```

<!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" >
    <xml id="Egitmen" src="Xmldeger.xml"></xml>
<head>
    <title>Untitled Page</title>
</head>
<body>
<table border=1 DATASRC=#Egitmen>
    <tr>
        <td><span DATAFLD="ad"></span></td>
        <td><span DATAFLD="soyad"></span></td>
        <td><span DATAFLD="sehir"></span></td>
        <td><span DATAFLD="ucret"></span></td>
    </tr>
</table>
</body>
</html>

```

2) <xml id= "Egitmen" src= "XmlDeger.xml"> bizim xmldeger.xml dosyamızı html içerisine çağırabiliyoruz. Table DATASRC kısmında ise id den gelen değeri alabiliriz. DATAFLD ile de ad,soyad,sehir,ucret kısımlarını çekiyoruz (Şekil 11.41).

