

**FATİH TEKNİK VE ENDÜSTRİ MESLEK LİSESİ**

**C# ve ASP.NET**  
**DERS NOTLARI**

**DERS ÖĞRETMENLERİ**

**SEMA AYDOĞDU**

**MUSTAFA ALTINOK**

**HAKKI AYDIN**

**(4062339/Invxdb)**

**KONYA 2010**

# C# İLE ASP.NET

## ASP.NET NEDİR?

Asp.net Microsoft tarafından tasarlanan web uygulama dilidir. Programcılar asp.net kullanarak dinamik web siteleri, web uygulamaları ve XML web servisleri geliştirebilirler. Asp.net .Net platformunun bir parçasıdır ve asp nin devamı olarak nitelendirilmektedir.

Bu Framework .NET kütüphanesinin kullanıldığı anlamına gelir yani daha hızlı ve verimli uygulamalar geliştirebilmemizi ve bunları istediğimiz programlama diliyle yazabilmemizi sağlar.

Asp.net de programlanan web sitelerinin uzantısıdır. Eğer bir web sayfasının uzantısı aspx ise bu demektir ki, bu web sitesi Asp.net kullanarak tasarlanmıştır.

### Asp.Net ile çalışmak için istenilen programlar:

Microsoft Visual Studio 2005 (tasarımı ve kodları yazmak için)

Microsoft SQL Server 2005 (veritabanı işlemleri için)

### ASP.NET' in Özellikleri

- Dinamik Web Uygulamaları Geliştirilmektedir.
- .NET Framework platformunun çok önemli bir bölümüdür.
- Web uygulamaları çok kısa bir zamanda geliştirilebilmektedir.
- Uygulama geliştirmekte birçok dil kullanılabilir.
- Uygulama geliştirmede içerik ve kod bir birinden farklı dosyalarda yer alabilir.
- Yazılan kodlar hemen derlenebilmektedir.
- Nesneye yönelik programlama yapılabilmektedir.

## .NET FRAMEWORK NEDİR?

.NET Framework, Microsoft tarafından geliştirilen, açık İnternet protokolleri ve standartları üzerine kurulmuş komple bir uygulama geliştirme platformudur. Buradaki uygulama kavramının kapsamı çok geniştir. Bir masaüstü uygulamasından tutun, bir web tarayıcı uygulamasına kadar her şey bu platform içinde düşünülmüştür ve desteklenmiştir.

Bu uygulamaların birbirleriyle ve geliştirildiği ortam fark etmeksizin dünyadaki tüm uygulamalarla iletişimi için kolayca Web Servisleri oluşturulmasına imkân verilmiştir.

## C# Programlama Dili Temelleri

C#, güçlü, modern, nesne tabanlı ve aynı zaman type-safe(tip-güvenli) bir programlama dilidir C# kullanmak için CLR ve .NET Framework sınıf kütüphanesine ihtiyacımız vardır. Bu da demek oluyor ki C#, JAVA, VB ya da C++ değildir. C,C++ ve JAVA' nın güzel özelliklerini barındıran yeni bir programlama dilidir. Sonuç olarak C# ile kod yazmak hem daha avantajlı hem daha kolay hem de etkileyicidir.

### Değişkenler (Variables)

Bilgisayarlar, kullanıcılar tarafından kendilerine verilen bilgileri, verileri hafıza adı verilen birimlerinde tutarlar. Hafızada tutulan bilgilerin kimisi sabit değerken, kimi bilgilerin değerleri değişkendir.

Programda bir değişken belirlendiğinde (değişken atandığında), bilgisayar bu değişkenin değerinin tutulması için hafızada bir alan, hücre belirler. Değişkenin aldığı tüm değerler bu hücreye yazılır, program içinde o değişken her çağrıldığında bu hücredeki değer getirilir.

## HAFIZADAKİ DEĞER DEĞİŞKEN İSMİ

Hafıza alanı 1  
Hafıza alanı 2  
Hafıza alanı 3  
Hafıza alanı 4  
Hafıza alanı 5

Ali	Ad
Barış	Soyad
Gül	AnneAd
Veli	BabaAd
Konya	DogumYeri

Program yazarken çeşitli değişkenler kullanılır. Programda hangi değişkenlerin kullanılacağı değişken atama işlemiyle yapılır. Değişken atandıktan sonra ona bir ilk değer verilir, program akışı esnasında değişkenin değeri değiştirilebilir, var olan değeri silinebilir, bir değişken içindeki bir değer başka bir değişkene atanabilir.

Bir değişkenin yaşam döngüsünü aşağıdaki şekilde ifade edebiliriz.

- Değişken atanır.
- Değişkene ilk değeri verilir.
- Değişken çağrılır, gerektiğinde içeriği değiştirilir.

Bilgisayarda tutulacak değerlerin, verilerin özellikleri farklı olduğundan veriler veri tiplerine (türlerine) ayrılmıştır. Örneğin, sayısal değerleri tutmak için ayrı, metin bilgilerini tutmak için ayrı veri tipleri kullanılır. Tam sayılar için integer veri tipinin kullanılması uygunken, “Millî Eğitim Bakanlığı” ifadesi için string (dize, dizgi) veri tipinin, günün tarihini tutmak için Date veri tipinin kullanılması gerekir.

Değişken tanımlama işlemi sırasında değişkene bir isim verilirken, değişkenin hangi tipte veri tutacağı da belirtilir. Benzer veri tipleri arasında işlemler yapılabilir. Örneğin, “integer” veri tipindeki 2 değişken içindeki değerler toplanabilir, fakat içeriği “Ahmet” olan “string” veri tipli bir değişkenle, içeriği 20 olan “integer” veri tipli bir değişken toplanamaz.

C# taki adı	Açıklama	Max ve Min aralık yada değeri
sbyte	8 bit işaretli tamsayı	-128 : 127
short	16 bit işaretli tamsayı	-32.768 : 32.767
int	32 bit işaretli tamsayı	-2.147.483.648 : 2.147.483.647
long	64 bit işaretli tamsayı	-9.223.372.036.854.775.808 : -9.223.372.036.854.775.807
byte	8 bit işaretli tamsayı	0 : 255
ushort	16 bit işaretli tamsayı	0 : 65.535
uint	32 bit işaretli tamsayı	0 : 4.294.967.295
ulong	64 bit işaretli tamsayı	0 : 18.446.744.073.709.551.615
float	32 bit tek kayan sayı	+yada - $1,5 \cdot 10^{-45}$ : + ya da - $3,4 \cdot 10^{38}$
double	64 bit çift kayan sayı	+yada - $5 \cdot 10^{-324}$ : + ya da - $1,7 \cdot 10^{308}$
decimal	128 bit ondalıklı sayı	+yada - $1,5 \cdot 10^{-28}$ : + ya da - $7,9 \cdot 10^{28}$
bool		true ya da false
char	Karakterleri temsil eder	16 Unicode karakterleri

### 1. Tam Sayı Veri Tipleri

Tam sayılar ondalık kısım içermeyen sayılardır. Örneğin, 10, 2450, -45, 245678 sayıları tam sayılardır. Gündelik hayattan tam sayı değerler içeren durumlar aşağıda sıralanmıştır:

Sınıf mevcudu, okuldaki öğrenci sayısı, bir sınıftaki kız öğrenci sayısı, boy uzunluğu (cm olarak, örneğin 175 cm).

### 2. Ondalık Sayı Veri Tipleri

6.8, 3.56, 6.9876 gibi sayılar ondalık sayıları oluşturmaktadır. Yaşantımızda birçok ondalık sayı içeren durumlar vardır. Örneğin; Boy uzunluğu (m olarak, örneğin 1.75 m), maaş miktarı (930.25 YTL)

### 3. String Veri Tipi

String, bir karakter dizisini, grubunu ifade eder. Her türlü karakter grubu bir string oluşturabilir. Örneğin, “Seda”, “Ev Adresiniz”, “235” ifadeleri.

“235” ifadesi tam sayı tipinde mi yoksa bir string tipinde mi diye düşünülebilir? Eğer bir sayı üzerinde matematiksel işlemler yapılmıyorsa o string tipinde tanımlanabilir.

Örneğin, telefon numaraları tamamen sayısal ifadelerden oluşur fakat telefon numaraları üzerinde matematiksel işlemler yapmayız. Hiçbir zaman iki telefon numarasının toplanması, çıkarılması ihtiyacı duyulmamıştır. Okul öğrenci numaraları da birer sayı olmalarına rağmen, bu numaralar üzerinde de aritmetiksel işlemler yapılmaz. Bu nedenle telefon numarasını ve öğrenci numarasını tutacak değişken string tipinde tanımlanabilir. Hangi tipte tanımlandığında hafızada az yer kaplıyorsa o tip seçilmelidir.

Ayrıca sayılardan oluşan bir string tipindeki değişken ile tam sayı tipindeki bir değişken arasında aritmetik bir işlem yapılamaz. Çünkü değişkenlerin veri tipleri birbirinden farklıdır.

### 4. Char Veri Tipi

Char (Character) veri tipi, bir karakterlik veri tutulmasını sağlar. Bilgisyardaki her harf, rakam, sembol (a, C, d, 1, 7, ^, \$, [, ?) karakter olarak isimlendirilir.

Bilgisayar dünyasında kelimeler arasındaki boşluklar da karakter olarak değerlendirilir. Boşluk karakteri de diğer karakterler gibi bilgisayarın hafızasında yer kaplar. Char veri tipindeki bir değişkenin içeriğinde ara (boşluk) tuşuna basılarak üretilen boşluk karakteri tutulabilir.

### 5. Boolean Veri Tipi

Gündelik yaşamda doğru-yanlış, açık-kapalı, evet-hayır, 1-0 gibi birbirinin karşıtı olan durumlar vardır. Boolean veri tipi bu tür durumlarda tercih edilir. Boolean veri tipi sadece True (Doğru) ve False (Yanlış) değerlerinden birini alabilir. Örneğin, “onay” isimli bir değişkenin değeri ya “True”dur, ya da “False”dur. Bu değişken iki değere aynı anda sahip olamaz, örneğe göre düşünürsek bir işlem ya onaylanmıştır ya onaylanmamıştır.

### Değişken Tanımlama Kuralları

- Değişken ismi bir harf ile başlamalıdır.

```
string Ad1; //Doğru
```

```
string 1Ad; //Yanlış
```

- Değişken isminde boşluk bulunamaz.

```
string Ad_Soyad; //Doğru
```

```
string Ad Soyad; //Yanlış
```

- Değişken isminde sadece harfler, rakamlar ve alt çizgi karakteri bulunabilir.

```
string ali_veli; //Doğru
```

```
string ali+veli; //Yanlış
```

- Değişken ismi C# komutu olmamalıdır.

```
string true; //Yanlış
```

### Tür Dönüşümleri

Program içerisinde değişkenlerle ilgili tür dönüşümleri yapmak durumunda kalabilirsiniz. Örneğin, sayısal bir veriyi string değişken türüne, string bir veriyi sayısal değişken türüne dönüştürmeniz gerekebilir.

## Convert

Convert metoduyla dönüştürme işleminde derleyici tarafından izin verilen tüm türlere dönüştürme işlemi yapılabilir. Dönüştürme işlemi yapmadan önce dönüştürülecek bilginin hangi türlere dönüştürülebileceğine dikkat edilmelidir. Convert metodunu kullanırken; Convert yazıp nokta karakteri yazıldığında açılan menüden istenilen dönüşüm türü seçilerek dönüştürülecek bilgi parantez içinde yazılmalıdır.

### Örnek:

```
Int x;  
X= Convert.ToInt32(textBox5.Text);
```

## Aritmetiksel Operatörler

Bu operatörler aritmetik işlemlerinde, sayılarla veya sayı tutan ifadelerle kullanılır.

- Çarpma 

```
int sayi = 100;  
sayi = 200 * 2;
```
- Bölme 

```
double bolum;  
bolum = sayi / 23;
```
- Çıkarma 

```
int sonuc = bolum - 100;
```
- Toplama 

```
int toplam;  
toplam += sonuc; // Bu ifade, "toplam = toplam + sonuc" ile aynı anlama gelir
```

İPUCU: Aritmetik operatörleri, eşittir ifadesi ile beraber kullanılırsa, işlem değişkenin kendisi ile yapılır.

- Mod alma 

```
int kalan = toplam % 42; // Sonuç, toplam değişkenindeki değerin 42 ile bölümünden kalan sayıdır.
```

## Karşılaştırma Operatörleri

Bu operatörler veri tiplerini birbirleriyle karşılaştırmak için kullanılır. Bu operatörler ile yapılan işlemlerin sonucunda **true** ya da **false** değeri döner. Karşılaştırma operatörleri yalnızca sayı tipleri üzerinde yapılmaz.

- Küçük 

```
double sayi = 1.5;  
float sayi2 = 1.3;  
sayi2 < sayi // Sonuç: True
```
- Küçük Eşit 

```
sayi2 <= sayi // Sonuç: True
```
- Büyük 

```
sayi2 > sayi // Sonuç: False
```
- Büyük Eşit 

```
sayi2 >= sayi // Sonuç: False
```
- Eşit 

```
sayi2 == sayi // Sonuç: False
```
- Eşit Değil 

```
sayi2 != sayi // Sonuç: True
```

## Koşul Operatörleri

Veri tipleri ve değişkenler üzerinde kontrol yapılırken birden fazla koşula ihtiyaç duyulabilir. Bu durumda, koşulları birbirleriyle karşılaştıracak operatörler kullanılır. Bu kontrollerden dönen değerler **Boolean** tipinde olduğu için, koşul operatörleri de bu değerler üzerinde işlem yaparlar.

## && (And)

Bu ifade, verilen koşulların kesişimini alır. Eğer tüm koşulların değeri **true** ise sonuç da **true** olur. En az bir tane **false** değeri olan koşul varsa, sonuç **false** olur.

## || (Or)

Bu ifade, verilen koşulların birleşimini alır. Eğer tüm koşulların değeri **false** ise sonuç **false** olur. En az bir tane **true** değeri varsa sonuç **true** olur.

## ! (Değil)

Bir koşulun değerini tersine çevirir. Koşul **false** ise **true**, **true** ise **false** olur.

## Karar Yapıları ve Döngüler

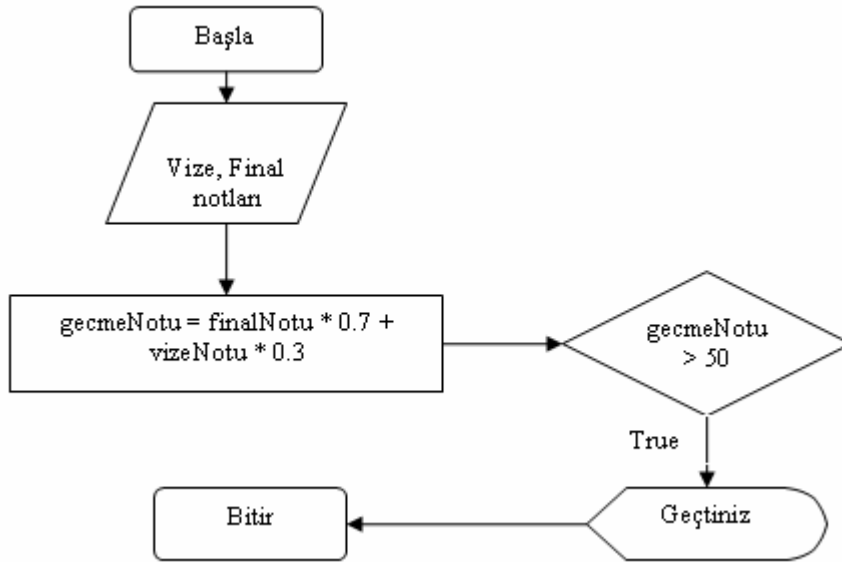
### İf-else:

**if** karar yapısı, bir koşul sağlandığı zaman yapılacak işlemleri kapsar. Kontrol edilecek koşul ifadesinin sonucu **true** değerini alırsa, küme parantezleri arasındaki kodlar çalıştırılır.

### if(Koşul)

```
{  
  
}
```

Örnek: Vize ve final notunu kullanıcıdan aldıktan sonra, geçme notunu hesaplayan ve notun elliden büyük olması durumunda ekrana “geçtiniz” mesajını çıkartan algoritma.



```
double gecmeNotu;  
int finalNotu = Convert.ToInt32(textBox1.Text);  
int vizeNotu = Convert.ToInt32(textBox2.Text);  
gecmeNotu = finalNotu * 0.7 + vizeNotu * 0.3;  
if(gecmeNotu > 50)  
{  
    MessageBox.Show("Geçtiniz tebrikler...");  
}
```

**else** ifadesi, **if** yapısındaki koşulun sağlanmadığı bütün durumlarda devreye girer.

```
if(Koşul)
{ //Diğer kodlar }
else
{ //Diğer kodlar }
```

## Döngüler

“for” Döngüsü

**For** döngüsü bir işlemin belirli sayıda yapılması için kullanılır. for döngüsünün yapısı aşağıdaki gibidir.

```
for(başlangıç işlemleri; devam şartları; ara işlemler)
{ komutlar; }
```

```
for (int i = 0; i < 10; i++)
{
    MessageBox.Show(“Merhaba”);
}
```

Bu döngüde 3 parametre vardır.

- İlk parametre sayacın başlangıç değerini belirler. Örnekte, sayaç değişkeni tanımlanıp 0 değeri atanmıştır.
- İkinci parametre bir koşul ifadesidir. Bu koşul sağlandığı sürece döngü devam eder. Örnekteki döngü, i değeri 10dan küçük olduğu sürece devam edecektir.
- Üçüncü parametre, her döngüden sonra yapılması gereken işlemi belirtir. Örnekte, her işlemten sonra i değeri bir artırılır.

for döngüsü devam şartları sağlandığı sürece komutları işlemektedir. Başlangıç işlemleri sadece bir defa çalışmaktadır. Devam şartları bool (true veya false) türünden olmaktadır. Aşağıdaki örneği inceleyelim: 1’den 10’a kadar olan sayıları toplayan program.

```
int i, toplam;
toplam=0;
for ( i = 1; i <=10; i ++ )
{
    toplam=toplam+i;
}
```

## “while” ve “do while” Döngüleri

while döngüsünde ilk önce şart kontrol edilir. Eğer şart sağlanıyorsa parantez içindeki komutlar işlenir.

```
int i=0, toplam=0;
while (i <= 10)
{
    toplam=toplam+i;
    i++;
}
Label1.Text=toplam.ToString();
```

do while döngüsünde ise şart kontrolü döngü sonunda yapılır. Bunun için şart sağlanmıyor olsa bile döngü en az bir kere çalışır.

```
int i=0, toplam=0;
do
{
toplam=toplam+i;
i++;
}
while (i <= 10);
```

## Diziler

Dizi deęişkenleri, aynı tipte birçok veriyi bir arada tutmayı sağlar. Benzer işlemlerde kullanılan deęişkenler bir dizi altında listelenebilir. Örneęin kullanıcıdan alınan isimler **String** tipinde bir dizi içinde toplanabilir.

```
string [] isimler;
```

Dizilerin kaç eleman içereceęi, dizi tanımlanırken ya da daha sonra belirtilebilir:

```
string [] isimler = new string[10];
```

```
string [] isimler;
```

```
// ...
```

```
isimler = new string[4];
```

Dizilerin indisleri sıfırdan başlar. Örnekteki isimler dizisinin 4 tane **String** tipinden elemanı vardır.

Dizilerin elemanlarına ulaşmak için, istenilen elemanın indisi verilmesi gerekir.

```
isimler[0] = "Ali";
```

```
isimler[1] = "Ahmet";
```

```
isimler[2] = "Mehmet";
```

```
isimler[3] = "Ayşe";
```

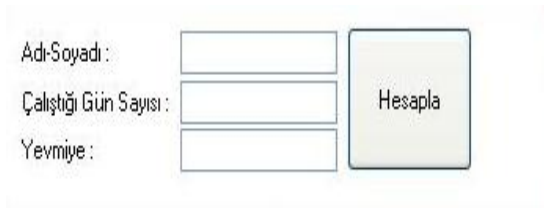
```
MessageBox.Show(isimler[3]);
```

Dizilere tek tek deęer atanabildięi gibi, tanımlarken de başlangıç deęerleri atanabilir.

**string [] isimler = {"Ali","Ahmet","Mehmet","Ayşe"};**

### Örnek Uygulamalar:

Bir işçinin yevmiye sini hesaplayan form tasarlayınız.(örneęin 15 günlük yevmiye si 15\*50 TL=750 TL)



```
int gsayi , yevmiye;
gsayi=Convert.ToInt32 (textBox2.Text);
yevmiye= gsayi* 50;
textBox3.Text=Convert.ToString(yevmiye);
```

Bir sayının tekmi, çiftmi olduğunu gösteren form tasarlayınız.



Sayı Giriniz

```
int sayi, mod ;
sayi=Convert.ToInt32 (textBox1.Text);
mod= sayi % 2;
if ( mod =1)
{
    Response.Write(“Sayı Tektir.”)
}
else
{
    Response.Write(“ Sayı Çifttir.”)
}
```

## ASP.NET WEB UYGULAMALARI

Web uygulamaları, **.aspx** uzantılı arayüz dosyası ve **.aspx.cs** uzantılı kod dosyalarından oluşur. Örneğin **default.aspx** isimli ASP.NET sayfasının, sunucu taraflı kodları **default.aspx.cs** isimli dosyada tutulur.

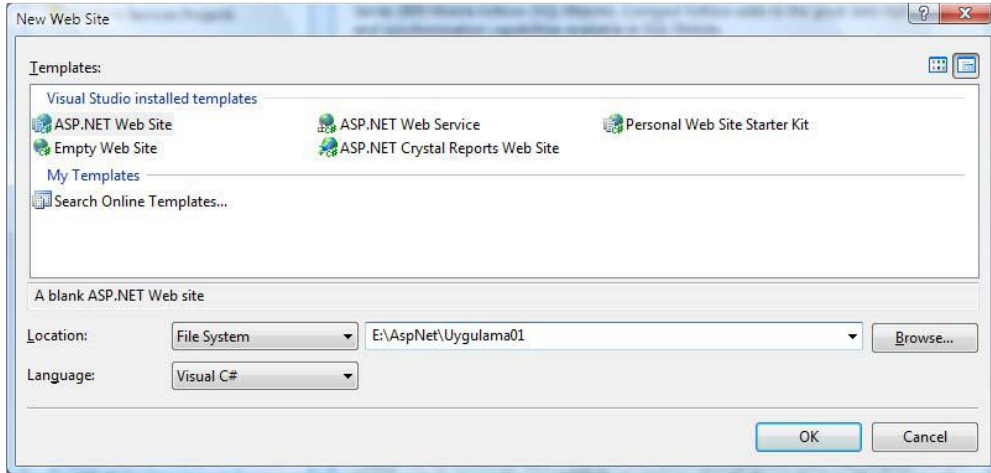
Kullanıcı arayüz sayfası ve kod sayfasının ayrı tutulmasının yararı, web programcısına ve web tasarımcısına ayrı kaynaklar sunarak bağımsız çalışma ortamı sağlamaktadır.

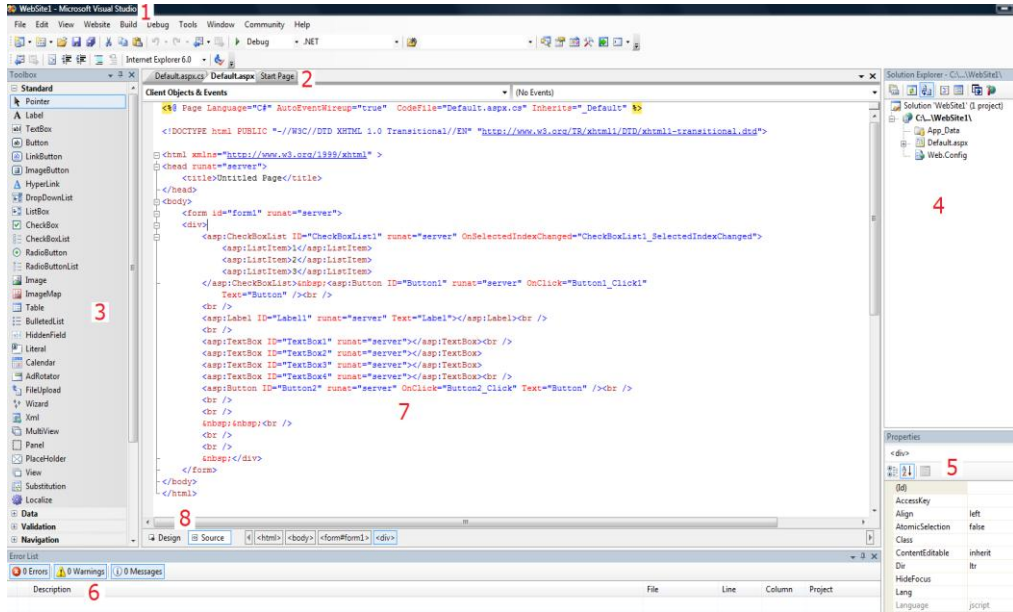
İlk olarak yeni bir çalışma sayfası açalım **File > New > Web Site** seçeneğini tıklarız ve karşımıza New Web Site penceresi gelecek buradan değiştirmeniz gereken yerleri ayarladıktan sonra **OK** düğmesine tıklarız. Asp.Net sayfamız açılacaktır.

Templates: **ASP.NET Web Site** işaretli olmalıdır,

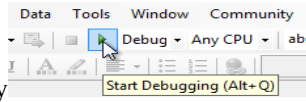
Location: **File System** seçilir, **Browse...** Düğmesine tıklanarak dosyaların kaydedileceği yer belirlenir,

Language: **Visual C#** programlama dili seçilir.

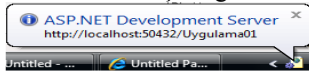




Sol tarafımızda Toolbox'ımız yer almaktadır bütün nesnelere bu panelde yer almaktadır. Sağ Üst tarafta Solution Explorer bulunmaktadır burada sitemizin her şeyi yer alır yani alt sayfalar, resimler, css dosyaları, v.b burada görüntülenir değiştirilebilir. Sağ Alt tarafta Properties paneli yer almaktadır burada seçili olan nesnenin özellikleri yer almaktadır nesne ile ilgili tüm değişiklikleri bu panelden yapabiliriz. Artık sayfanın sol alt tarafında bulunan design tuşunu kullanarak tasarım penceresini açabilir ve uygulamamıza başlayabiliriz...



Hemen üst tarafta bulunan play işareti veya **F5** tuşu sayesinde uygulamalarımızı çalıştırabiliriz. Uygulamalarımızı başlattığımızda karşımıza gelen soruyu **OK** düğmesine tıklayıp onayladıktan sonra



taskbarımızın üstünde çıkan **Visual Web Developer Web Server** ifadesi web uygulamalarını geliştirmek için kullanılan **local Web Server** olup, IIS yapısının ekstra özelliklerini içermez. Önceleri IIS de ayarlama sorunları yaşadığımız olmuştur. .NET 2005 platformu kendi IIS ini çalıştırdığından IIS bazındaki sorunlar ortadan kalkmıştır.

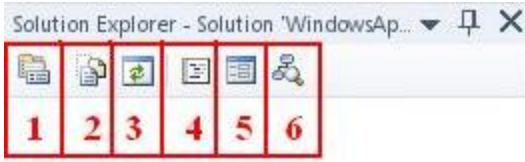
**1) Uygulama Başlığı:** Oluşturduğumuz uygulamaya verdiğimiz ad birçok programdan da alışık olduğumuz gibi sol en üst köşede gözükür.

**2) Tab Pages (Sayfa Sekmeleri):** Bu editörde açık sayfalar Windows ayar pencerelerinden alışık olduğumuz *sekmeler halinde gözükür*. Açık olan sayfalardan istenilene ilgili sayfaya ait sekmeye tıklanarak geçilebilir.

**3) Toolbox:** Uygulamamıza ekleyeceğimiz araçları (componentleri) seçeceğimiz pencere **Toolbox** penceresidir. Bu pencerede bulunan araçları projemize sürükleyip bırakma ile dahil edebiliriz. Toolbox penceresini ekranda göremiyorsanız yukarıdaki menülerden View → Toolbox'ı seçerek o pencerenin ekranda görülmesini sağlayabilirsiniz.

- ❖ Standart
- ❖ Data
- ❖ Validation
- ❖ Navigation
- ❖ Login
- ❖ WebParts
- ❖ HTML
- ❖ Crystal Reports

**4) Solution Explorer (Proje Gezgini):** Bu pencere sayesinde uygulamanızda bulunan dosyaları görebilirsiniz. Ayrıca dosyaları yeniden isimlendirebilir, yer değişikliği yapabilir ya da silebilirsiniz. Bunu etkin kullanabilmek için pencereye ait özellikleri incelememizde fayda var.



**1-Properties:** Solution Explorer penceresinde seçilen forma ait özellikleri gösteriyor.

**2-Show All Files:** Açık olan uygulamadaki bütün dosyaları gösteriyor.

**3-Refresh (Yinele):** Uygulamanın içinde bulunan sayfaları gösteriyor. Bu daha çok uygulama dışında klasöre bir şey eklendiğinde onu Pencere gezgininde görebilmemiz için kullanılır.

**4-View Code (Kodu Göster):** Seçili formun koduna geçiş yapmanızı sağlar.

**5-View Designer (Tasarım Kısmını Göster):** Seçili formun tasarım tarafına geçiş yapmanızı sağlar.

**6-View Class Diagram (Sınıf Diyagramını Göster):** Seçili formun sınıf yapısını diyagram şeklinde gösterir. Yeni başlayanlar için henüz anlamsız gelebilir ama işe yarayan bir buton.

**5) Properties window (Özellikler Penceresi):** Sayfanın üzerine yerleştirdiğimiz her bir nesnenin kendine ait özellikleri bulunur. Örneğin bir butonu düşünelim; butonun bir genişliği, uzunluğu vardır ve genelde üzerinde bir yazı yazar; işte bunlar butonun özelliğine dair birer örnektir. Butonun bu ve daha birçok özelliğini bu pencereden görebiliriz. Eklemiş olduğunuz herhangi bir nesnenin özelliklerine o nesneyi seçerek ulaşabilirsiniz. Özelliğini görmek istediğiniz nesneye bir kez tıklayın, **properties penceresinde** o nesneye ait özelliklerin açıldığını göreceksiniz. Bu pencereye özelliklerini görmek istediğiniz nesnenin üzerine sağ tıklayarak, açılan menüden Properties seçeneğini seçip ulaşabilirsiniz.

**6) Error List (Hata Listesi):** Uygulamalarda üç çeşit hata ile karşılaşılabilir. Bunlar:

**Syntax Error** (Yazım kurallarına aykırılıklar),

**Runtime Error** (yürütme esnasında beliren hatalar)

**Logic Error** (Mantık hataları) dır.

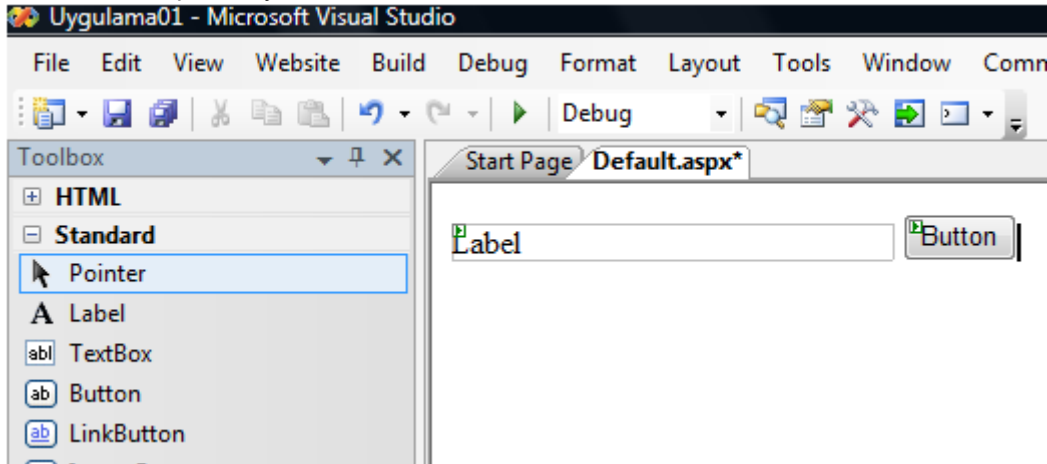
Vs 2005'in sunduğu kolaylıklardan biri de hata bulmada. Syntax hataları henüz daha derlemeye geçilmeden hata olan yerin altı kırmızı ile çizilir ve hatanın kaynaklanabileceği sebep Hata Listesinde gözükür. Örneğin

**C# (s şarpta)** satır sonlarına ; **(noktalı virgül)** koymak zorundayız, bu c#' ın yazım kurallarındandır. Yazım esnasında bir satır sonuna; koymadan alt satıra yazmaya başladığımızda ekranda altı kırmızıçizgili ifadelerle karşılaşırız. bu derlemeye geçmeden bize orada bir yazım hatası olduğunu gösterir. Hatanın neden kaynaklanabileceğini görmek istediğimizde **Error List** ' e bakabilir ya da fare ile kırmızıçizgilerin üzerine gelip çıkan alt bilgidен faydalanabiliriz. Runtime Error' lar derleme esnasında yakalanan hatalardır, buna örnek olarak **int a=3/0** ' ı ya da dönüştürme hatalarını verebiliriz. Bu tür hataların altı mavi çizgi ile çizilir ve Error List' te bu hatanın neden kaynaklandığını görebiliriz. Mantık hataları için ise Break Points kullanarak programı adım adım izleme yöntemi kullanılır.

**7-8) Design and Source buttons:** **Design** sayfamızın tasarım aşamasını, **Source** ise HTML kodunu gösterir İlk açıldığında Source seçili durumdadır. Biz burada Design butonuna tıklayarak tasarım görünümüne geçiyoruz.

## ASP.NET NESNELERİ

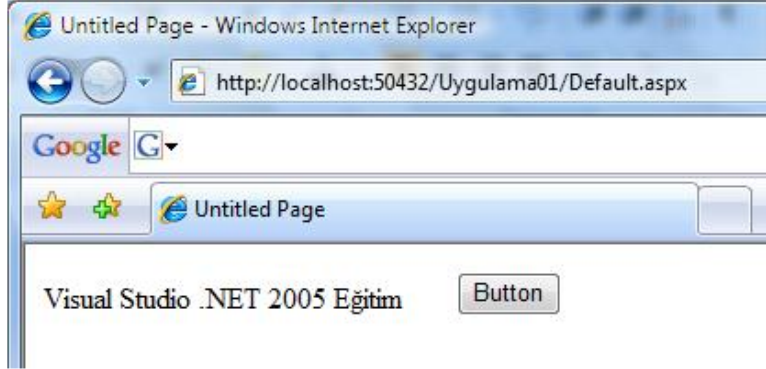
Sayfamıza Toolbox penceresinden Design kısmında iken sayfamıza bir label ve buton ekleyelim. Ekledikten sonra butonumuzun üstünü çift tıklayalım.



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.



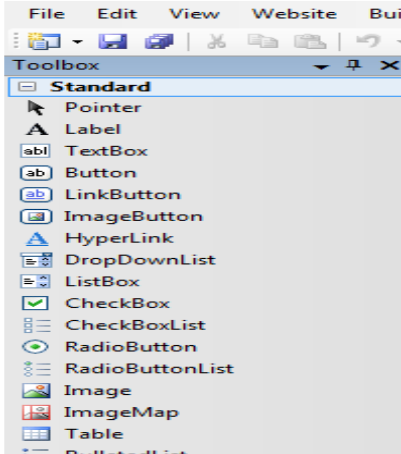
### \_ Server Kontrol

Kullanıcı arabirimi veya diğer ilişkisel fonksiyonlar oluşturmak üzere Server üzerinde çalışan componentlere denir. Başka bir deyişle Web formlarda kullanılan kontrollerin sunucu ile haberleşerek çalışabilmesi için `runat="server"` bildiriimi kullanılır. Server kontroller **runat="server"** özelliğine sahiptirler.

```
<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>
```

## STANDART KONTROLLER

ASP.Net platformumuzu açtığımızda **Toolbox** penceresindeki **Standart** bölümündeki kontrollerdir.



### ❖ Label

Web sayfasında kişilere mesaj iletmek için kullanılır. 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

#### ❖ Textbox

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

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

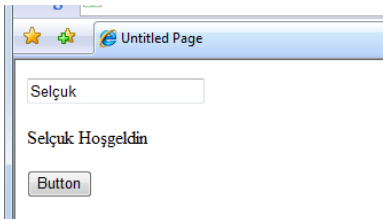
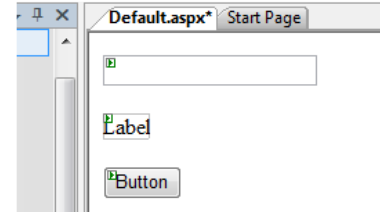
Tüm Windows uygulamalarında Tamam, Uygula, İptal vb. olarak gördüğümüz basma (click) özelliğine sahip kontroldür. Web sayfalarımızda bir işlemi onaylatmak, form göndermek, kayıt eklemek, giriş yapmak gibi işlemlerde kullanacağız.

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

#### Örnek Uygulama:

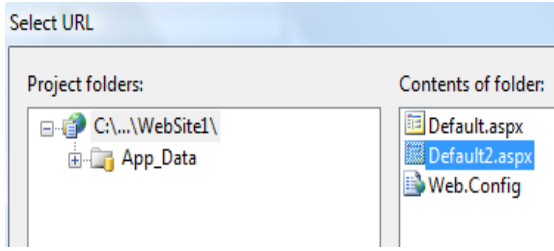
Butonumuzun üstünü çift tıklayarak Button1\_Click özelliğine aşağıdaki kodu yazalım.

```
protected void Button1_Click(object sender, EventArgs e)
{
    this.Label1.Text = this.TextBox1.Text + " Hoşgeldin";
    //Label1 in yazımı textbox a yazılan yazıyı yaz.
}
```



#### ❖ Link Button

Birçok özelliği butona benzeyen Link butonu link vermek için kullanılır. Click özelliğine kodumuzu yazarak yönlendirme yapabiliriz. Link vermek için Properties penceresinden “PostBackUrl” seçeneği belirlenir.



#### ❖ Image Button

LinkButton ile işlevi aynıdır. Tek farkı Link yazısının yerine resim gelir. Resim getirmek için Properties penceresinden “ImageURL” seçeneğinden resmi belirlenir.

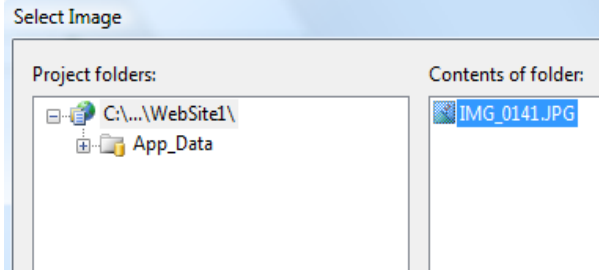


Image Button’a resim ekledikten sonra bir sayfaya yönlendirmek için ImageButton’u çift tıklayınız. Click özelliğine aşağıdaki kodu yazınız. Kodumuzu yazıp çalıştırdıktan sonra resmi tıkladığımızda web sayfası açılacaktır.

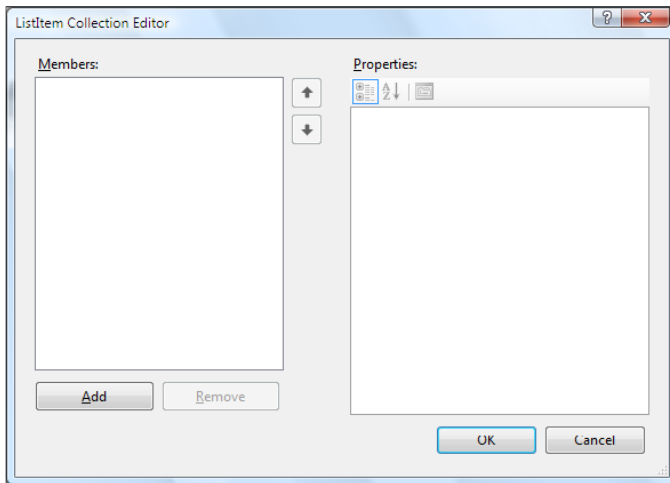
```
protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect("http://www.netsoft.com.tr");
}
```

**Bu kod da sayfaya giriş yapıldığı an yazılan adrese yönlendirme yapar**

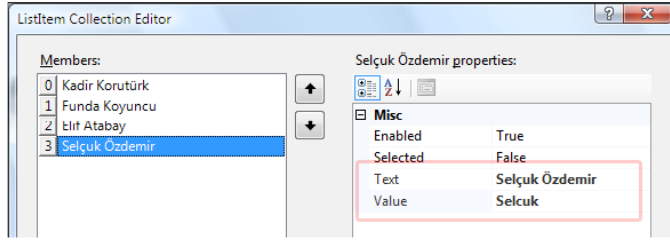
#### ❖ Dropdownlist

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

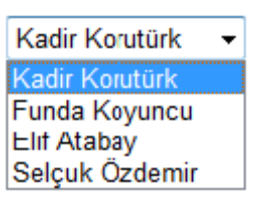
- 1) Yeni bir Web Sayfası açalım.
- 2) Toolboxdan DropDownList ekleyelim.
- 3) DropDownList’i seçtikten sonra Properties penceresinden Items seçeneğini tıklayınız. Karşımıza ListItems Collection Editor açılacaktır.



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



5) Text özelliğine yazdığımız değerler DropDownList de ekranda görünecektir.

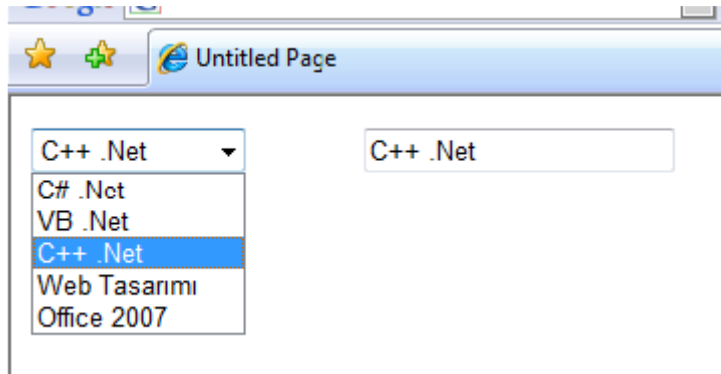


### Örnek uygulama:

- 1) Yeni bir Web sayfası açalım. Web sayfamıza TextBox ve DropDownList ekleyelim.
- 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 değerın TextBox da görünmesi sağlanacaktır.



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

## AutoPostBack Özelliği

Bir sunucu kontrolünün web sunucuya otomatik olarak bilgi göndermesini sağlar.

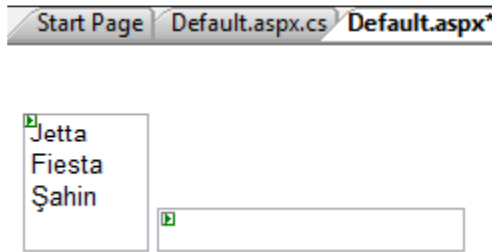
- Bu özelliği destekleyen kontroller;
  - DropDownList,
  - ListBox,
  - CheckBox,
  - CheckBoxList,
  - RadioButton,
  - RadioButtonList,
  - TextBox
  - Button

### ❖ Listbox

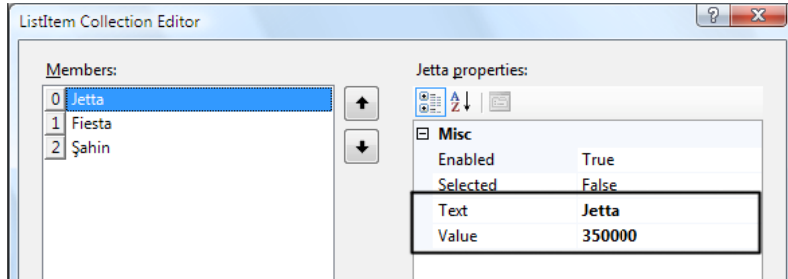
Tüm özellikleri DropDownList ile aynıdır. Tek fark karşımıza direk olarak liste şeklinde gelmesidir. Kodlama olarakda DropDownList de geçerli olan tüm kodlar ListBox içinde geçerlidir.

### Örnek uygulama:

1) Yeni bir Web sayfası açalım. Web sayfamıza TextBox ve ListBox ekleyelim.



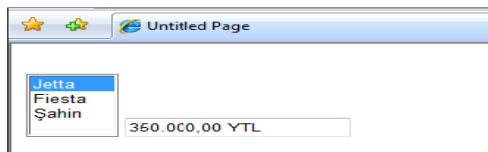
2) ListBox'ımıza Properties penceresinden Items seçeneğini kullanarak değerlerimizi ekleyelim.



3) Değerlerimizi girdikten sonra ListBox'ımızı çift tıklayarak kod alanımıza geçelim. Aşağıdaki kodumuzu yazarak çalıştıralım. Tabi bu arada Listbox'ımızı seçerek AutoPostBack seçeneğini True yapmayı da 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 ayraç olarak ayarlayacaktır.



### ❖ Checkbox

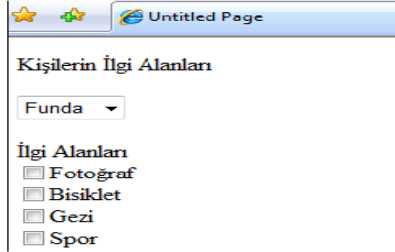
Genel kullanımı bir formda birden fazla seçim yapmak için kullanılır. Aşağıdaki örneğimizde seçtiğimiz dersleri bir textboxa ekleyelim.



Özellik Adı	Tanımı
Checked	İşaretli Olup Olmaması
Text	Ekranda Görünecek Yazısı
Enabled	Aktif Olup Olmaması
BorderWidth	Çerçeve Kalınlığı

### Örnek uygulama:

1. Web Sayfamıza 4 CheckBox ve bir adet DropDownList ekleyelim. CheckBox' larımızın properties penceresinden Text özelliğini kullanarak değerleriniz değiştirilelim.

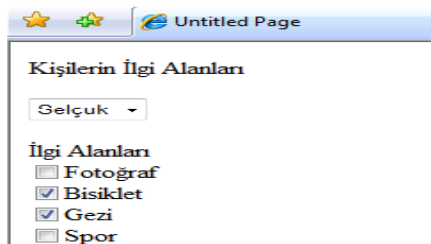


2. DropDownList'e items özelliğinden üç isim girelim (Örn: Funda,Selçuk,Ahmet) ve AutoPostBack özelliğini True yapalım.

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.

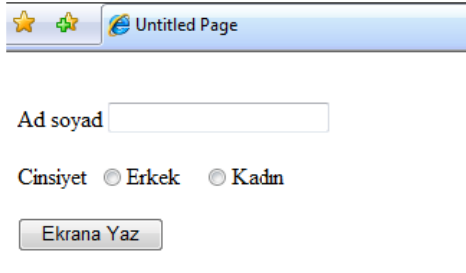


### ❖ Radiobutton

CheckBox'a benzer, tek farklılık bir grup ismi verilen radiobuttonlardan sadece bir tanesini seçme durumumuz olacaktır

### Örnek uygulama:

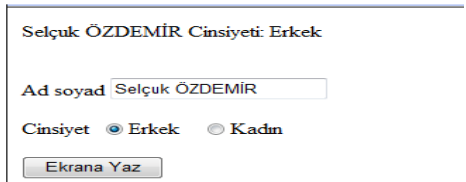
- 1) İlk örneğimizde açmış olduğumuz web sayfasına 2 RadioButton, birer adette TextBox ve Button Ekleyelim. RadioButtonların text özelliğinden Erkek – Kadın olarak değiştirelim.
- 2) RadioButtonlarımızı tıklayarak Properties penceresinden GroupName kısımlarına cinsiyet yazalım. Böylece bu gruba adında 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 TextBox' daki ismi alarak cinsiyetini ekrana yazacaktır.



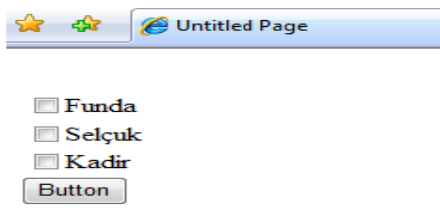
### ❖ Checkboxlist

CheckBox'a benzer bir yapıya sahiptir. Listbox ile CheckBox' ın birleşmesi de denilebilir

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

### Örnek uygulama:

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

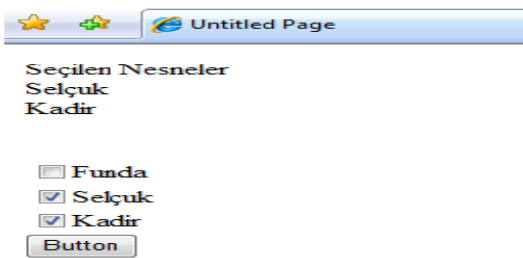


- 3) Aşağıdaki kodu ekleyerek çalıştıralım. CheckBoxlist'ten isimleri seçtikten sonra butona bastığımızda seçtiğimiz isimleri ekranda görebiliriz.

```
protected void Button1_Click1(object sender, EventArgs e)
{
    String s = "Seçilen Nesneler<br>";

    for (int i = 0; i < CheckBoxList1.Items.Count; i++)
    {
        if (CheckBoxList1.Items[i].Selected)
        {
            s = s + CheckBoxList1.Items[i].Text;
            s = s + "<br>";
        }
    }

    Label1.Text = s;
}
}
```

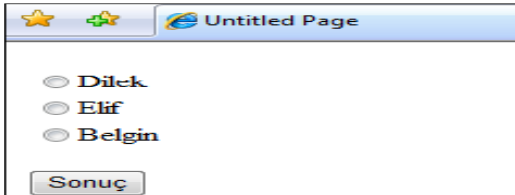


#### ❖ Radiobuttonlist

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

#### Örnek uygulama:

- 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



- 3) Kod alanına geçtikten sonra aşağıdaki kodu yazdıktan sonra çalıştırınız. Radio butonlarından birini seçtikten sonra butona basınız

```
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");
    }
}
}
```

### Belgin Muhasebe

- Dilek  
 Elif  
 Belgin

Sonuç

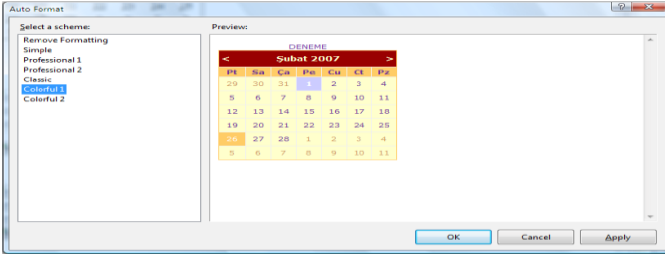
Böylece ListBoxdaki gibi index numarasından seçtiğimiz RadioButton algılanarak, ekrana bilgiler yazılmıştır.

#### ❖ Calendar

Ekrana bir takvim getirerek tarih bilgilerini almamızı sağlar. Ayrıca kullanıcıların tarih formatlarında hata yapmalarını engellemek amaçlı kullanılır Calendar nesneminin özellikleri aşağıdaki listede yer almaktadır.

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

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

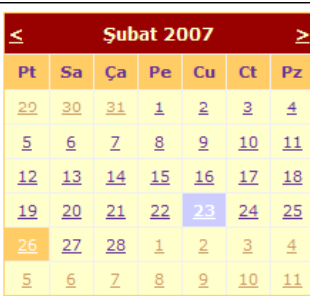


#### Örnek uygulama:

- 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



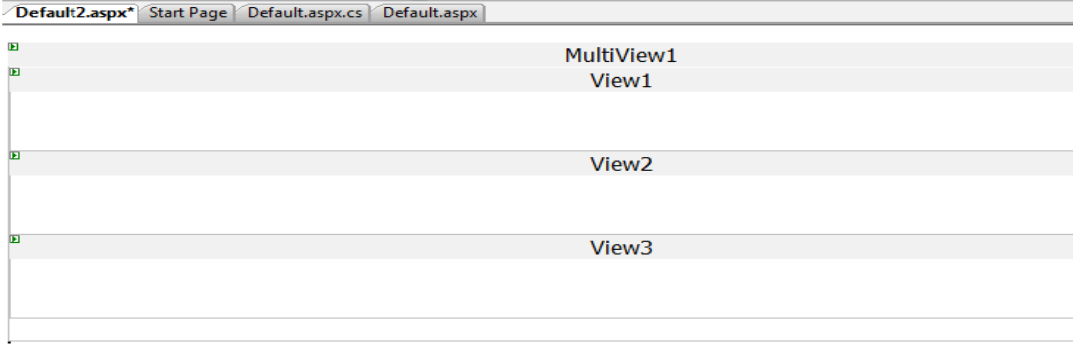
23.02.2007

#### ❖ Multiview - View

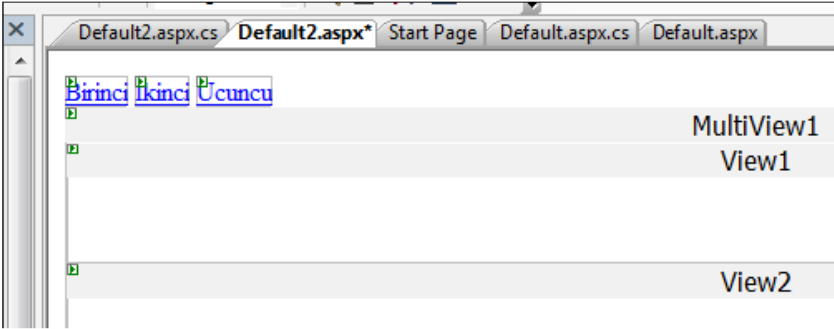
Kullanımı tab menüye benzeyen, oldukça kullanışlı olan Multiview ve View nesneleri beğeneceğinizi düşünüyorum Multiview bilmemiz gerek bir ActiveViewIndex özelliği, hangi View nesnesinin görünmesini istiyorsak onun index değerini yazmamız gerekmektedir. Index numaraları "0" dan başlanmaktadır. Eğer -1 değerini belirlersek ekranda hiçbir View görünmeyecektir.

### Örnek uygulama:

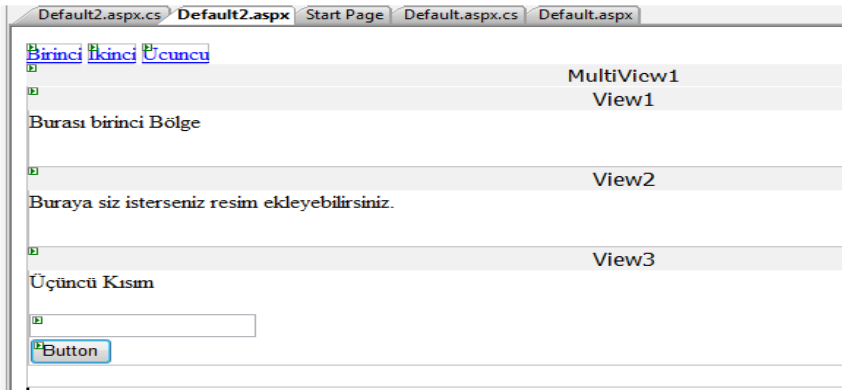
- 1) Bir web sayfası açalım ve bir tane Multiview ekleyelim.
- 2) Daha sonra Multiview'in içerisine 3 adet alt alta View ekleyelim



- 3) Daha sonra MultiView'in üstüne 3 adet Link Button ekleyerek değerlerini değiştirelim



- 4) View1, View2 ve View3 ün içerisine istediğiniz nesneleri ekleyebilir veya yazabilirsiniz



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

```
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;
}
```

## Birinci İkinci Üçüncü Üçüncü Kısım

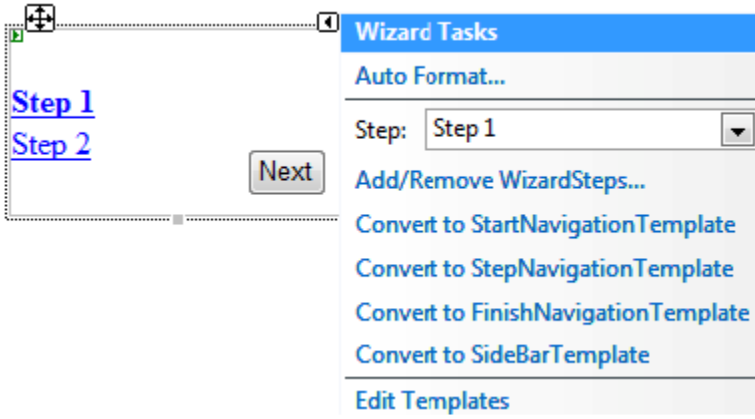
  

### ❖ Wizard

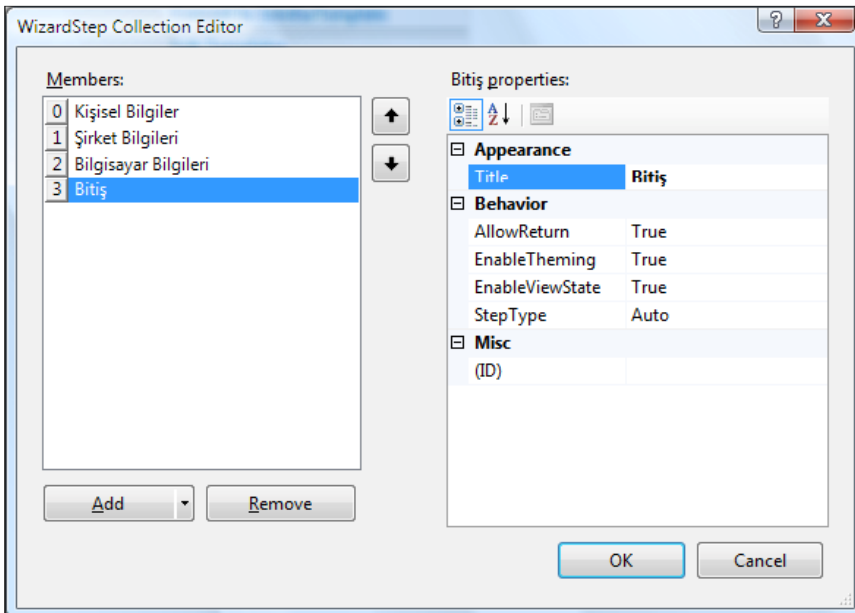
Web sayfalarında kayıt olurken; kullanıcıya sorulan soruları farklı formlar da gösteririz. Bunu yapmak için ya her formu bir sayfaya atarız, yada paneller kullanarak visible özelliklerini true-false yaparak paneli ayarladık. Bunun yerine Asp.Net 2.0 da Wizard kontrolü bu açıdan oldukça işimize yaracaktır.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

### Örnek uygulama:

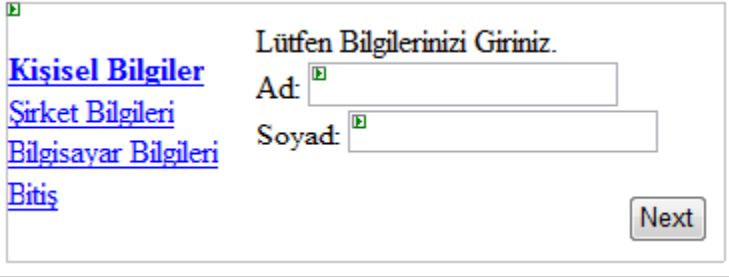
1) Web sayfamıza Wizard kontrolü ekleyelim.



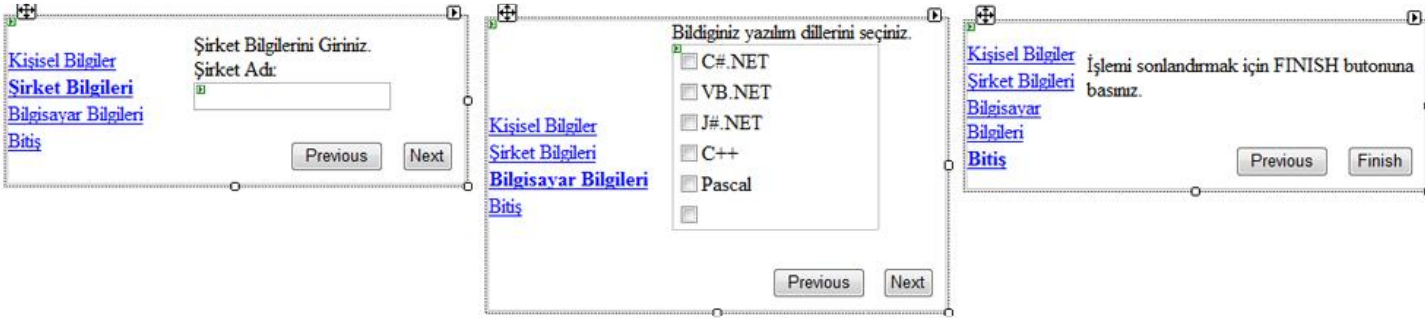
2) Karşımıza gelen pencereden Add/Remove WizardSteps seçeneğini seçiniz. Karşımıza bir pencere gelecektir. Bilgileri şekildeki gibi değiştirip OK butonuna basınız.



3) Daha sonra wizard ın Kişisel Bilgileri yazısını tıklayarak. Aşağıdaki şekil gibi bir görüntü oluşturunuz.



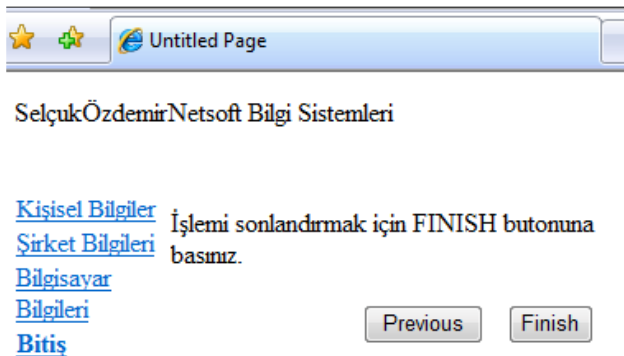
4) Daha sonraki adımları da tek tek yazılarını tıklayarak gerekli biçimlendirmeyi yapınız.



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 bilgilerindeki textbox' lar için kodu yazdım
}
```

5) Kodumuzu çalıştırdıktan sonra textboxlarımıza değer girerek sonucu görebiliriz.



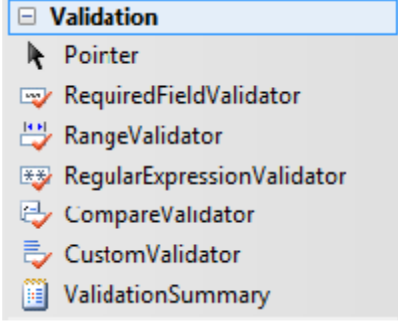
Böylelikle bir kaç işlemimizi tek bir alanda yaparak işlemlerimizi daha hızlı gerçekleştirebiliriz. Finish butonunu veritabanı işlemine bağlayarak Wizard kontrolü işinize yarayacaktır.

## SAYFA YÖNLENDİRME KOMUTLARI

**Response.Redirect("hesap.aspx?tc=3333456567");** sayfanın hesap sayfasına gideceğini ve adres çubuğundan ise diğer sayfaya tc numarasın taşıdığını tanımlamaktadır. 1 değer taşır.  
**Response.Redirect("hesap.aspx?y1="+n1+"&y2="+n2+"&y3="+n3);** burada ise diğer sayfaya 3 değer taşımaktadır. Adres çubuğuna değerler değişkenler yardımı ile alınmaktadır.  
**Server.Transfer("hesap.aspx?tc=3333456567")** redirect komutu ile aynı işlemi yapmaktadır. Tek farkı adres çubuğunda bilgiler gizli gönderilir.  
**Request.QueryString["y1"]** adres çubuğunda taşınan değeri almayı sağlar.

## VALIDATION KONTROLLER

Web sayfalarında en önemli bir konu halini alan bilgi girişi ve bunların kontrollerini yapan kontrollerdir



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.

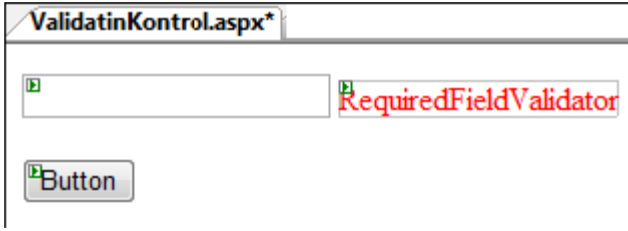
Validation Kontrol Adı	Açıklama
Required Field Validator	Kontrol içerisine bir değer girilip girilmediğini kontrol eder. Atamış olduğumuz kontrol boş geçemez.
Range Validator	Belirlediğimiz iki değer arasında kontrol eder. Örneğin yaş aralığı için kullanılabilir.
Reguler Expression Validator	Karakter 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.

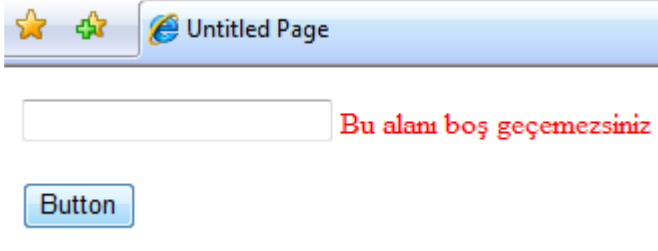
### Örnek uygulama:

1) İlk uygulamamızda RequiredFieldValidator kontrolünü kullanacağız. Bunun için web sayfamıza birer adet textbox, button ve RequiredFieldValidator kontrollerini ekleyelim





- 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. Fakat TextBox’ımıza bir şeyler yazıp butona bastığımızda hata mesajı kaybolacaktır.

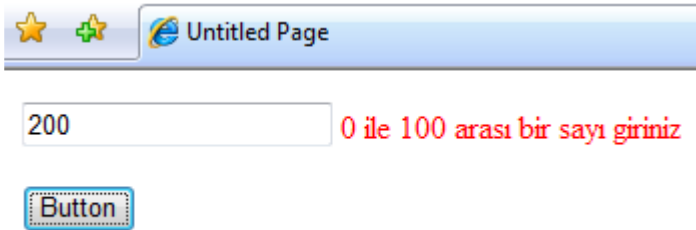


#### Örnek uygulama:

- 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

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

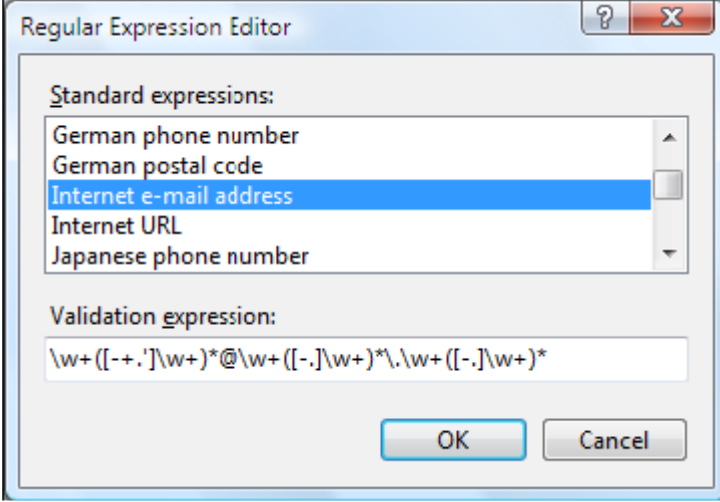
- 4) Değerleri girdikten sonra CompareToValidate seçeneğinden Textbox objenizi seçiniz. Seçtikten sonra kodumuzu çalıştıralım ve textbox a “200” gibi büyük bir rakam girelim. RangeValidation hata verecektir. Eğer siz aralık olarak iki tarih aralığı belirlemek istiyorsanız Range Validation’nın Type özelliğini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



#### Örnek uygulama:

- 1) Bu uygulamamızda ise RegularExpressionValidator kontrolünü anlatacağım. Bunu için Web sayfamıza birer adet textbox ve buton nesnelere yerleştiriniz. TextBox nesneminizin yanınada RegularExpressionValidator kontrolünden koyalım.
- 2) RegularExpressionValidator kontrolüne ait properties penceresinden ErrorMessage kısmına “Yanlış E-Mail” yazısını, ControlToValidate seçeneğinden de textbox1’i seçiniz.

3) Properties penceresinde bulunan ValidationExpression seçeneğinin tıkladığımızda 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



4) Burada textbox'a bilgi girişi yapıldığında "@" ve "." işaretlerini kontrol edecektir



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

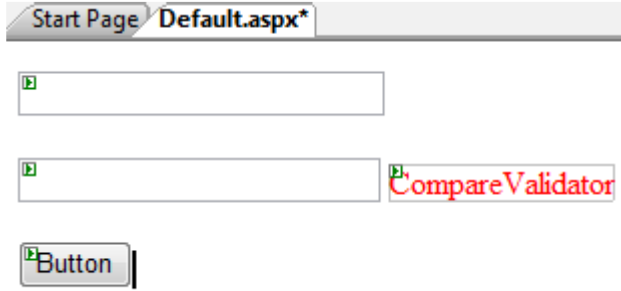
[^]	Belirtilen aralık dışındaki tür değerler girilecek	[^1-3] =1,2 ve3 dışındaki tüm değerler
\s	Boşluk Karakteri	\s = , ,
\S	Boşluk dışındaki tüm karakterler	\S+ = abc,123
\d	Rakamsal değer girilir	\d+ = 1234567890
\D	Rakam olmayan tüm karakterler	\D+ = abcdefgh
\w	Kelimelerde kullanılan tüm karakterler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

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 ekleyelim.

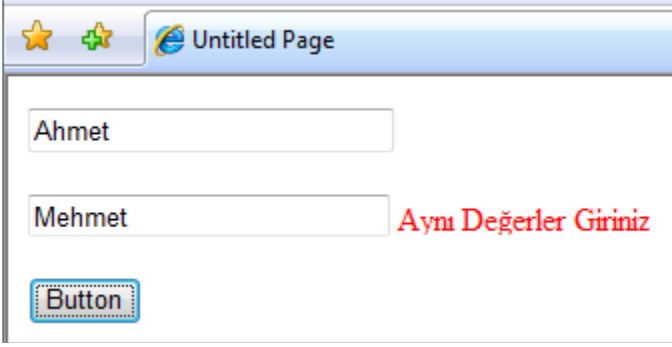
**Örnek uygulama:**



2) CompareValidator kontrolünün properties penceresinde ErrorMessage yazısını “Aynı Değerler Giriniz” olarak değiştiriniz. Daha sonra ControlToCompare seçeneğinde textbox1’i, ControlToValidate seçeneğinde ise textbox2’i seçiniz

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

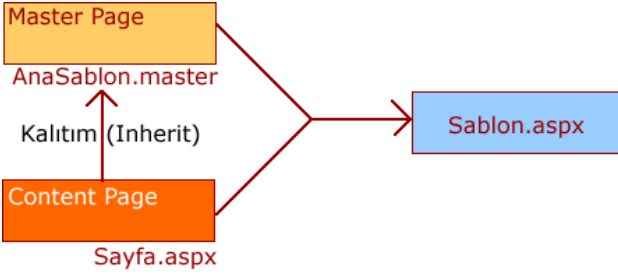
3) Daha sonra kodumuzu çalıştıralım. Textbox1’e ve Textbox2 kontrollerine aynı değerleri yazmamız gerekmektedir. Buradaki seçeneklerden ControlToCompare seçeneğinde TextBox1’i seçtiğimiz için kontrol bu nesne üzerinden olmaktadır.



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

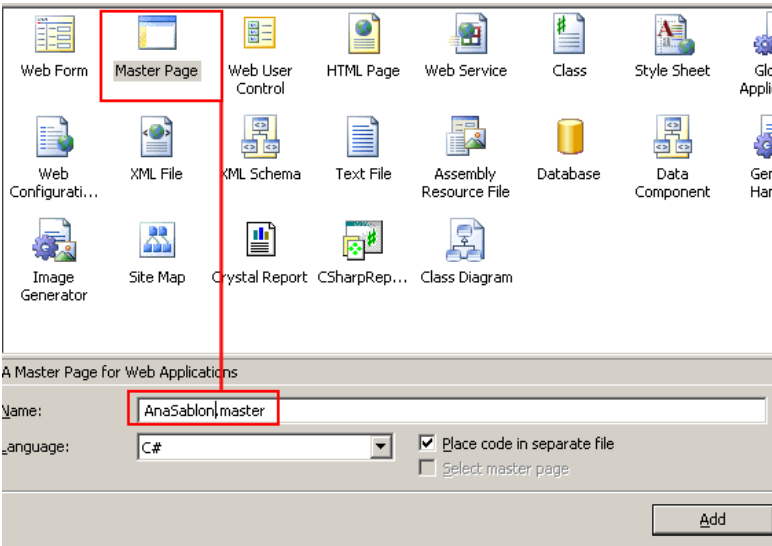
## MASTER PAGE UYGULAMASI

İnternet sitelerini göz önüne aldığımızda, siteye ait sayfaların sıklıkla aynı şablonları kullandığını görürüz. Özellikle, header(Üst), footer(Alt), navigasyon(Menüler) çoğunlukla siteye ait tüm sayfalarda aynı yerlerde kullanılır. Bu, siteye ait sayfaların standart olarak aynı görünümde olmasını sağlamakla kalmaz, değişen içeriğinde ortak bir şablon üzerinde durmasına imkân tanır. Bir Master Page, uygulandığı diğer aspx sayfalarının nasıl görünmesi gerektiğine karar veren bir şablona gibidir. Ancak, sağladığı ContentPlaceHolder bileşeni sayesinde, Master Page' leri uygulayan diğer aspx sayfalarının, istenilen içeriğe sahip olmasını da sağlamaktadır. En güzel yanı ise, normal bir aspx sayfası gibi tasarlanabilmesi, yani html, image, server control gibi üyeleri içerebilmesidir.

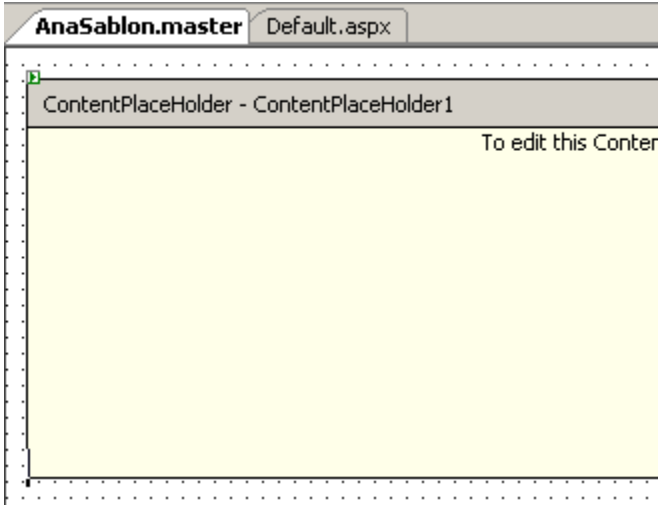


*Master Page ve aspx sayfalarının ortak çalışma mimarisi.*

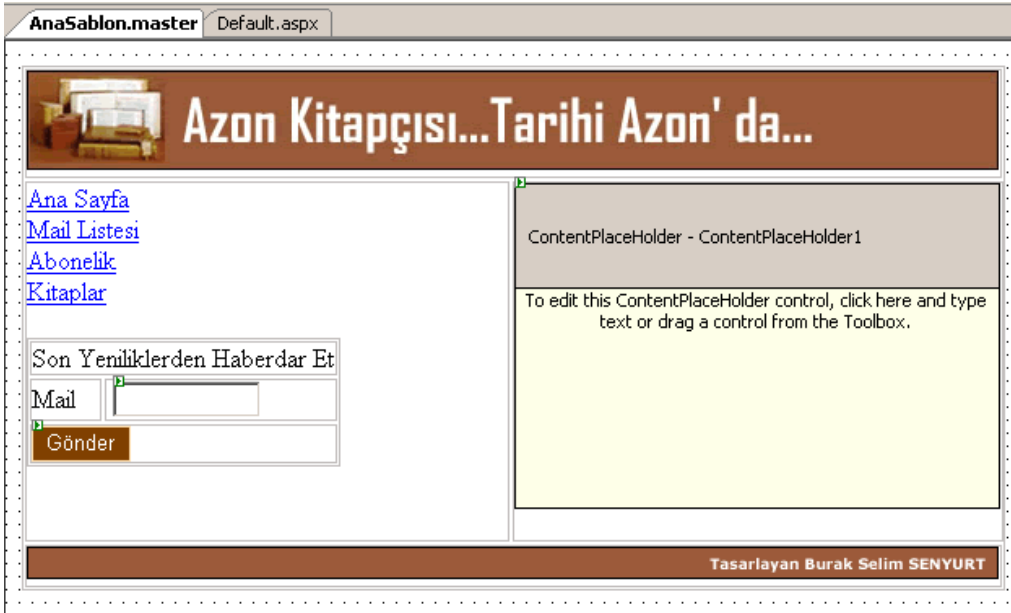
İlk olarak, Visual Studio.Net 2005' te bir web sitesi açalım. Sitemize Master Page eklemek için tek yapmamız gereken, Solution' ımıza sağ tıklamak ve Add New Item' den gelen pencerede, Master Page tipini seçmektir. Master Page' ler master uzantılı dosyalardır.



Bu işlemin ardından Master Page' in standart olarak aşağıdaki gibi oluşturulduğunu görürüz.

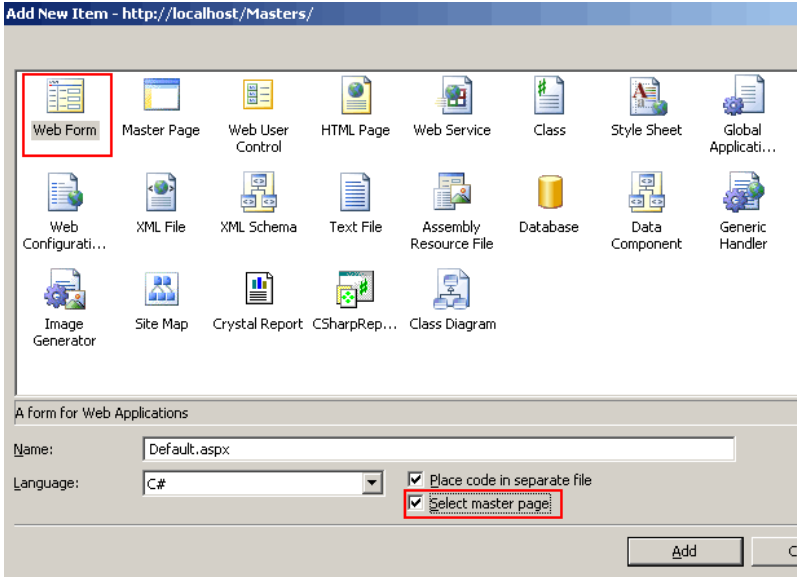


İşte burada ContentPlaceHolder1 bileşenimiz, bu Master Page' i uygulayacak olan sayfaların serbestçe erişebilecekleri ve içerik oluşturabilecekleri alanları tanımlamaktadır. Elbette ki bir Master Page' in bu şekilde olması beklenemez. Bu nedenle Master Page' imizi aşağıdaki ekran görüntüsünde olduğu gibi tasarlayabiliriz. Dikkat edecek olursanız, Master Page' lerde, normal aspx sayfaları gibi tasarlanabilirler. Bir başka deyişle, Html kodları, aspx bileşenleri vb. içerebilirler.

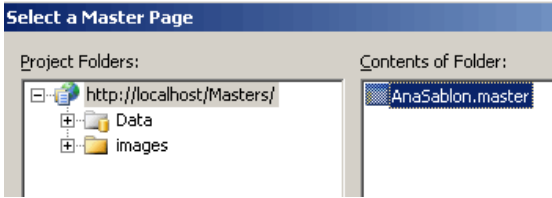


Burada standart olarak bir web sayfasının tasarlanmasından farklı bir işlem yapılmamıştır. En önemli nokta Master Page' i uygulayacak sayfaların içeriklerini yazabilecekleri ContentPlaceHolder bileşeninin kullanılmasıdır. Dilersek bir Master Page içinde, birden fazla ContentPlaceHolder bileşenine de yer verebiliriz. Master Page' in aspx kodlarına baktığımızda normal aspx sayfalarına göre en önemli değişik page direktifi yerine master direktifinin kullanılmasıdır. Master direktifi sayfanın bir Master Page olduğunu belirtmektedir.

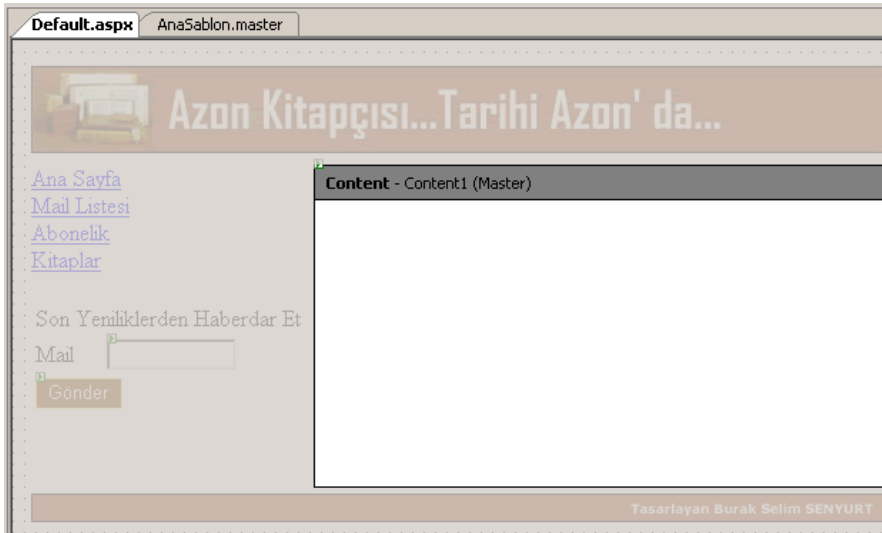
Yeni bir aspx sayfasına oluşturduğumuz Master Page' i nasıl uygulayacağımıza bir bakalım. Öncelikle, Add New Item iletişim kutusunu açalım ve dosya tipi olarak Web Form' u seçelim. Ardından, sayfamıza uygulamak istediğimiz Master Page' i seçebilmek amacıyla, Select Master Page kutucuğunu işaretleyelim.



Bu durumda Add butonuna bastığımızda, sayfamıza uygulamak istediğimiz Master Page' i seçeceğimiz iletişim kutusu ekrana gelecektir.



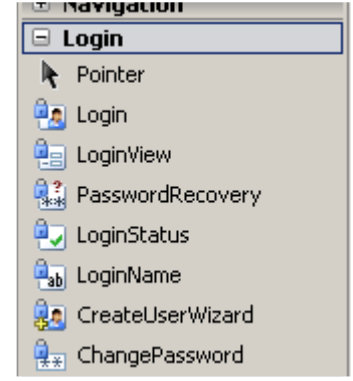
Bu adımı da tamamladığımızda, default.aspx sayfamız aşağıdaki gibi oluşturulacaktır. Dikkat edecek olursanız, sadece Master Page' deki ContentPlaceHolder bileşeninin bulunduğu alan düzenlenebilir yapıdadır. Diğer kısımlar için düzenleme ve değiştirme gibi işlemleri gerçekleştirme imkanımız yoktur. Bu sayede web formunun, Master Page' in izin verdiği görünümde olması ve kendisine ayrılan alanda istediği içeriği oluşturmasına izni verilmiş olunur.



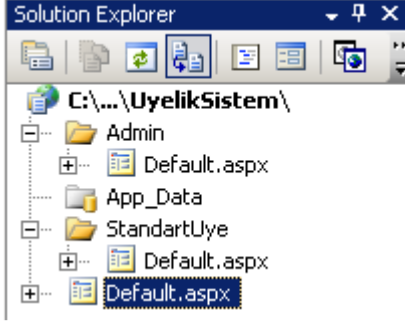
# ÜYELİK SİSTEMİ

## LOGIN KONTROLLER

Hemen her sitede gördüğümüz üyelik alanlarını yazmak için 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. Login nesnelerinin çalışması için veritabanının oluşması ve yetkilerin atanması gibi bazı güvenlik düzeylerinin yapılması gerekmektedir.

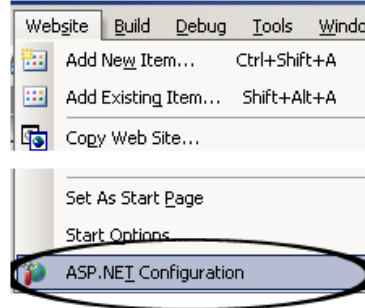


## UYGULAMA 1

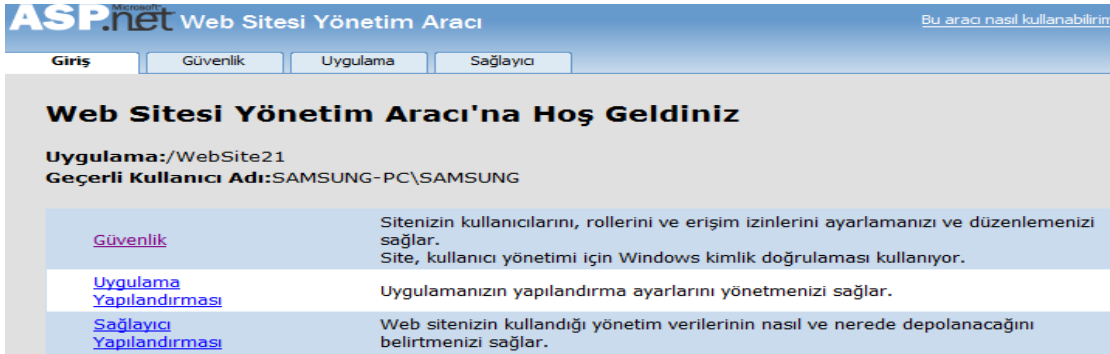


1) Yeni bir web projesi açalım. Solution Explorerda proje isminin üstünü sağ tıklayarak New Folder seçeneğini seçelim ve StandartUye, Admin adından iki tane klasör oluşturalım.

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



3) Karşımıza Asp.Net Web Site Administration Tool ( Web sitesi yönetim aracı) web sayfası açılacaktır. Gelen pencereden Security (Güvenlik) linkini tıklayınız.



- 4) Security (Güvenlik) bölümünü tıkladıktan sonra güvenlik tabında adım adım ayar yapacağımızdan **Use the security Setup Wizard to configure security step by step** ([Güvenliği adım adım yapılandırmak için güvenlik Kurulum Sihirbazı'nı kullanın.](#)) linkini tıklayınız.
- 5) Karşımıza gelen Güvenlik Kurulum Sihirbazı'nın (Security Setup Wizard) ilk penceresinde İleri butonuna basarak diğer adıma geçiniz.
- 6) Karşımıza iki seçenek gelir: İnternette (From the Internet) ve Yerel ağdan (From a local area network). Biz uygulamamızı internete taşıyacağımızdan dolayı İnternette (From the Internet) seçeneğini seçerek İleri butonuna basalım.

**Erişim Yöntemini Seçin:**

Sitenizin güvenliğini oluşturmada ilk adım kullanıcıları tanımlamaktır (kimlik doğrulama). Kullanıcıları tanımlama yöntemi, kullanıcıların sitenize nasıl eriştiğine göre değişir.

Kullanıcıların sitenize erişim biçimini belirtmek için aşağıdaki yöntemlerden birini seçin ve İleri'yi tıklayın.

**İnternet'ten**  
Uygulamanız, İnternet'teki herkesin erişebileceği genel bir sitedir. Kullanıcılar, oluşturduğunuz oturum açma sayfasına kullanıcı adlarını ve parolalarını girerek uygulamanıza oturum açabilirler.

**Yerel ağdan**  
Uygulamanız yalnızca özel yerel ağda (intranet) çalışıyor. Kullanıcılar, Windows etki alanları ve kullanıcı adlarıyla tanımlanırlar ve uygulamanıza açıkça oturum açmaları gerekmiyor.

7) Karşımıza Gelişmiş Sağlayıcı Ayarları (Advanced Provider Settings) kısmına geçeceğimizi bildiren bir ileti gelecektir. İleri diyerek bir sonraki aşamaya geçelim.

8) Daha sonra Roller tanımlanması gerektiğini belirten bir seçim gelecektir. Bu Web sitesi için rolleri etkinleştir (Enable roles for this Web site) seçeneğini işaretleyip İleri düğmesine tıklayarak devam edelim.

9) Daha sonra karşımıza Rol isimlerini girmemiz gereken bir pencere gelecektir. Buraya GenelKullanici ve Yonetici adında iki tane Rol tanımlayalım. Rol ismini yazdıktan sonra Rol Ekle butonuna basalım.

Bu Web sitesi için rolleri etkinleştirdiniz.

**Yeni Rol Oluştur**

Yeni Rol Adı:  
Yonetici

**Varolan Roller**

GenelKullanici	<a href="#">Sil</a>
Yonetici	<a href="#">Sil</a>

10) Roller ekledikten sonra İleri düğmesini tıklayarak devam edelim. Daha sonra karşımıza admin kullanıcıları oluşturmak için bir kayıt alanı gelecektir. Bilgileri girdikten sonra Kullanıcı Oluştur butonuna tıklayınız.

Bu sayfaya kullanıcı kimliğini, parolasını ve e-postasını girerek kullanıcı ekleyin. Parola sıfırlama veya unutulmuş bir parolayı isteme işleminde kullanılacak bir soru ile kullanıcının vermesi gereken yanıtı da belirtebilirsiniz.

**Kullanıcı Oluştur**

**Yeni Hesabınız için Kaydolun**

Kullanıcı Adı:

Parola:

Parolayı Onayla:

E-posta:

Güvenlik Sorusu:

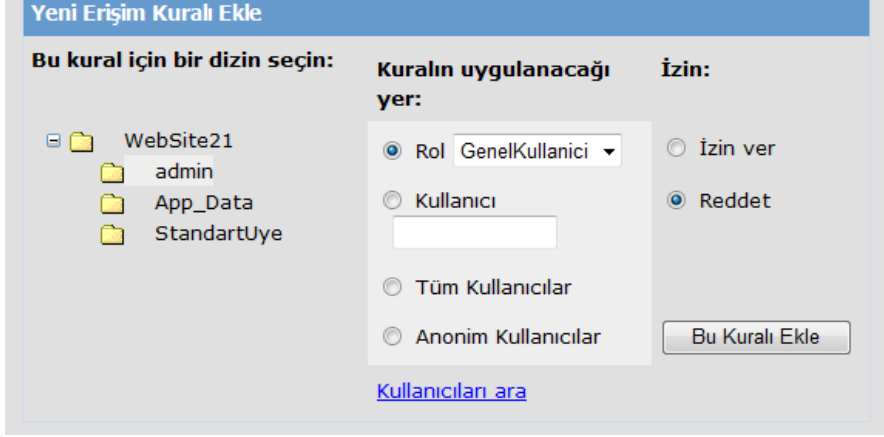
Güvenlik Yanıtı:

Etkin Kullanıcı



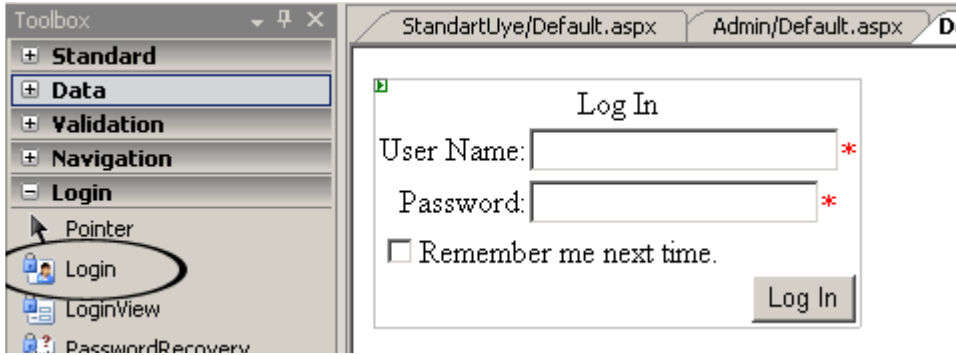
11) Kullanıcı oluşturduktan sonra karşımıza kullanıcının başarılı şekilde oluşturduğunu belirten bir yazı gelecektir. Eğer başka bir kullanıcı oluşturmak istiyorsanız Devam butonuna tıklayınız. İşlemimiz bittiği için biz İleri düğmesine tıklayarak işlemimize devam edelim.

12) Daha sonrasında karşımıza Roller atayacağımız sayfa gelecektir. Burada GenelKullanici'ya üye olan kişilerin daha önceden oluşturduğumuz Admin klasörüne girmemesini sağlayacağız. Bunu için sol taraftan Admin klasörünü seçiniz. Daha sonra İzin kısmından reddet seçeneğini seçerek Bu Kuralı Ekle butonuna tıklayınız. Fakat StandartUye klasöründe İzin ver diyerek giriş izni vermemiz gerekecektir.



13) Rol tanımlamalarımızı yaptıktan sonra İleri düğmesine basarak işlemimize devam edelim. Bu işlemden sonra yetkilendirme işlemlerimiz bitmiştir. Son düğmesine tıklayarak işlemimizi bitirelim.

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



15) Hiçbir kod yazmadan sayfamızı çalıştıralım. Biraz önce kullanıcı oluşturmada kullandığımız bilgilerden kullanıcı adı ve şifresini girip Log In butonuna basalım. Eğer kullanıcı bilgilerinin 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 butonunun üstündeki yazı
LoginButtonType	Giriş butonunun tipini belirler
Orientation	Login ekranının 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	Üye olmak isteyenlerin yazıyı tıkladıklarında açılacak sayfa
PasswordRecoveryText	Şifremi unuttum yazısı
PasswordRecoveryUrl	Şifremi unuttum yazısını tıklayınca açılacak sayfa

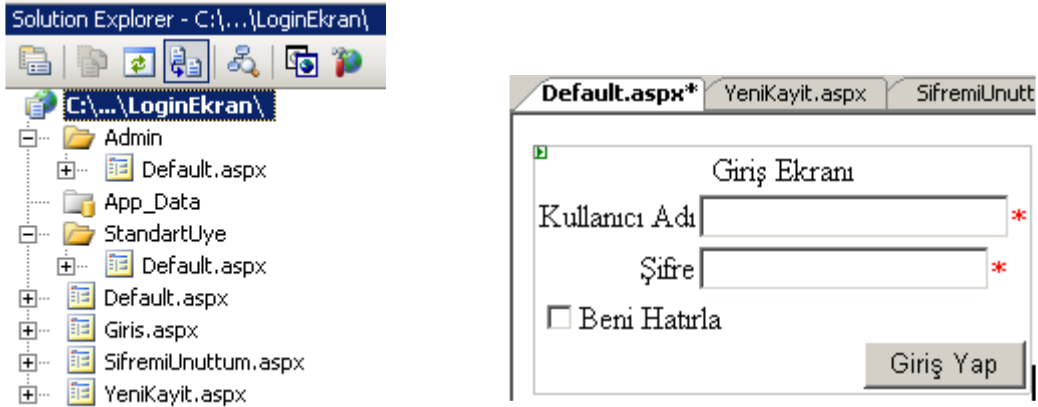
ASP.NET Configuration kısmında kullanıcı, rol veya kuralları oluştururken Sql Server hataları alma durumunuz olabilir. Bunun nedenlerinden bazıları, Framework de sorun olması veya açık olan Configuration sayfasının fazla süre açık kalmasından dolayı zaman aşımına uğramış olabilir. Bundan farklı olarak da Asp.Net user yetki hatası verebilir. Bunu da

Command satırına gelerek (cmd.exe) C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\ aspnet\_regiis -i

Komutunu çalıştırmak gerekecektir.

## UYGULAMA 2

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



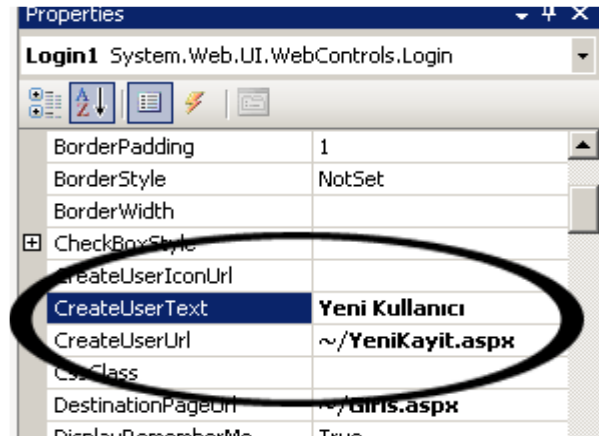
2) Root'daki Default.aspx dosyamızda Login nesnesini yerleştirelim.

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

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. Aksi takdirde hata mesajı verecektir.

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.

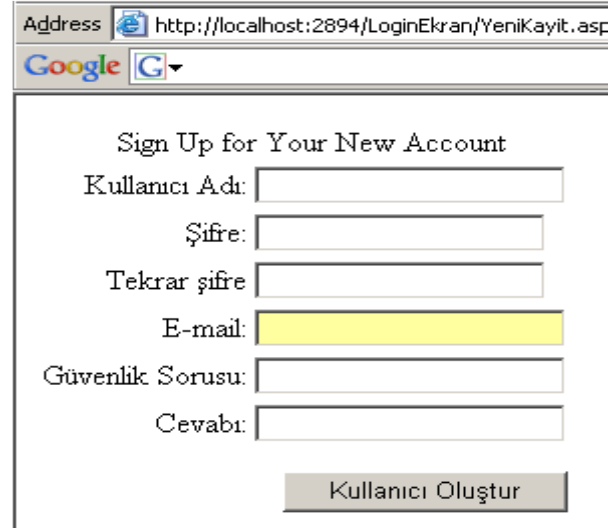
6) Properties penceresinden PasswordRecoveryText özelliğine "Şifremi Unuttum" yazısını, PasswordRecoveryUrl kısmına ise SifremiUnuttum.aspx dosyamızı atayalım.



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

8) CreateUserWizard seçeneğini sayfamıza eklediğimizde kayıt için gerekli bir form gelecektir. Bu formu properties penceresinden değerlerimizi Türkçeleştirebiliriz.

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



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

Sign Up for Your New Account

Kullanıcı Adı:

Şifre:

Tekrar şifre

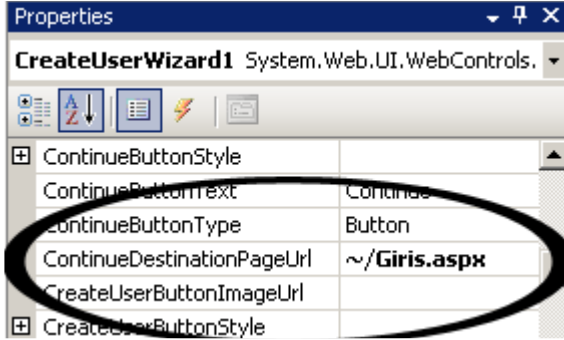
E-mail:

Güvenlik Sorusu:

Cevabı:

Kullanıcı Oluştur

10) Continue butonuna bastığımızda göndermemiz gereken bir sayfa var ise properties penceresinden ContinueDestinationPageUrl özelliğine dosyanızı atamanız yeterlidir.

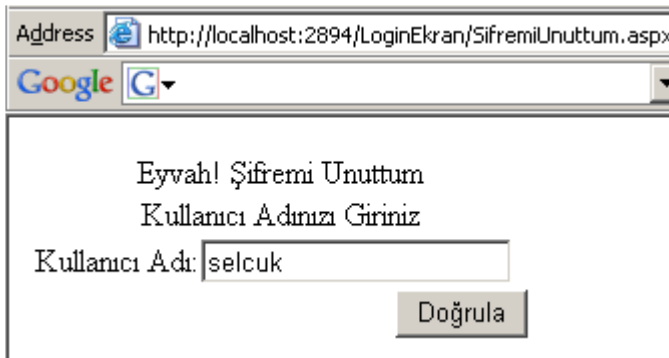


11) Böylece artık default.aspx sayfamızdan yeni kayıt olduğumuz kullanıcılar giriş yapabilirler.

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



13) PasswordRecovery nesnemizi properties penceresinden yine üstündeki yazılarımızı değiştirebiliriz.



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

Eyvah! Şifremi Unuttum

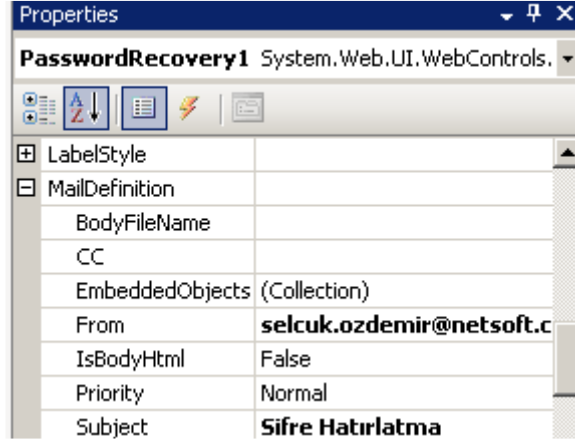
Kullanıcı Adınızı Giriniz

Kullanıcı Adı:

Doğrula

14) Kullanıcı adımızı girdikten sonra Doğrula butonumuzu tıkladığımızda Güvenlik Soru gelecektir. Bu soruyu da doğru cevaplamanız gerekmektedir.

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.



16) SMTP hatasını aşmak için bu kadarı yeterli değil. Tabii ki de sisteminizde SMTPServer yapısını kurulu olması gerekmektedir. Bu ayarlardan sonra Kişiyeye “Şifre Hatırlatma” başlığında From kısmındaki mail üzerinden kişiyeye şifresi gönderilecektir.

## VERİTABANI

Bir veritabanı isminden de anlaşılacağı gibi bilgilerin depolandığı hiyerarşik bir yapıdır. Başka bir tanımla dijital 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 benzese de aslında çok fazla ilgisi yoktur. ADO.NET (Activex Data Object) tamamıyla yeniden geliştirilen yapısı ile verilere ulaşır. ADO.NET verilere erişirken ADO gibi sadece OLEDB Provider değil, onun dışında SQL **Provider(sağlayıcı)** 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 adresinden OPD.NET’i indirmeniz gerekmektedir.

**Data Provider (Veri Sağlayıcı)** bileşeni veri kaynağına özgüdür. .NET Framework dört adet

Data Provider içerir bunları sırası ile sayacak olursak;

- 1) Tüm OLE DB veri kaynakları ile iletişim kurabilen geniş kapsamlı bir sağlayıcı.
- 2) Microsoft SQL Server 7.0 ve sonraki sürümleri için düzenlenmiş bir SQL Server sağlayıcısı.
- 3) ODBC Veri kaynakları için düzenlenmiş bir ODBC sağlayıcısı.
- 4) Oracle Veri Tabanları için düzenlenmiş Bir Oracle Veri sağlayıcısı.

```
Örneğin;  
using System;  
using System.Collection.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Text;  
using System.Windows.Forms;  
using System.Data.SqlClient; //Sql Server Veri Sağlayıcısı Kullanacağınızı Belirtirsiniz.  
using System.Data.OleDb; //OleDbVeri Sağlayıcısı Kullanacağınızı Belirtirsiniz.
```

\*\*\*ADO.Net mimarisinde SQLSERVER kullanacağımız için mutlaka "using System.Data.SqlClient" tanımını yapmayı unutmayınız.\*\*\*

## ***ADO.NET Nesneleri:***

### **1) Connection**

ADO.NET'in veri kaynağıyla canlı ilişkide olan tarafını. NET Veri Sağlayıcısı oluşturuyor. Bu sağlayıcı, veri kaynağına gönderilecek her türlü komut ve ondan dönecek her türlü yanıt için bir köprü vazifesi görüyor. Tüm bu operasyonlar için, bu sağlayıcının aktif bir bağlantısı olması gerekiyor. İşte bu bağlantı, **Connection** nesnesiyle temsil ediliyor. Connection nesnesi, ADO.NET'in en temel nesnesidir.

### **2) SqlConnection**

Microsoft SQL Server veritabanlarına erişmek için **System.Data.SqlClient** aduzayı altındaki **SqlConnection** nesnesini kullanılır. Bu bağlantı nesnesi (daha doğrusu nesneyi sunan sağlayıcı), sadece SQL Server için tasarlandığı için başka bir veritabanı erişiminde kullanılmaz. SQL Server'in daha alt sürümlerine erişim içinse eskiden olduğu gibi yine OLE DB kullanılacak. Böyle bir zorunluluk olmamasına rağmen ayrı bir sağlayıcı kullanılır. Aslında OLE DB hala bir seçenek, ancak veritabanı erişimi gibi masraflı bir işte daha yüksek performans arzuluyorsak, tercihimiz SQL Server.NET veri sağlayıcısı olmalıdır. Çünkü bu sağlayıcı, SQL Server ile **TDS (Tabular Data Stream)** paketleri seviyesinde, yani SQL Server'in **anladığı dilde** haberleşiyor. Böylece OLE DB gibi bir tercümanı direk saf dışı bırakmış oluyorsunuz.

### **3) DataSet**

DataSet nesnesi herhangi bir veri kaynağı ile ilişkilendirilmiş veri kümesini temsil eder. Bu veri kaynağı bir veritabanı olacağı gibi XML formatlı bir kaynakta olabilir. Nitekim DataSet nesnesinde tutulan veriler hafızada XML formatında tutulmaktadır. DataSet içindeki veriler aynen veri tabanında olduğu gibi tablolar şeklinde saklanır. Yani bir DataSet nesnesi bir yada birden fazla tablo barındırabilir. DataSet'in yapısındaki tablolar **DataTable** isimli sınıfla temsil edilir. Bu da DataSet'in DataTable türünden nesnelere sahip olduğunu gösterir. Bu koleksiyonun adı **Tables** dır. Tables koleksiyonun Add() ve Remove() metotları kullanılarak DataSet'e yeni tablo eklenip çıkarılabilir. Tables koleksiyonundaki tablolara indeksleyici yardımıyla erişilebilir. İndeksleyicinin parametresi tablonun ismi olabileceği gibi tablonun DataSet'teki sıra numarası da olabilir.

### **4) DataAdapter**

DataAdapter nesnesi DataSet nesnesi ile veritabanı arasında tercümanlık yapar diyebiliriz. Veritabanından aldığı bilgileri DataSet içerisine doldurur ve DataSet den aldığı bilgileri veritabanına geri gönderir.

### **5) DataReader**

DataReader .NET platformunda karşımıza yeni çıkan ve herhangi bir kaynaktaki veriye FORWARD-ONLY ve READ-ONLY olarak ulaşan bir bileşendir. DataSet 'in belleği fazlaca kullanması bazı uygulamalarda sorun olabilir. Eğer veri tabanındaki kayıtlara salt okunur olarak erişmek yeterli oluyor veya veriler üzerinde bir değişiklik yapılması gerekmiyorsa Dataset yerine DataReader nesnesi ile çalışarak bellek daha verimli kullanılabilir.

## 6) Command

Command nesnesi, veri kaynaklarına karşı icra edilecek komutları temsil eden **ADO.NET** nesnesidir. **ADO**'daki Command nesnesi ile benzer özelliklere sahiptir. Command nesnesi de bir **Provider** nesnesidir ve temsil ettiği komutları icra ettirebilmesi için kurulmuş (açık) bir bağlantıya ( Connection nesnesine) ihtiyacı vardır.

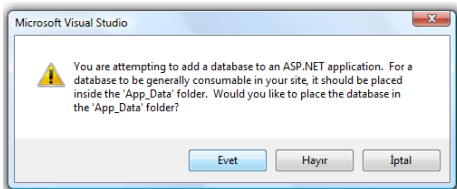
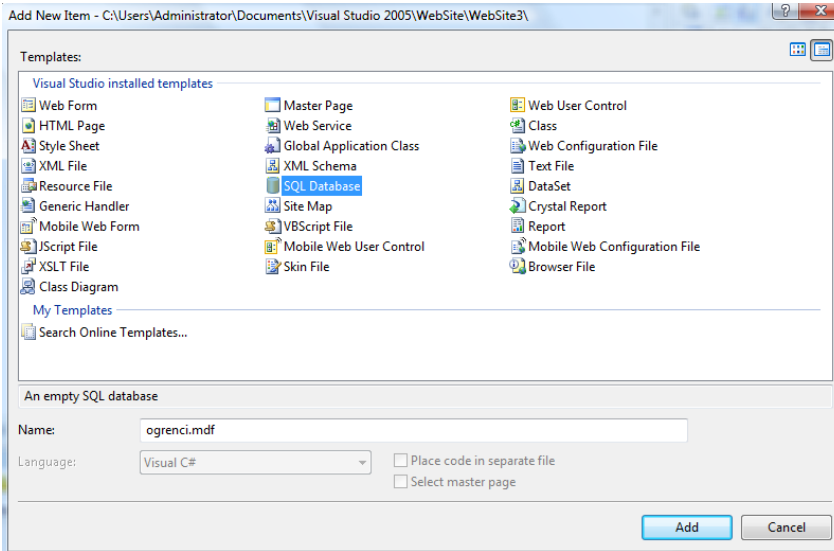
## Veritabanı Oluşturma

Asp.net' de veri tabanı ve veriler gireceğimiz Tablolar oluşturmayı uygulamalı olarak görelim;

Örneğimizde **"Ogrenci"** adında bir database (veritabanı), bunun içerisinde **"tblbilgiler"** adında bir tablomuz yer almaktadır. Veritabanımızı yaptıktan sonra içerisine bilgiler giriniz.

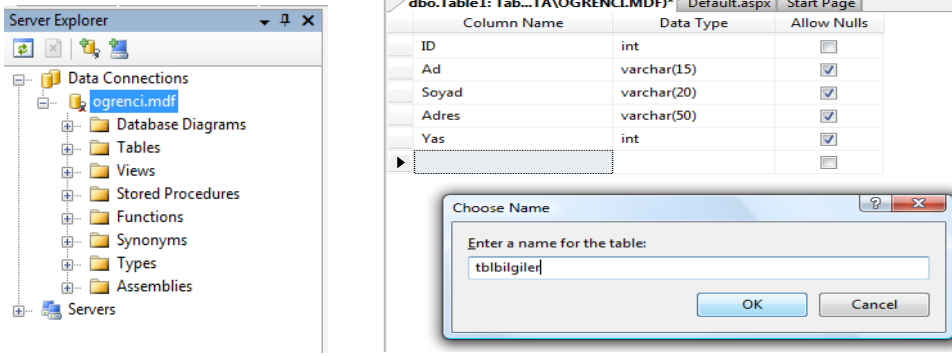
### Uygulama:

- 1) File→ New→Web Site seçeneğini seçerek yeni bir web sitesi oluşturalım,
- 2) Website→Add New Item menüsünden **"SQL Database"** seçilerek name bölümüne veritabanımızın adını giriniz **"ogrenci"**.



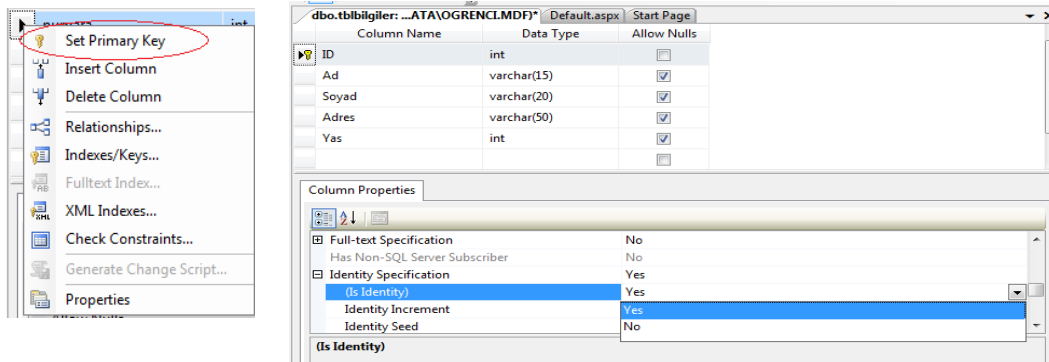
Oluşturduğumuz veritabanını web projesinin içerisindeki App\_Data klasörüne ekliyoruz.

- 3) "Server Explorer" penceresinde veritabanını seçiniz ve **Data** menüsünden **Add New→Table** seçilerek veriler gireceğimiz tablo oluşturulur. Bilgiler girilerek kaydedilir.



\*\*\* Tablo üzerinde sağ tuş → açılan menüden "Show Table Data" seçeneği tıklanarak, tabloya bilgiler girilir. Düzenleme yapmak için aynı yoldan "Open Table Definition" seçeneği tıklanır. \*\*\*

#### 4) Sıralı bir işlem yapacaksa "ID" satırında sağ tuş **"Set Primary Key(birincil anahtar)"** seçilir. "Column Properties" panelinden özellikler ayarlanır.



Veritabanımızı ve içerisinde bilgilerimizin olduğu tabloyu oluşturduk...

### **Veritabanına Bağlantı**

#### **SqlDataSource ile Veritabanına Bağlantı:**

**"Data Source=mssql.yukselcinar.net.;Initial Catalog=altinok\_yuksel;Persist Security Info=True;User ID=hakki;Password=0000";**

**="@Data Source=.\SQLEXPRESS;Initial Catalog=pejo;Integrated Security=True"**

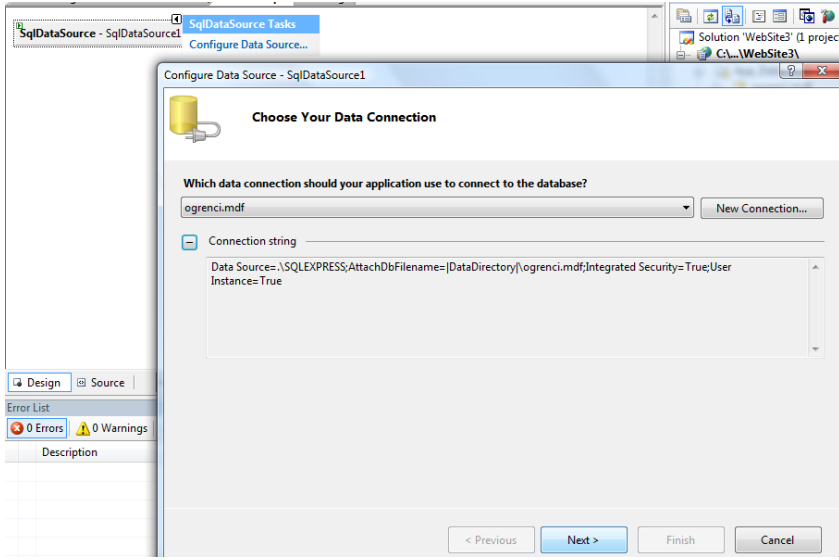
**SqlDataSource:** aslında tüm veritabanlarına bağlanmak için kullanılır. Aslen Sql Server ve Oracle gibi server tabanlı veritabanlarına bağlanmak için kullanılır.

**GridView:** Veriyi düzenli bir şekilde ekranda görüntülemek ve üzerinde işlem yapmak için kullandığımız sunucu kontrolüdür.

Uygulamamızda SqlDataSource nesnesini kullanarak GridView nesnemize datalarımızı getireceğiz.

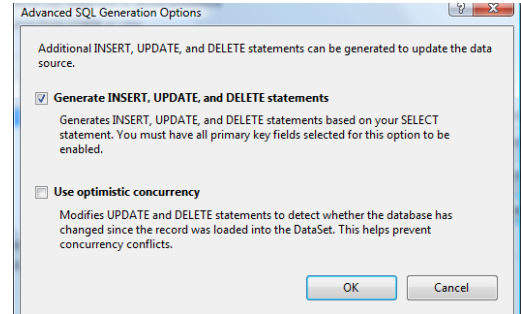
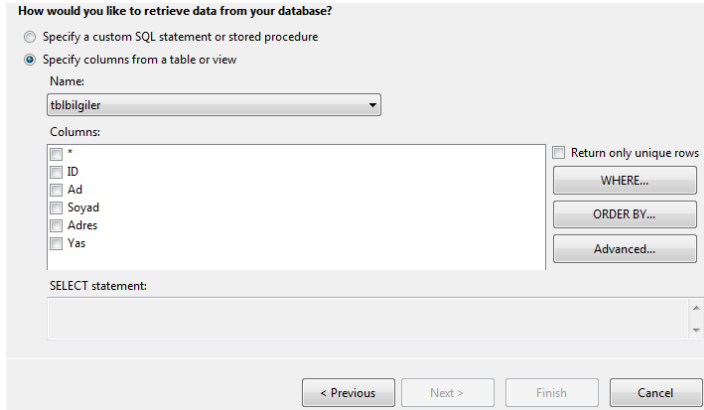
#### **Uygulama:**

- 1) Web sayfamıza SqlDataSource nesnesi koyunuz. Daha sonra Configure Data Source kısmında tıklayarak Configure Data Source penceresini açınız oluşturduğumuz veritabanı adını seçiniz,

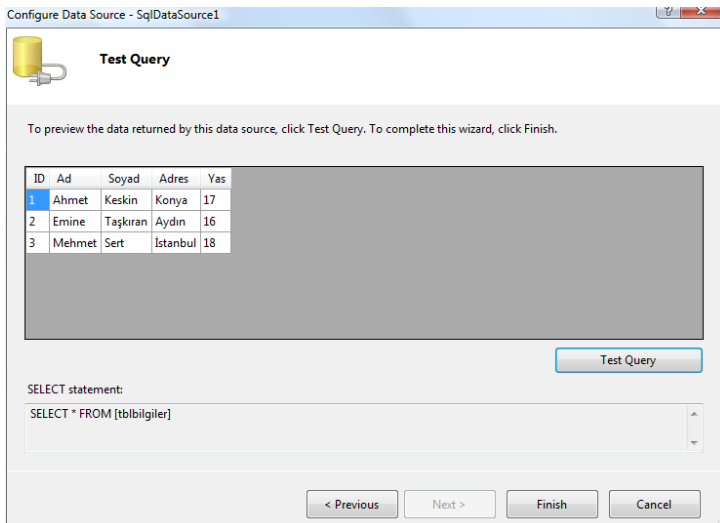


\*\*\* Connection string bölümündeki kod satırı daha sonraki uygulamalarımızda veritabanına bağlantı için kullanılacak \*\*\*

- 2) Daha sonra Next butona basarak Connectionstring parametresi gelecektir. Burayı geçmek içinde next butonuna basınız,
- 3) Daha sonraki pencerede işlemlerin görülmesi gereken kolonları seçmemiz gerekecektir. Bütün kolonları seçmek için " \* " checkbox'ını tıklayarak "**Advanced**" butonuna basınız. Daha sonra ekle, güncelle ve sil seçeneklerini ayarlayacağımız seçenek işaretlenir.

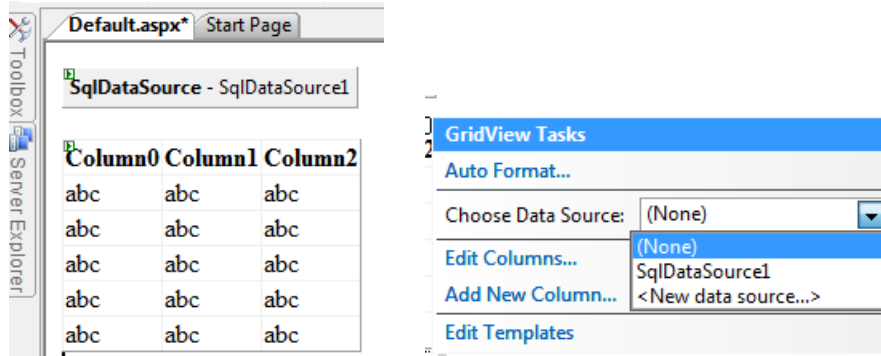


- 4) Daha sonraki pencerede işlemimizin sonuna gelmiş bulunmaktayız. Test Query butonuna bastığınızda karşımıza tblbilgiler deki tüm datalar gelecektir. İşlemimizi Finish diyerek bitirelim.

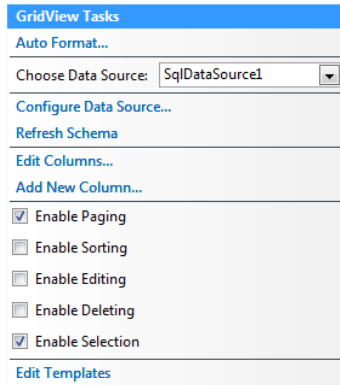




- 5) Böylece SqlServer üzerinde bulunan Ogrenci veritabanındaki tblOgrenciler tablosuna bağlanmış durumdayız.
- 6) Web sayfamıza bir adet GridView koyunuz, GridView nesnemizi tıklayarak properties penceresinden DataSourceID özelliğinden SqlDataSource1 seçeneğini tıklayınız.



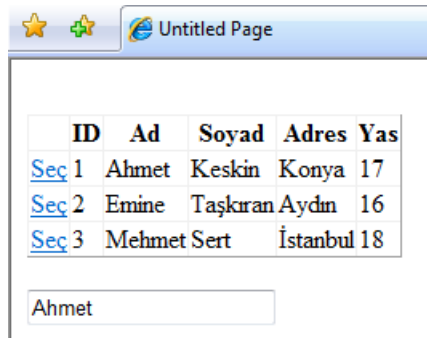
- 7) Artık kodumuzu çalıştırarak GridView'e datalarımızın aktarıldığını görebiliriz.
- 8) Database bağlantı işlemimiz bittikten sonra GridView seçeneğini tıkladığınızda üstte çıkan ok işaretini tıkladığınızda karşımıza gelen pencereden Enable Paging seçeneğini işaretleyerek sayfalarını oluşmasını sağlayabiliriz. Her sayfada görünecek kayıt sayısını properties penceresinden Page Size değerini değiştirerek ayarlayabilirsiniz.



- 9) Ek olarak bir textbox nesnesi ekleyelim, GridView'i dan Enabled Selection seçeneğini seçiniz. Böylelikle seçtiğimiz satırı belirlemiş olacağız.
- 10) 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;
}
```

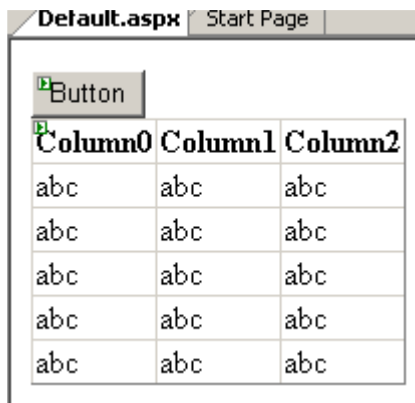
11) Böylece seçtiğimiz satıra ait Ad kolonu textbox'a aktarılacaktır.



### **ADO.NET Nesneleri ile Veritabanına Bağlantı:**

#### **Uygulama:**

1) Web sayfamıza birer GridView ve Button nesnelimizi ekleyelim.



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

```
using System.Data; //EKLEMEYİ UNUTMAYIN
using System.Data.SqlClient; //EKLEMEYİ UNUTMAYIN
protected void Button1_Click(object sender, EventArgs e)
{
```

@"Data

```
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\ogrenci.mdf;Integrated
Security=True;User
Instance=True";
```

```

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. Bağlantımızı 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= .\ SqlExpress  
Data Source=Selcuk\\SqlExpress  
Data Source=192.168.1.100

**AttachDbFilename:** Bu alana Sql Serverda oluşturduğumuz veritabanımızın ismini yazmamız gerekiyor.  
|DataDirectory|\ogrenci.mdf

**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 sonuç Dataset e aktarılarak sanal bir tablonun oluşturulması sağlanmaktadır. Artık DataSet'i GridView'e aktararak sonuçların görüntülenmesi sağlanacaktır.

### 4) Artık kodumuzu çalıştırarak butonumuza bastığımızda karşımıza datalarımız gelecektir.



Datalarımızı getirdikten sonra, veritabanımıza kayıt eklemek çıkartmak gibi sql sorgularla işlemlerimizi gerçekleştirebiliriz. Yeni bir uygulama ile bu adımları gerçekleştirelim.

### Uygulama:

1) Bu uygulamamızda Web sitemizi dört textbox, dört buton ve bir tanede GridView yerleştirelim,

```
@"Data
Source=.\SQLEXPRESS;AttachDbFi
lename=|DataDirectory|\T11A_A.
mdf;Integrated
Security=True;User Instance=True
";
```

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

2) Veritabanımıza ekleyeceğimiz bilgiler için Button1 olayına aşağıdaki kodu yazalım,

```
string ad, soyad, adres, yas;
ad = this.TextBox1.Text;
soyad = this.TextBox2.Text;
adres = this.TextBox3.Text;
yas = this.TextBox4.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");
```

3) Veritabanımızdaki bilgileri silmek için Button2 olayına aşağıdaki kodu yazalım,

```
string ad;
ad = this.TextBox1.Text;
baglanti = new SqlConnection(yol);
string sil = "delete from tblbilgiler where Ad='" + ad + "'";
sadp = new SqlDataAdapter(sil, baglanti);
```

```

verial = new DataSet();
sadb.Fill(verial);
Response.Write("Adı " + ad + " olan Kayıt Silindi");

```

#### 4) Veritabanımızdaki bilgileri Güncellemek için Button3 olayına aşağıdaki kodu yazalım,

```

string ad, adres, yas;
ad = this.TextBox1.Text;
adres = this.TextBox3.Text;
yas = this.TextBox4.Text;
baglanti = new SqlConnection(yol);
string guncelle = "Update tblbilgiler set adres='"+adres+"',yas='"+yas+"'
where
ad='" + ad + "'";
sadb = new SqlDataAdapter(guncelle, baglanti);
verial = new DataSet();
sadb.Fill(verial);
Response.Write("Yaş Güncellendi");

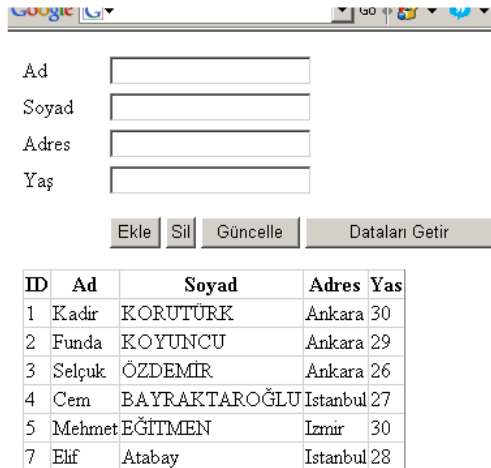
```

#### 5) Veritabanımızdaki bilgileri ekrana getirmek için Button4 olayına aşağıdaki kodu yazalım,

```

baglanti = new SqlConnection(yol);
string getir = "select * from tblbilgiler";
sadb = new SqlDataAdapter(getir, baglanti);
verial = new DataSet();
sadb.Fill(verial);
this.GridView1.DataSource = verial.Tables[0];
this.GridView1.DataBind();

```



ID	Ad	Soyad	Adres	Yaş
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

#### DATALIST:

En çok ve oldukça faydalı olan bir kontrol olan DataList birçok site yapma aşamanızda faydalı olacaktır. Şimdi düşünelim bir sitede sol tarafda kategorilerin bulunduğu ve bu kategorilere bağlı makale başlıkları ve bu makalelerinde detaylarını kullanıcılara okutmanın olduğunu kolay göreceksiniz.

Uygulamamıza başlamadan önce **ogrenci** veritabanımıza bir tablolar ekleyelim. Tablolalarımızın yapıları aşağıda yer almaktadır. Bu tabloların biri sol menü için kategoriler bulunacaktır. Diğer tablomuzda ise makale durumlarını bildirecek bir tablo olacaktır.

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"/>

Table - dbo.tblMakale	Table - dbo.tblKategori	Summary
Column Name	Data Type	Allow Nulls
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

Tablolarımıza girilen değerler aşağıda yer almaktadır,

Table - dbo.tblMakale	Table - dbo.tblKategori	Summary
ID	Kategori	KatDurum
1	C#.NET	csharp
2	ASP.NET	aspnet
3	C++.NET	cplus
4	Web Tasarımı	web

Dikkatli yazın

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

tblKategorideki KatDurum kolonlarını dikkatle yazınız.

tblMakale tablosundaki KayDurum kolonuna bağlı olarak sorgu çekilecektir.

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

### Uygulama:

- 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.

```
{
string yol = @"Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\ogrenci.mdf;Integrated
Security=True;User Instance=True ";
SqlConnection baglan = new SqlConnection(yol);
string sorgu = "select * from tblkategori";
SqlDataAdapter sadp = new SqlDataAdapter(sorgu, baglan);
DataSet ds = new DataSet();
sadp.Fill(ds);
this.DataList1.DataSource = ds.Tables[0];
this.DataList1.DataBind();
}
```

- 2) 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, DataList kod içerisine gelerek aşağıdaki kodu yazınız.

```
</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>
```

- 3) Kodumuzu yazdıktan sonra çalıştırdığımızda Kategoriler ekrana gelecektir.

C# .NET  
ASP.NET  
C++ .NET  
Web Tasarımı

4) 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.

5) Default.aspx deki source kısmındaki kodu değiştirerek kodumuzu çalıştıralım.

```
<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>
```

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

6) Artık Default2.aspx sayfamızda Gonder değişkenine ait Asp.Net linkini tıkladığımızdan dolayı aspnet değerini getirmiştir.

7) Şimdi ise Default2.aspx sayfasından bu değeri alarak buna göre makalelerimizin gelmesini sağlayacağız.

8) Default2.aspx sayfamıza bir tane Repeater kontrolü yerleştirelim. Default2.aspx'in Form\_Load olayına kodumuzu yazalım.

```
{
string gelenveri = Request.QueryString["Gonder"];
string yol = @"Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\ogrenci.mdf;Integrated
Security=True;User Instance=True";
SqlConnection baglan = new SqlConnection(yol);
string sorgu = "select * from tblmakale where KayDurum=" + gelenveri + """;
SqlDataAdapter sadp = new SqlDataAdapter(sorgu, baglan);
DataSet ds = new DataSet();
sadp.Fill(ds);
this.Repeater1.DataSource = ds.Tables[0];
this.Repeater1.DataBind();
}
```

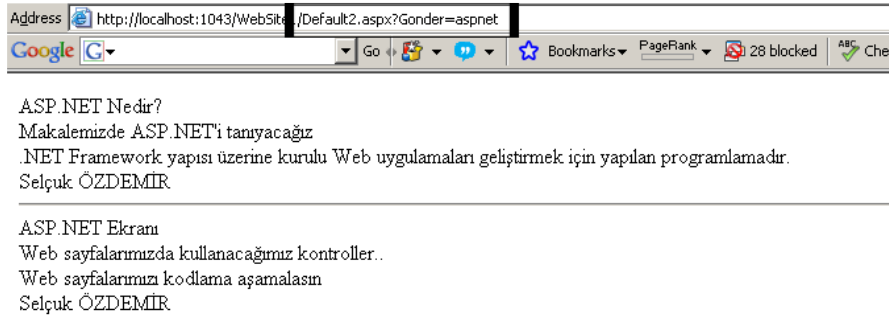
9) Kodumuzu yazdıktan sonra datalarımızın gelmesi DataList nesnemize benzemektedir. Bunun için Source kısmına gelerek kodu yazınız.

```

<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>

```

- 10) Default.aspx sayfamızı çalıştırarak kategorilerden birini tıkladığımızda tblMakale den tıklanan linke göre makaleler listelenecektir.



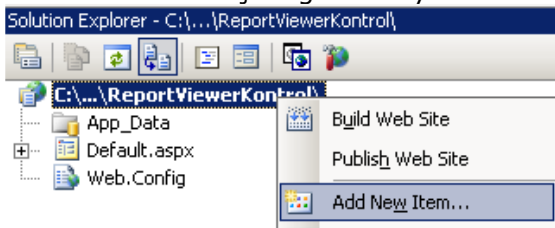
Bu uygulama sonucunda data üzerinde daha esnek kodlama söz konusu olacaktır.

## 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.

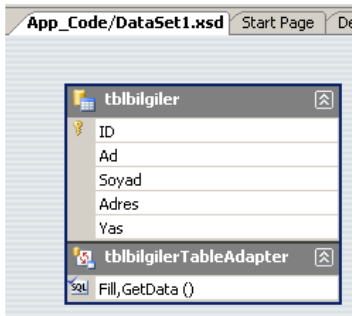
## Uygulama:

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

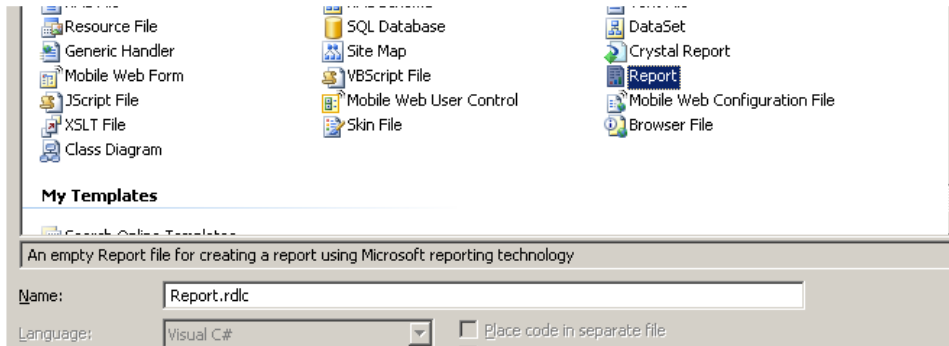




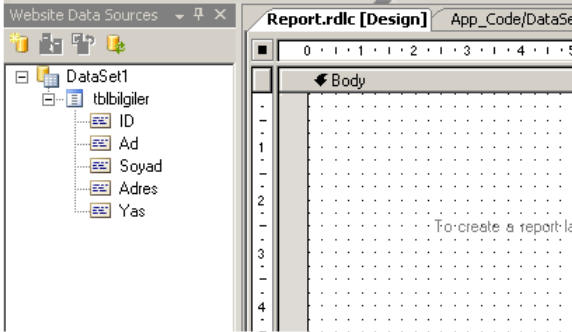
- 2) Karşımıza gelen Add New Item penceresinden DataSet objesini seçerek Add butonuna basınız. Karşımıza gelen uyarı penceresinde Yes diyerek geçelim. Solution explorerda App\_Code klasörü oluşacak ve içerisine DataSet1.xsd dosyasını eklendiğini göreceksiniz. Eklendikten sonra karşımıza bağlantı sihirbazı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



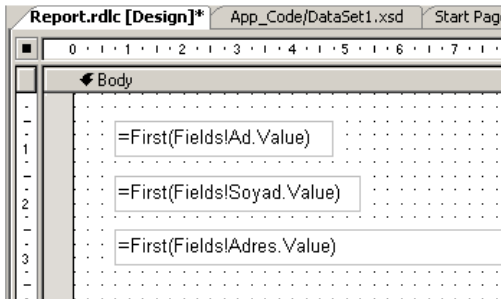
- 3) Yine projemizin üstünde sağ tıklayarak bu sefer Report nesnesini ekleyelim.



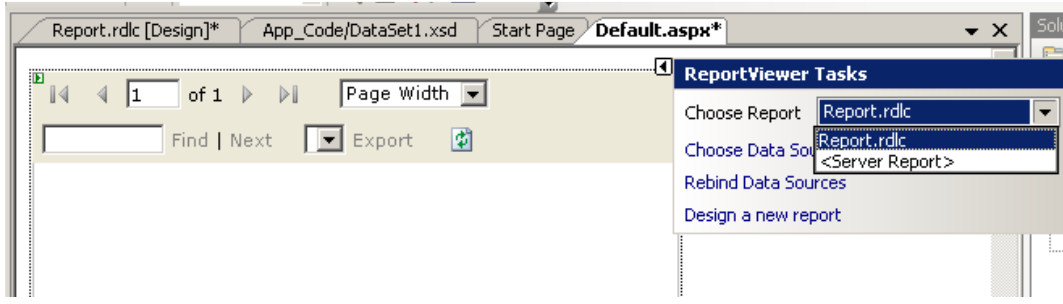
- 4) Report nesnemizi ekledikten sonra karşımıza Report Designı gelecektir. Kolonlarımızı DataSource kısmında görmemiz gerekiyor.



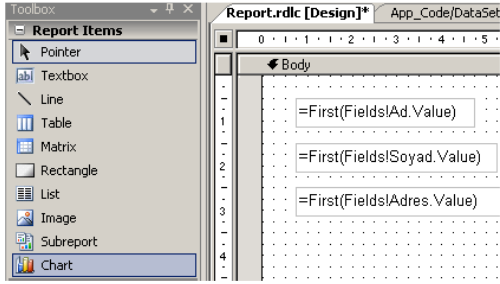
- 5) Kolonlardan Ad, Soyad ve Adres kolonlarını sürükleyerek tasarım alanına koyunuz.



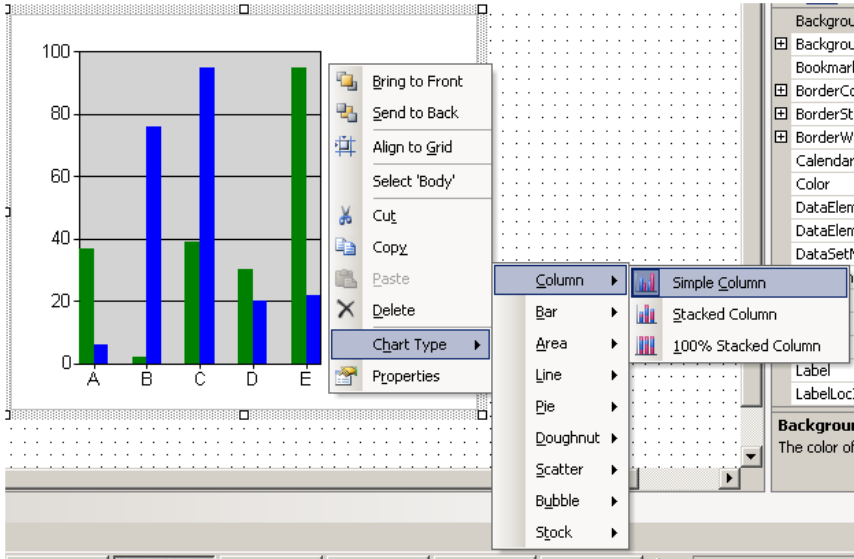
- 6) Default.aspx sayfasına ReportViewer kontrolü ekleyiniz. Chose Report kısmından Report.rdlc dosyanızı seçiniz.



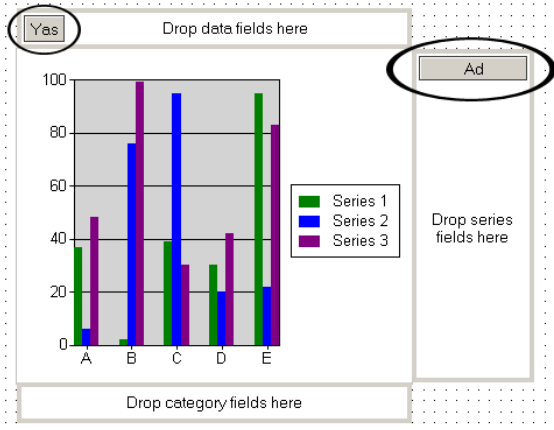
- 7) Kodumuzu çalıştırdığımızda kaydın geldiğini görebiliriz. Report tasarım alanına gelerek tasarım alanına chart (grafik) ekleyelim.



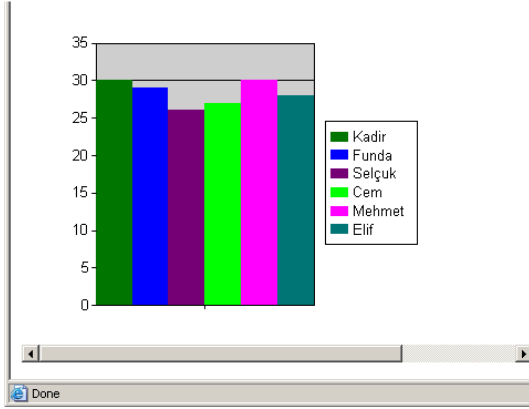
- 8) Grafik tipini değiştirerek istediğiniz görüntüyü elde edebilirsiniz. Tasarım alanına eklenin grafik'i sağ tıklayarak Chart type'ı belirleyiniz (Şekil 4.36). Biz örneğimizde Simple Column seçeneğinden göstereceğiz.



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



10) Default.aspx sayfamızı çalıştırarak grafiğimizin oluşması sağlanacaktır.



Böylelikle basit bir düzeyde raporlamada ReportViewer'ı kullanabiliriz. Daha profesyonel raporlamalar için CrystalReport kullanılacaktır.