

**T.C.  
MİLLÎ EĞİTİM BAKANLIĞI**

# **BİLİŞİM TEKNOLOJİLERİ**

**NESNE TABANLI PROGRAMLAMADA  
WEB UYGULAMALARI  
482BK0166**

**Ankara, 2012**

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- **PARA İLE SATILMAZ.**

# İÇİNDEKİLER

|   |    |
|---|----|
| AÇIKLAMALAR .....   | ii |
| GİRİŞ .....   | 1  |
| ÖĞRENME FAALİYETİ-1 .....                                     | 3  |
| 1. ASP.NET.....   | 3  |
| 1.1. Web Sunucu .....   | 3  |
| 1.2. ASP.NET ile Web Uygulamaları Oluşturma .....             | 4  |
| 1.2.1. Web Form Oluşturma .....                               | 5  |
| 1.2.2. Web Formunu Test Etme .....                            | 13 |
| 1.3. Tema Oluşturma ve Kullanma .....                         | 14 |
| UYGULAMA FAALİYETİ .....                                      | 16 |
| ÖLÇME VE DEĞERLENDİRME .....                                  | 18 |
| ÖĞRENME FAALİYETİ-2 .....                                     | 20 |
| 2. WEB FORM.....  | 20 |
| 2.1. Veriyi Web Sunucusunda Doğrulama.....                    | 20 |
| 2.2. Veriyi Web Tarayıcısında Doğrulama .....                 | 23 |
| 2.3. İstemci Doğrulaması .....                                | 23 |
| 2.3.1. RequiredFieldValidator Kontrolü .....                  | 24 |
| 2.3.2. RangeValidator Kontrolü.....                           | 26 |
| 2.3.3. ValidationSummary Kontrolü .....                       | 27 |
| UYGULAMA FAALİYETİ .....                                      | 29 |
| ÖLÇME VE DEĞERLENDİRME .....                                  | 31 |
| ÖĞRENME FAALİYETİ-3 .....                                     | 32 |
| 3. GÜVENLİK VE ERİŞİM .....                                   | 32 |
| 3.1. Form Tabanlı Güvenliği Yönetme .....                     | 32 |
| 3.2. Veriyi Görüntülemek ve Sorgulamak .....                  | 40 |
| 3.3. Veriyi Sayfalarda Görüntüleme .....                      | 43 |
| 3.4. Veriyi Düzenleme .....                                   | 44 |
| 3.5. Formlar Arasında Gezinme .....                           | 47 |
| UYGULAMA FAALİYETİ .....                                      | 52 |
| ÖLÇME VE DEĞERLENDİRME .....                                  | 54 |
| ÖĞRENME FAALİYETİ-4 .....                                     | 55 |
| 4. WEB SERVİSLERİ .....                                       | 55 |
| 4.1. Web Servisi .....  | 55 |
| 4.2. SOAP (Simple Object Access Protocol) Rolü .....          | 56 |
| 4.2.1. Web Servisi Tanımlama Dili .....                       | 56 |
| 4.2.2. Web Servislerinin İşlevsel Olmayan Gereksinimleri..... | 56 |
| 4.2.3. Windows Communication Foundation ‘ ın Rolü.....        | 57 |
| 4.2.4. Web Servisi Oluşturma.....                             | 57 |
| 4.3. Web Servisleri, İstemciler ve Proxy’ler .....            | 62 |
| UYGULAMA FAALİYETİ .....                                      | 63 |
| ÖLÇME DEĞERLENDİRME.....                                      | 65 |
| MODÜL DEĞERLENDİRME .....                                     | 66 |
| CEVAP ANAHTARLARI .....                                       | 69 |
| KAYNAKÇA .....  | 71 |

# AÇIKLAMALAR

|  |   |
|--|---|
| <b>KOD</b>                                     | <b>482BK0166</b>  |
| <b>ALAN</b>                                    | <b>Bilişim Teknolojileri</b>  |
| <b>DAL/MESLEK</b>                              | <b>Veritabanı Programcılığı</b>   |
| <b>MODÜLÜN ADI</b>                             | <b>Nesne Tabanlı Programlamada Web Uygulamaları</b>   |
| <b>MODÜLÜN TANIMI</b>                          | Öğrencinin, internet sayfalarında kullanacağı veriye web formları kullanarak güvenli bir şekilde erişebilme alışkanlığının kazandırıldığı bir öğrenme materyalidir.   |
| <b>SÜRE</b>                                    | 40/32   |
| <b>ÖN KOŞUL</b>                                | Nesne Tabanlı Programlamada Veri Yönetimi modülünü tamamlamış olmak   |
| <b>YETERLİK</b>                                | Web uygulamalarını oluşturabilmek   |
| <b>MODÜLÜN AMACI</b>                           | <b>Genel Amaç</b><br>Bu modül ile gerekli ortam sağlandığında; web uygulamaları oluşturabileceksiniz.<br><b>Amaçlar</b><br><b>1.</b> ASP.NET uygulamalarını hatasız olarak oluşturabileceksiniz.<br><b>2.</b> Web formlarını oluşturup çalışmalarını test edebileceksiniz.<br><b>3.</b> Web sunucuda ve tarayıcıda veri doğrulamayı hatasız bir şekilde yapabileceksiniz.<br><b>4.</b> Form üzerindeki ya da sayfada görüntülenen verilerin güvenliğini sağlayabileceksiniz.<br><b>5.</b> Bir web servisini amacına uygun olarak oluşturup kullanabileceksiniz. |
| <b>EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI</b> | <b>Ortam:</b><br>Atölye, laboratuvar, bilgi teknolojileri ortamı ( İnternet ) vb. kendi kendinize veya grupta çalışabileceğiniz tüm ortamlar<br><b>Donanım:</b><br>Nesne tabanlı programlama yazılımını çalıştırabilecek yeterlikte bilgisayar, yedekleme için gerekli donanım (cd yazıcı, flash bellek), raporlama için yazıcı, kâğıt ve kalem vb.   |
| <b>ÖLÇME VE DEĞERLENDİRME</b>                  | Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz.<br>Öğretmen modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış testi, boşluk doldurma, eşleştirme vb.) kullanarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek sizi değerlendirecektir.   |

# GİRİŞ

## Sevgili Öğrenci,

Bundan önceki bütün modüllerde öğrendiklerinizi uygularken daha çok Windows uygulamalarını kullandınız. Ancak bildiğiniz üzere bu işin web kısmı da kullanılmaktadır. Bu modülde C# dilinin zengin içeriğini kullanarak web uygulamaları geliştirmenin ne kadar kolay olduğunu göreceksiniz.

Bildiğiniz üzere internet, dünyanın bilinen en geniş ağıdır. Çünkü milyarlarca kullanıcısı mevcuttur. Bu kullanıcılar internet sayfalarını veya web uygulamalarını kullanarak değişik işlemler gerçekleştirmektedirler. Fatura ödemelerinden bankadaki hesaba ulaşma işlemlerine; sanal mağazalarda alışveriş yapmaktan arkadaşlarıyla olan iletişimlerine kadar pek çok işlem internet ortamında gerçekleştirilmektedir. İşte bu durum web uygulamalarının günümüzde ne kadar yaygın olarak kullanıldığının bir göstergesidir.

C# dilinin, güçlü C altyapısı ile geçerliliğinin giderek arttığı bundan önceki modüllerde sık sık dile getirildi. Ayrıca 2003 yılında piyasaya hızlı bir giriş yapan ASP.NET teknolojisi şu anda yaygınlığı giderek artan bir yapıya sahiptir. Böylece web uygulaması geliştirmek için güçlü bir dil ile web uygulamalarına yeni bir yol çizen ASP.NET teknolojisini birleştirmek, her zaman sizleri bir adım öne çıkaracaktır.



# ÖĞRENME FAALİYETİ-1

## AMAÇ

ASP.NET teknolojisini kullanarak bir web uygulaması oluşturup hatasız bir şekilde bu web uygulamasını kullanabileceksiniz.

## ARAŞTIRMA

- Günümüzde piyasada web uygulaması oluşturmak için kullanılan pek çok teknoloji mevcuttur. ASP ve ASP.NET bunlardan sadece iki tanesidir. Aslında ASP.NET, ASP'nin bir üst versiyonu olarak lanse edilmesine rağmen daha yeni bir teknolojidir. Sizler de ASP.NET' in ASP teknolojisinden ne gibi üstünlükleri olduğunu araştırınız. Bu araştırmanızı öğretmeninize teslim edecek veya sınıfta tartışacak şekilde hazırlayınız.

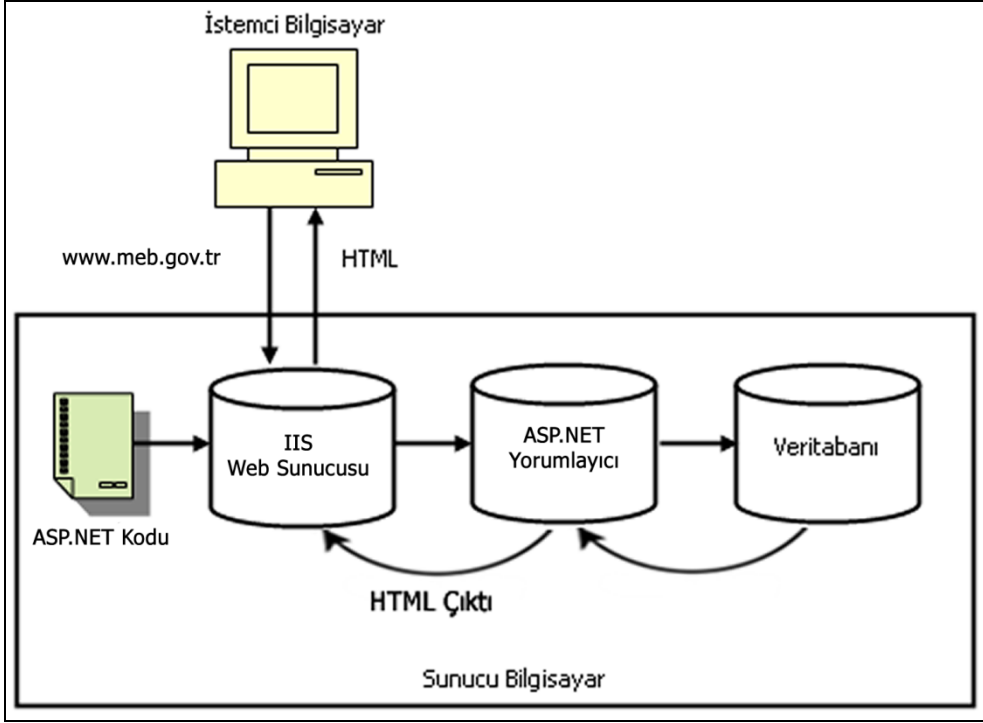
## 1. ASP.NET

İnternet, sunucu-istemci modeline göre çalışır. Fiziksel olarak bu durum, genelde istekte bulunan bir bilgisayar (istemci) ve bu isteği karşılayan özel bir bilgisayar (sunucu) anlamına gelmektedir. Tarayıcıda, bir web sayfası adresini yazıp *Enter* tuşuna basmak bir istekte (Request) bulunmaktır. Bu isteğimizi bir yerlerde ki bir sunucu karşılar. Bu modelin diğer adı istek-cevap (Request-Response) şeklindedir. İnternetteki bütün web sayfalarının çalışma prensibi bu modele dayanır. Bir istek olmadan herhangi bir şekilde bir sunum gerçekleşmez. Bu modelin her iki ayağı üzerinde çalışan web sitelerinin işleyişinde her bir ayak farklı çalışma mantığına ve metotlarına sahiptir. En basit hâliyle bir web sayfası, içinde sadece HTML kodlarını barındıran bir yapıya sahiptir. Bu sayfa internette herhangi bir sunucu bilgisayarda muhafaza edilir ve talep gelmesi durumunda bir kopyası istemciye gönderilir. Burada sunucu, sadece sayfanın bir kopyasını ( herhangi bir işlem yapmadan) istemciye gönderir. İstemcinin bilgisayarında ki tarayıcı gelen bilgiyi (HTML bilgisi) yorumlayarak doğru bir biçimde gösterir.

### 1.1. Web Sunucu

İnternet dosyalarının tamamını barındıran, herhangi bir istekte bu dosyalardan birini veya bir kaçını istemciye gönderen programın adıdır. Bu program kendine gelen isteklere göre tarayıcıya istenen bilgileri gönderen bir programdır. Aşağıdaki Resim 1-1'deki resmi inceleyiniz. Bu resimde;istemci bilgisayardan gelen bir isteğin ,web sunucu tarafından nasıl cevaplandırıldığını görebilirsiniz. Bu resimde istemci bilgisayar [www.meb.gov.tr](http://www.meb.gov.tr) adresini

sunucudan talep etmektedir. Bu talebi ilk karşılayan web sunucu yazılımıdır. Eğer talep edilen sayfa sadece HTML kodlarından oluşuyorsa bu durumda web sunucu sayfayı istemciye gönderir. Eğer istenen sayfada ASP.NET kodları kullanılmışsa bu durumda web sunucu, bu kodların HTML koduna dönüştürülmesi için ASP.NET yorumlayıcısına başvurur. ASP.NET yorumlayıcısı, kodları HTML koduna dönüştürürken veritabanı etkileşimi kullanılmışsa ilgili veritabanındaki verilerle ilgili işlemleri gerçekleştirerek kodların tamamını HTML koduna dönüştürerek web sunucuya gönderir. Web sunucu da sayfanın en son halini ama yine HTML hâlini istemciye gönderir.



**Resim 1.1: ASP.NET' in Web sunucuda çalışması**

Resim 1-1'de görüldüğü gibi istemci bir web uygulamasını talep ederken http'yi (Hiper Metin İletim Protokolü) kullanır. Web sunucuda yine istemciye cevap verirken http'yi kullanır. HTML (Hiper Metin İşaretleme Dili), pek çok tarayıcının anladığı ve görüntüleyebildiği bir dildir.

## 1.2. ASP.NET ile Web Uygulamaları Oluşturma

Bu bölümde kullanacağımız ASP.NET güçlü bir yapı olarak karşımıza çıkmaktadır. Bu yapının özellikleri aşağıda listelenmiştir. Bu özellikleri dikkatlice inceleyerek ASP.NET ' in karşımıza nasıl güçlü bir yapı olarak çıktığını anlayabilirsiniz.

- Görsel bir program hazırlar gibi (sürükle-bırak mantığı ile) formunuzu rahat ve hızlı bir şekilde oluşturabilirsiniz.



- Sunum ekranını ayrı bir sayfaya, kodlamayı da ayrı bir sayfaya yaparak uygulamanızın okunabilirliğini artırabilirsiniz.
- ASP.NET kodlamayı yapabilmek için ayrıca bir programlama dili öğrenmeye gerek yoktur. NET Platformunun size sunduğu herhangi bir programı (C#, VB, C++,... vb.) kullanarak kodlamayı gerçekleştirebilirsiniz. ASP.NET formları derlenerek performansı artırmak için web sunucuda önbelleğe alınır.
- Veritabanından alınan verileri görüntülemek, düzenlemek ve bakımını yapmak için güçlü veri kontrolleri sağlar. Bu kontrollerle verilerinize hızlı bir şekilde erişebilir veya düzenleyebilirsiniz.
- Bir uygulamadaki tüm web sayfalarına ortak bir düzen sağlamak için MasterPage yapısını; web sitesinde tutarlı bir görünüm ve hava oluşturmak için theme yapısını; kullanıcıların kendi gereksinimleri doğrultusunda özelleştirebilecekleri sayfalar oluşturmak için WebParts yapısını kullanabilirsiniz.
- Kullanıcıların web uygulamanıza kolayca erişebilmesine izin verebilir, oturum açmaya çalışan kullanıcıları doğrulayabilir ve web sitenize kimin eriştiğini bilebilmeniz için kullanıcı bilgisini sorgulayabilirsiniz.
- Web sitenizi yapılandırmak için görsel bir arabirim olan *Yönetimsel Araçlar* arabirimini kullanmanıza imkân sağlar.

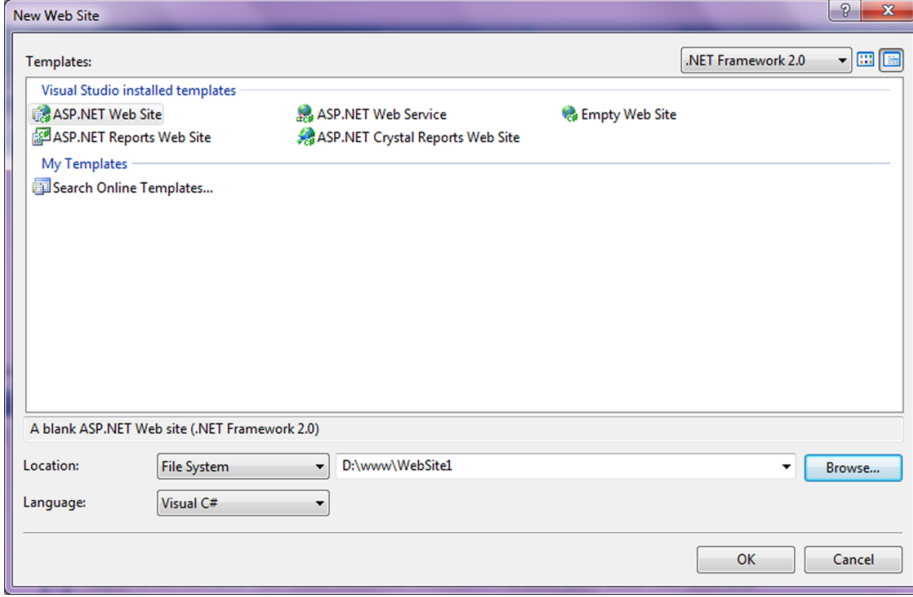
ASP.NET kullanan bir web uygulaması; tipik olarak bir ya da daha çok ASP.NET sayfası ya da web formu, kod dosyaları ve yapılandırma dosyalarından oluşur. Bu web formu, temelde NET Framework'a özel bir takım etiketler içeren bir HTML dosyası olan aspx uzantılı bir dosyada tutulur. Bir aspx dosyası, sayfanın görünümünü ve düzenini tanımlar. Genellikle bir takım işlemlerin gerçekleştirilmesi için kullanılan kodlamalar ayrı bir dosyada tutulur.

### 1.2.1. Web Form Oluşturma

Örnek olarak bir web uygulaması oluşturalım. Bunun için *File > New > New Web Site* seçeneğini tıkladığımızda aşağıdaki Resim 1-2'deki gibi bir ekranla karşılaşırız. Bu ekranda *Template, Location, Language*, olmak üzere üç ayarlama gerçekleştireceksiniz.

- *Template*; nasıl bir web uygulaması oluşturacağını belirlediğiniz kısımdır. Bu uygulama için *ASP.NET Web Site* seçeneğini seçmelisiniz.
- *Location*; uygulamanızı bilgisayarınızda nereye saklayacağınızı ayarlamak için kullandığınız kısımdır. *File System, HTTP* ve *FTP* seçeneklerinden oluşmaktadır. *File System*, projenizi bilgisayarınızda bir yere saklamak için kullanılır. *HTTP*, eğer bilgisayarınız aynı zamanda *IIS* gibi bir web sunucu içeriyorsa bu seçeneği kullanmalısınız. *FTP*, ise uzak bilgisayardaki bir yere projenizi saklamak istediğinizde kullanmanız gereken seçenektir. Bu modüldeki bütün uygulamalarda bu seçenek *File System* olarak seçilecektir. Bu durumda uygulamanız, kullanmış olduğunuz Nesne Tabanlı Programlama Yazılımı tarafından sağlanan geliştirici sunucusu (Development Server) tarafından sunulacaktır.

- Language; web uygulamanızı hangi kodlama dilini kullanarak yazacağınızı belirlediğiniz kısımdır. Daha öncede vurgulandığı gibi uygulama kodu olarak NET Framework'un desteklediği ve istediğiniz herhangi bir kodlama kullanabilirsiniz. Bu modülde C# kodlaması seçilecektir.

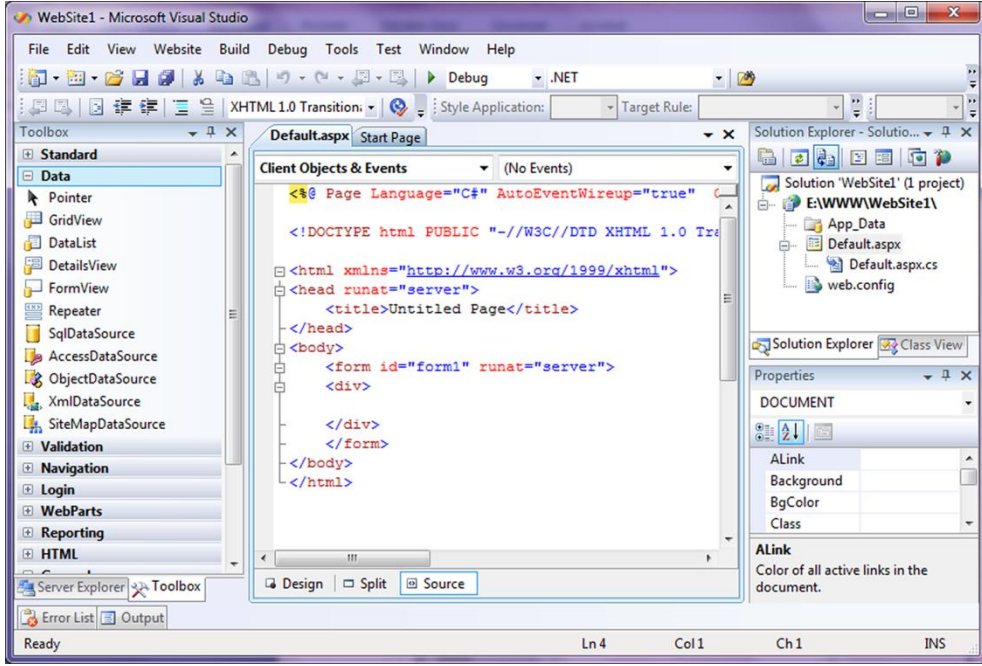


**Resim 1.2: Yeni bir web uygulaması oluşturma ekranı**

OK butonuna tıklayarak web sitenizi oluşturunuz. Karşınıza aşağıdaki Resim 1-3'te yer alan ekran gelmiş olacaktır. Bu ekranda menü çubuğu, araç çubukları, özellikler (Properties), araç kutusu (Toolbox) ve proje penceresi (Solution Explorer) olmak üzere paneller ve tasarım ekranı yer almaktadır.

- Proje Penceresi (Solution Explorer): Web sitenizde yer alan bütün öğeleri görüntüleyen, yeni öğe eklemek, silmek, adını değiştirmek, kopyalamak, taşımak gibi işlemlerin hızlı bir şekilde gerçekleştirilmesini sağlayan penceredir.
- Özellikler (Properties): Tasarım ekranında seçili olan bir öğe ile ilgili bütün özellikleri görüntüleyen ve bu özellikleri değiştirmenizi sağlayan penceredir.
- Araç Kutusu (Toolbox): Tasarım ekranına bir web form öğesi eklemek için kullanılan penceredir.

Resim 1-3'te görüldüğü gibi yeni bir web uygulaması oluşturduğunuzda proje penceresinde (Solution Explorer) de gördüğümüz gibi otomatik olarak dört adet öğe oluşmaktadır. Bunlar; App\_Data klasörü ve Default.aspx, Default.aspx.cs, Web.config dosyalarıdır.



Resim 1.3: Web uygulaması geliştirme ekranı

Web uygulamanızda oluşan bu dosyaların açıklaması aşağıdaki gibidir.

- App\_Data: Web uygulamanızda kullanacağınız veritabanı dosyalarını bulduracağınız klasördür.
- Default.aspx: Web uygulamanızın tasarım ve görsel öğelerden oluşan sayfasıdır. Web form öğeleri bu sayfada bulunur.
- Default.aspx.cs: Default.aspx görsel sayfasında meydana gelecek olayların kodlamasının bulunduğu sayfadır. Günümüzdeki web uygulaması geliştirme mantığına göre tasarım sayfası (görsel öğelerden oluşan kısım) ile kod sayfası ayrı dosyalarda tutulmaktadır. Nesne Tabanlı Programlama Yazılımı ile oluşturacağınız web uygulamalarında, isterseniz siz de sayfaları buradaki gibi ayırabilir ya da hem tasarım hem de kod işlemlerini tek bir sayfa ile de gerçekleştirebilirsiniz.
- Web.config: XML biçiminde olan bu dosya güvenlik, önbellek yönetimi, sayfa derlemesi, vb. işlemler için oluşturulmuş bir yapılandırma dosyasıdır.

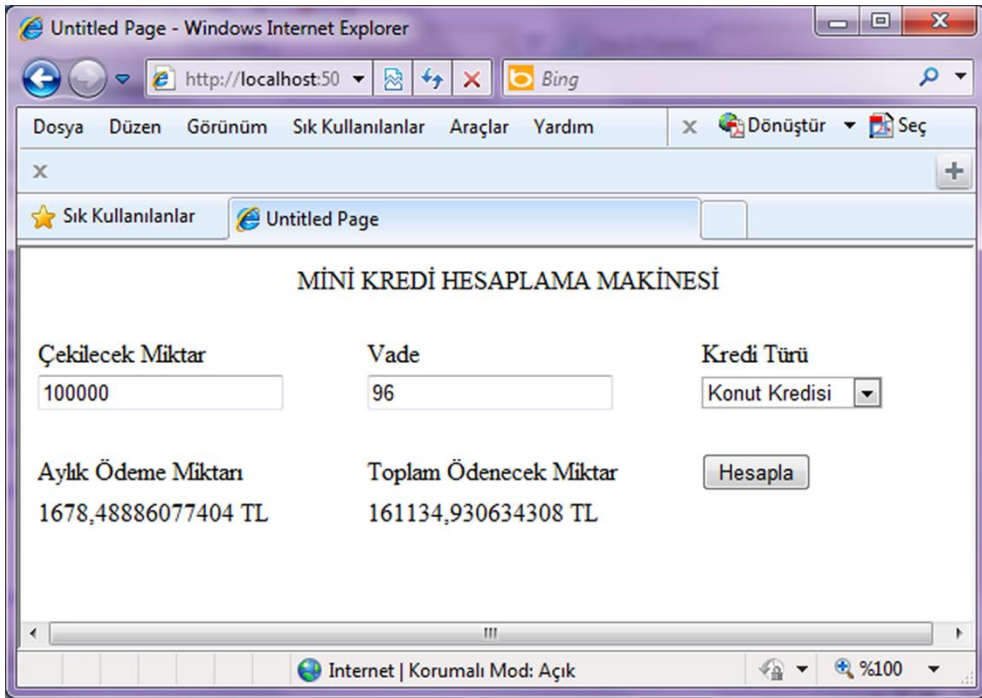
Ayrıca *Default.aspx* sayfasında yer alacak web formları, HTML etiketlerini görsel olarak oluşturabileceğiniz gibi kodlarla da oluşturabilirsiniz. Bunun için Nesne Tabanlı Programlama Yazılımı sizlere Resim 1-3'te gördüğünüz gibi *Source (kaynak)* görünümü sunmaktadır. *Design (tasarım)* görünümü; tasarım sayfasını görsel olarak oluşturmak için kullanılmaktadır. *Split (Bölünmüş)* ise ekranı ikiye bölerek hem görsel hem de kodlama ekranını görmek için kullanılan görünümdür.

Örnek 1-1: Bir bankanın kredi hesaplama programını web uygulaması olarak ve aşağıdaki kurallara göre gerçekleştiriniz.

- Kullanıcı kredi türü ve kredi miktarını klavyeden girecektir.
- Kredi türleri konut ve taşıt olarak belirlenecektir. Kullanıcı bunlardan herhangi birini seçebilecektir.
- Her bir kredi türünün farklı faiz oranları olacaktır. Konut kredisi %1,08 ve taşıt kredisi %1,25 olacaktır.
- Kredi hesaplaması için aşağıdaki formülü kullanınız.

$$Aylık\ Ödeme = \frac{anapara * faiz}{1 - \frac{1}{(1 + faiz)^{vade}}}$$

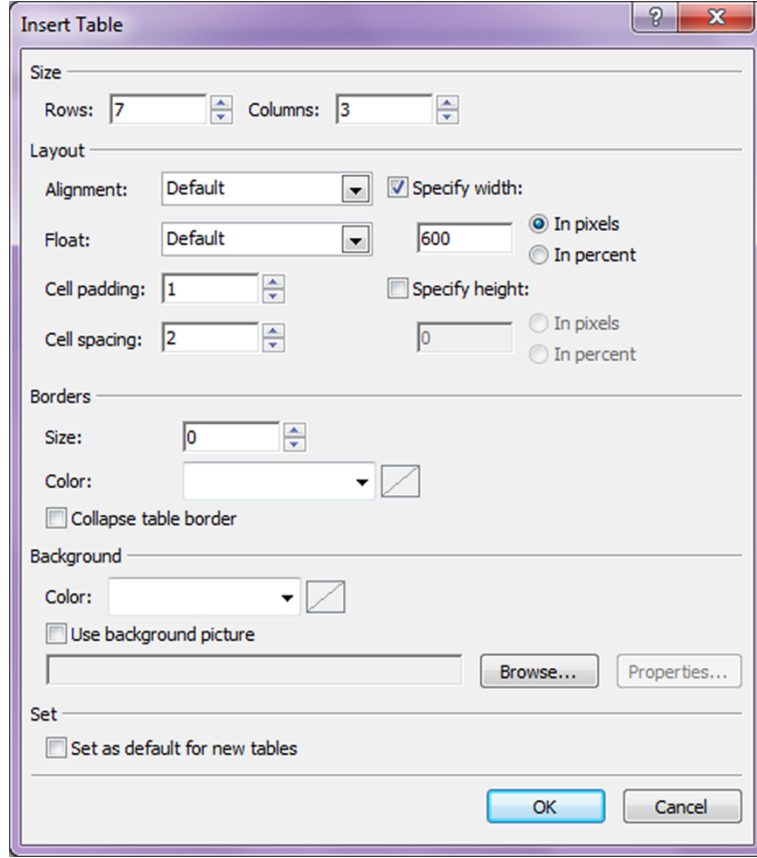
Web uygulamasının bitmiş hâli aşağıdaki Resim 1-4'te görüldüğü gibi olacaktır.



**Resim 1.4: Web uygulaması çıktı ekranı**

Yukarıdaki Resim 1-4'te görüldüğü gibi klavyeden “Çekilecek Miktar” ve “Vade Miktarı” girilip kredi türü seçilip “Hesapla” butonuna tıkladığında “Aylık Ödeme Miktarı” ve “Toplam Ödenecek Miktar” ekrana yazdırılmaktadır.

İlk olarak boş Default.aspx sayfanıza yedi satır ve üç sütundan oluşan tabloyu *Table > Insert Table* seçenekleriyle ekleyiniz. Bunun için karşınıza aşağıdaki Resim 1-5'te yer alan ekran gelecektir. Ayarlamaları bu ekrandaki gibi yapınız.



**Resim 1.5: Web sayfasına tablo ekleme ekranı**

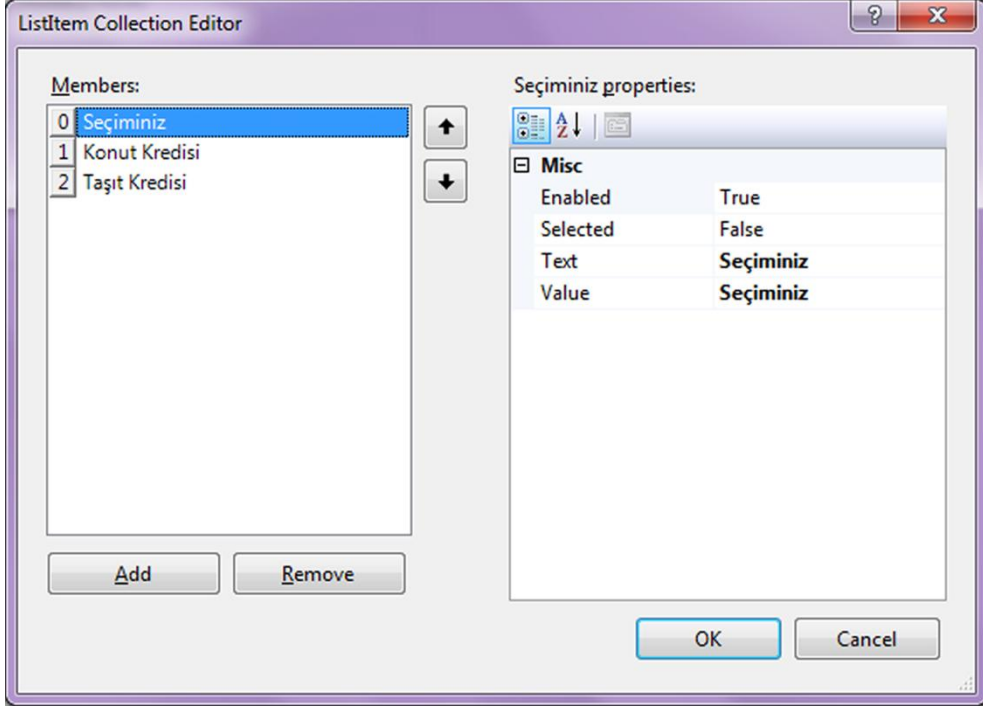
OK butonuna tıkladığınızda sayfanıza 600 piksel genişliğinde bir tablo eklenecektir. Bu tablonun en üst satırında yer alan hücreleri birleştiriniz. Bunun için birleştireceğiniz hücreleri seçiniz. *Table > Modify > Merge Cells* menü seçeneklerini tıklayınız. Tablonuzu aşağıdaki Resim 1-6’da görüldüğü gibi doldurunuz.

| MİNİ KREDİ HESAPLAMA MAKİNESİ  |                                |  |
|--------------------------------|--------------------------------|--|
| Çekilecek Miktar               | Vade                           | Kredi Türü                             |
| <input type="text" value="0"/> | <input type="text" value="0"/> | <input type="text" value="Seçiminiz"/> |
| Aylık Ödeme Miktarı            | Toplam Ödenecek Miktar         | <input type="button" value="Hesapla"/> |
| [Label1]                       | [Label2]                       |  |

**Resim 1.6: Program Design görünümü**

Resim 1-6’da görünen ekrandaki “Çekilecek Miktar” ve “Vade” için ekrana TextBox1 ve TextBox2 kontrolü ekleyiniz. “Kredi Türü” için Dropdownlist1 kontrolü, “Aylık Ödeme Miktarı” ve “Toplam Ödenecek Miktar” içinde Label1 ve Label2 kontrollerini, “Hesapla”

içinse Button1 kontrolünü ekleyiniz. “Kredi Türü” alanı için eklediğiniz DropDownList1 kontrolüne “Konut Kredisi” ve “Taşıt Kredisi” öğelerini giriniz. Bunun için DropDownList1 kontrolünü seçiniz. Özellikler (*Properties*) penceresindeki *Items* bölümüne tıklayınız. Karşınıza aşağıdaki Resim 1.7’de görünen ekran gelecektir.



**Resim 1.7: DropDownList kontrolüne öge ekleme ekranı**

Resim 1.7’de gördüğümüz gibi ekran Üyeler (Members) ve Özellikler (Properties) olmak üzere iki bölümden oluşmaktadır. *Add* butonu ile DropDownList kontrolüne öge ekleme, *Remove* butonu ile de öge çıkarma işlemi yapabilirsiniz. DropDownList kontrolünde görünecek olan metin, Özellikler (Properties) alanındaki *Text* bölümüne yazılır. Üyeler (Members) alanındaki metinlerin başında yer alan 0, 1 ve 2 sayıları öğelerin index numaralarını göstermektedir. Öğelere erişirken *Index* numaraları, *Value* değerleri ve *Text* değerlerini kullanabilirsiniz.

Resim 1.6’da yer alan “Hesapla” butonuna tıkladığınızda aşağıdaki Resim 1-8’deki gibi bir kodlama çalışacaktır. Bu kodlamada, DropDownList kontrolündeki hangi seçeneğin seçildiği *Index* numarası ile anlaşılacaktır (*DropDownList1.SelectedIndex* ile). *Math.Pow* fonksiyonu üs almak için kullanılmıştır (Örneğin,  $\text{Math.Pow}(2,3)=8$  ‘ dir.).

Daha öncede söylediğimiz gibi kod sayfası ile tasarım sayfasının ayrılmış hâli budur. Böylece web uygulamalarının optimizasyonu daha kolay, okunabilirliği daha rahat olacaktır.

```

protected void Button1_Click(object sender, EventArgs e)
{
    int anapara = Convert.ToInt32(TextBox1.Text);
    int vade = Convert.ToInt16(TextBox2.Text);
    int tur = Convert.ToInt16(DropDownList1.SelectedIndex);
    double faiz, aylık_odeme, toplam_odeme;
    if (tur == 1) //konut kredisi hesaplama
    {
        faiz = 0.0108;
        aylık_odeme = (anapara * faiz) / (1 - 1 / Math.Pow((1 + faiz), vade));
        toplam_odeme = aylık_odeme * vade;
        Label1.Text = aylık_odeme.ToString() + " TL";
        Label2.Text = toplam_odeme.ToString() + " TL";
    }
    else if (tur == 2) //taşıt kredisi hesaplama
    {
        faiz = 0.0125;
        aylık_odeme = (anapara * faiz) / (1 - 1 / Math.Pow((1 + faiz), vade));
        toplam_odeme = aylık_odeme * vade;
        Label1.Text = aylık_odeme.ToString() + " TL";
        Label2.Text = toplam_odeme.ToString() + " TL";
    }
}
}

```

**Resim 1.8: C# örnek program kodlaması**

Şimdi de web formumuza stil uygulayarak görünümün daha hoş olmasını sağlayacaksınız. Sayfamıza stil eklemek için *Format > New Style* menü seçeneklerini tıklayınız. Karşımıza aşağıdaki Resim 1-9'da yer alan ekran gelecektir.

Sayfanızda kullanmak üzere üç adet stili, aşağıdaki özelliklerde oluşturunuz.

*Baslık* stili için şu tanımlamaları yapınız.

Font: Arial

Size: 16px

Color: #000080

Font-Weight: Bold

*Metin* stili için şu tanımlamaları yapınız.

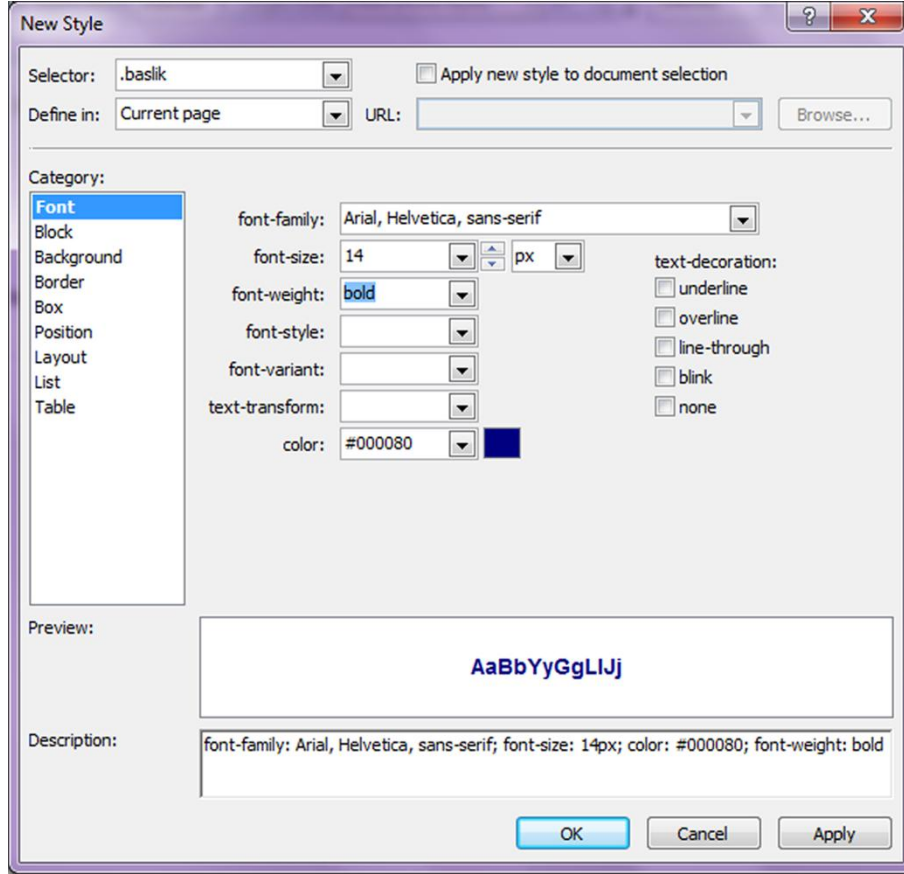
Font: Tahoma

Size: 14px

*Kontrol* stili için şu tanımlamaları yapınız.

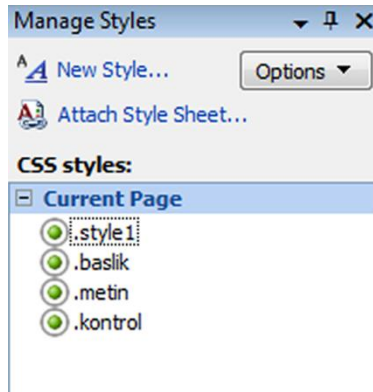
Font: Tahoma

Size: 12px



**Resim 1.9: Stil oluřturmak iin kullanılan ekran**

Stilleri tanımladıktan sonra dzenlenmesi, silinmesi veya yeni stil ilve edilmesi gibi iřlemleri *View > Manage Style* men seenekleri ile karřınıza Resim 1.10'daki gibi bir ekranda gerekleřtirebilirsiniz.



**Resim 1.10: Stil dzenleme paneli**




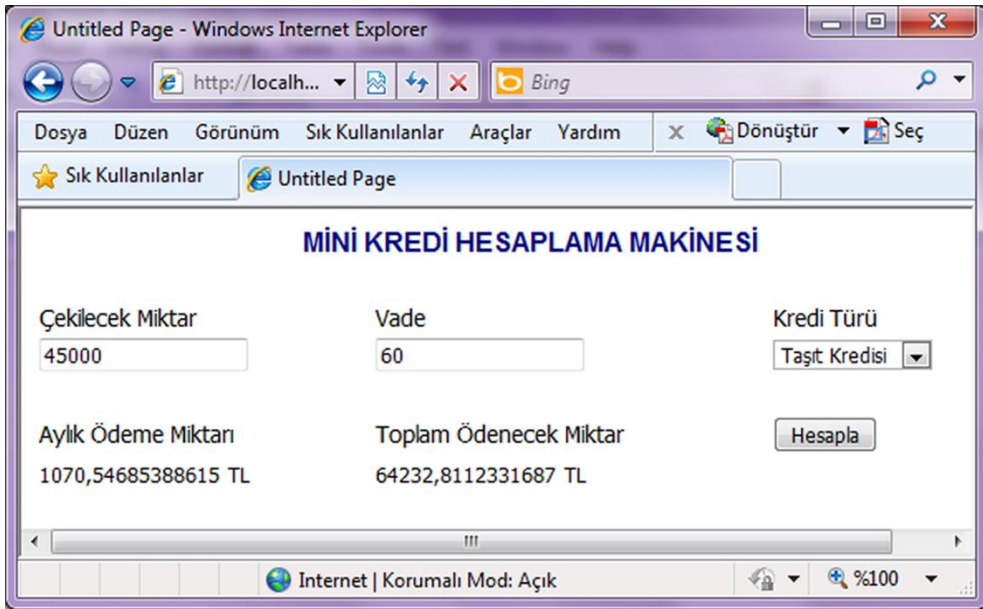
*Baslik* stilini “Mini Kredi Hesaplama Makinesi” metnine uygulayınız. Bunun için bu metnin bulunduğu hücreyi seçtikten sonra özellikler (properties) penceresindeki *Class* özelliğini *baslik* olarak değiştiriniz.

Kontrol stilini sayfada kullandığımız textbox, label, buton ve dropdownlist kontrollerine uygulayınız. Bunun için ilgili kontrolü seçtikten sonra özellikler (properties) penceresindeki *CssClass* özelliğini kontrol olarak değiştiriniz.

Sayfadaki geri kalan diğer yazılar için *metin* stilini, *baslik* stilini uyguladığımız gibi uygulayınız. Böylece sayfanıza stil uygulama işini bitirmiş olunuz.

### 1.2.2. Web Formunu Test Etme

Windows uygulamalarında olduğu gibi yapılan her uygulamanın test edilmesi o uygulamanın çalışma anında nasıl görüldüğünü görmek açısından önemlidir. Nesne Tabanlı Programlama Yazılımında geliştirdiğiniz web uygulamalarını da test ederken standart araç çubuğundaki  (Start Debugging) simgesini, Debug > Start Debugging menü seçenekleri ya da F5 kısayol tuşunu kullanabilirsiniz. Şimdi yukarıda oluşturmuş olduğunuz web uygulamasını, bu seçeneklerden herhangi birisiyle test ederek tarayıcıda nasıl görüldüğüne bakın. Web uygulamanız muhtemelen aşağıdaki Resim 1.11’deki gibi görünecektir.



Resim 1.11: Stil uygulanmış sayfa görünümü

### 1.3. Tema Oluşturma ve Kullanma

Web sitesini ilk kez oluşturduğunuzda form için stil tanımlamaları yaptınız. Bu stil form üzerindeki öğelerin yazı tipini, büyüklüğünü, rengini belirledi (Stillerle daha pek çok işlem yapılabilmektedir.). Bununla birlikte, bu şekilde tanımlanan bir stil yalnızca tek bir formda kullanılabilir. Pek çok formdan oluşan daha büyük bir web sitesinde tüm bu formları birbirleriyle tutarlı olacak biçimlendirilmiş hâlde tutmak zaman alıcı bir iştir. Yüzlerce sayfadan oluşan bir web sitesinde sadece yazı tipini değiştirmek istediğinizi düşünün. Bu durumda yapacağınız iş, formların bağlı olduğu stilleri ayrı ayrı değiştirmek olacaktır. İşte temalar (themes) burada bizim işimizi hayli kolaylaştıracaktır. Temalar, bir sayfa üzerindeki öğelere ya da bir web sitesindeki tüm sayfalara genel biçimde uygulayabileceğiniz özellikler, stiller ve resimler kümesidir.

Aslında temalarla yapabileceğiniz işlemi, bir.css stil dosyası ile da yapabileceğinizi düşünebilirsiniz. Ancak bu durumda.css stil dosyasındaki bir stil, sayfada yerel olarak tanımlanan bir stil varsa etki etmeyecektir. Temalar ise yerel olarak tanımlanan bütün stilleri etkisiz kılar. Gücü de buradan gelmektedir.

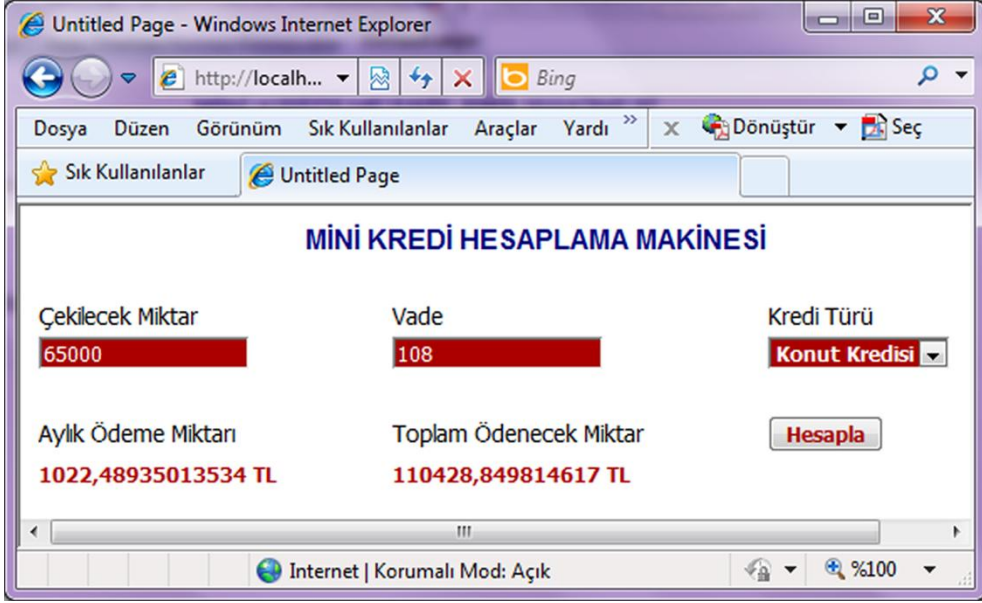
Bir tema, bir web sitesinin *App\_Themes* klasöründeki adlandırılmış bir alt klasörün içinde bulunan bir dosyadan ibarettir. Bu dosya.skin uzantısına sahip bir metin dosyasıdır. Aşağıda resim 1.12'deki web forma bir tema uygulanmıştır. Sizde bu tema uygulamasını adım adım gerçekleştiriniz.

- Proje Penceresinde (*Solution Explorer*), proje ismine farenin sağ tuşuna tıklayın. Gelen menüden *Add ASP.NET Folder > Theme* seçeneğini tıklayın. Proje penceresinde *App\_Themes* klasörünün ve altında *Theme1* alt klasörünün oluştuğunu göreceksiniz.
- *Theme1* alt klasörünün adını *Kirmizi* olarak değiştirin. Çünkü temanız kırmızı renk ağırlıklı olacaktır.
- Proje penceresinde (*Solution Explorer*), *Kirmizi* alt klasörünün isminin üzerinde farenin sağ tuşuna tıklayın. Gelen menüden *Add New Item* seçeneğini seçin. Gelen pencereden *Skin File* seçeneğini seçin ve dosyanın ismini *Kirmizi.Skin* olarak değiştirin. *Add* butonuna tıklayarak iletişim kutusunu kapatın.
- Skin dosyası kod penceresinde açık olarak ekrana gelecektir. Bu kodun en sonuna aşağıdaki kodları ilâve ediniz.

```
<asp:Textbox BackColor="#aa0000" ForeColor="#ffffff" Runat="Server"/>
<asp:Label BackColor="#ffffff" ForeColor="#aa0000" Runat="Server" Font-
Bold="True"/>
<asp:DropDownList BackColor="#aa0000" ForeColor="#ffffff"
Runat="Server" Font-Bold="True"/>
<asp:Button ForeColor="#aaffff" Runat="Server" Font-Bold="True"/>
```

Bu özellikler sayfanızda yer alan kontrollere etki edecektir. Aslında sayfanızdaki bu kontrollerde stiller olmasına rağmen, temanın bu stilleri etkisiz hâle getirerek kendi özelliklerini geçerli hale getirdiğini göreceksiniz.

- Tema uygulamak istediğiniz sayfayı Tasarım (Design) görünümünde açınız. Sayfanın boş bir yerine tıklayarak özellikler (Properties) panelinin açılması sağlayınız. Bu paneldeki Theme seçeneğini Kırmızı olarak değiştiriniz. Değişiklikleri kaydedip tekrar sayfanızı test ediniz. Sayfanızın en son görünümü aşağıdaki Resim 1.12’de görüldüğü gibi olmalıdır.



**Resim 1.12: Tema uygulanmış sayfa görünümü**

- Eğer temayı web sitenizdeki tüm sayfalara uygulamak isterseniz bu durumda projenizin web.config dosyasını aşağıdaki gibi değiştirin.

```
<configuration>
  <system.web>
    <pages theme="Kirmizi"/>
  </system.web>
</configuration>
```

Sizler de aynı uygulamayı gerçekleştirerek aynı sonuçları elde ediniz.

**Not:** Uygulamayı çalıştırdığınızda hata alırsanız solution Explorer penceresinden uygulama klasörünüze sağ tıklayarak property pages'ten. NET Framework versiyonunuzu kontrol ediniz.

## UYGULAMA FAALİYETİ

Bir firma internet üzerinden iş başvurusu kabul etmek için bir sayfa tasarlamak istemektedir. Bu sayfayı, kişilerin adı, soyadı, cinsiyeti, çalışmak istediği pozisyon, talep edeceği ücret şeklinde alanlar olacak şekilde tasarlayınız.

| İşlem Basamakları  | Öneriler   |
|--|--|
| ➤ Ad, soyad, cinsiyet, pozisyon, ücret metinleri için Label, bu alanlara veri girmek için TextBox kontrolünü kullanın.   |  |
| ➤ Sayfaya, çalışmak istenilen pozisyon için DropDownList sunucu denetimi ekleyin.  | ➤ DropDownList kontrolünün elemanlarına satış, muhasebe, reklam ve personel ekleyebilirsiniz.  |
| ➤ Sayfaya, cinsiyet alanı için RadioButtonList sunucu denetimi ekleyin.  | ➤ Bu kontrole öge eklemek tıpkı DropDownList şeklindedir. Öge olarak Bay ve Bayan ekleyebilirsiniz.  |
| ➤ Formun en sonuna “Gönder” isimli bir buton ekleyin. Butona tıklayınca kullanıcı tarafından girilen verileri aynı sayfaya yazdıracak şekilde kodlamasını yapınız. |  |
| ➤ Sayfadaki kontroller ve yazılar için stil tanımlaması yapınız.   | ➤ Sayfanın başlığı için baslik, Label kontrolü için yazi, TextBox, DropDownList ve RadioButtonList ve Button için de kontrol ismini kullanabilirsiniz. |
| ➤ Web siteniz için bir tema oluşturunuz.   | ➤ Bu tema Label, TextBox, DropDownList, RadioButtonList ve Button kontrollerini kapsayacak şekilde olabilir.   |
| ➤ Oluşturduğunuz temayı sayfaya uygulayınız.   | ➤ Tema uygulanınca sayfadaki stillerin nasıl geçersiz kaldığını görün.   |

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

| Değerlendirme Ölçütleri                          | Evet | Hayır |
|--|------|-------|
| 1. Web uygulaması oluşturduunuz mu?              |      |       |
| 2. Asp.Net sunucu denetimleri eklediniz mi?      |      |       |
| 3. Web formu için stil oluşturduunuz mu?         |      |       |
| 4. HTML denetimleri eklediniz mi?                |      |       |
| 5. Bir tema oluşturduunuz mu?                    |      |       |
| 6. Oluşturduğunuz temayı sayfaya uyguladınız mı? |      |       |

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme”ye geçiniz.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Oluşturulan bir temanın bütün web sitesinde kullanılabilir duruma gelmesi için bu temanın projenizdeki bir dosyada belirtilmesi gerekmektedir. Bu dosya aşağıdakilerden hangisidir?  
A) Web.config  
B) Default.aspx  
C) Default.aspx.cs  
D) Default.master
2. Web sitesinde oluşturulan temalar hangi klasör altında oluşturulur?  
A) Themes  
B) App\_Themes  
C) App\_Data  
D) Temalar
3. Yeni bir web uygulaması oluşturulduğunda proje penceresinde (solution explorer) hangisi oluşmaz?  
A) App\_Data  
B) Default.aspx  
C) Web.config  
D) App\_Themes
4. ASP.NET web siteleri için güvenlik, önbellek yönetimi, derleme gibi işlemler için kullanılan yapılandırma dosyası hangisidir?  
A) Web.config  
B) Default.aspx  
C) Default.aspx.cs  
D) App\_Data
5. ASP.NET web sitelerinde kullanılan tema dosyalarının uzantısı aşağıdakilerden hangisidir?  
A) aspx  
B) B) cs  
C) skin  
D) config

6. ASP.NET web sitesinde oluşturulan bir stilin TextBox kontrolüne uygulanabilmesi için bu kontrolün hangi özelliğine stil adı yazılır?
- A) CssClass
  - B) Style
  - C) Class
  - D) Css

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-2

## AMAÇ

Web formlara girilen verilerin doğruluğunu anlamak için istemci ve sunucu denetimlerini kullanabileceksiniz.

## ARAŞTIRMA

- Gündelik hayatınızda kullandığınız bilgilerin bazen doğruluğunu sağlamak isteyebilirsiniz. Örneğin; bankamatikten para çekeceğiniz zaman, şifre girerek kartın size ait olduğunu doğrulayabilirsiniz. Siz de bunun gibi doğruluğunu anlamının gerekli olduğu durumlara beş adet örnek veriniz. Örneklerinizi sınıfta arkadaşlarınızla paylaşacak ya da öğretmeninize teslim edecek şekilde hazırlayınız.

## 2. WEB FORM

Gerek Windows gerekse web uygulaması geliştirirken hep kontrollere doğru veriler girdiniz. Her şeyi olması gerektiği gibi yaptınız. Oysa gerçek hayatta bu böyle olmayacaktır. Bazı şeyler bazen olması gerektiği gibi olmayabilir. Örneğin, bir önceki öğrenme faaliyetinde yaptığımız “Kredi Hesaplama Makinesi” web uygulamasında kullanıcıdan almak istediği kredi miktarını TextBox kontrolüne girmesini istediniz. Kullanıcının normalde bu alana pozitif bir sayı girmesi gerekmektedir. Eğer kullanıcı buraya negatif bir sayı girerse hatalı bir giriş yapılmış demektir.

İşte bu durumları da düşünerek uygulama geliştirmek size her daim zaman kazandıracaktır. Bu hatalı girişleri, verileri sunucuya göndermeden gerçekleştirmek hem sunucuyu gereksiz yere yormayacak hem de zaman kazanmanıza yardımcı olacaktır. Ancak eğer doğrulayacağınız veri, veritabanında ise bu durumda bilgiyi veritabanındaki veriyle kıyaslamamız gerekmektedir. Böyle bir durumda bilginin sunucuya gönderilmesi gerekmektedir.

### 2.1. Veriyi Web Sunucusunda Doğrulama

Birtakım kurallara göre kullanıcının gireceği sunucuda veriyi doğrulamak, gereksiz yere sunucuyu yorabileceği gibi size ziyadesiyle zaman kaybettireceği için tercih edilmemelidir. Ancak kullanıcı tarafından girilen bilginin veritabanındaki bir bilgiyle mukayese edilmesi durumunda verinin elbette ki sunucuda doğrulanması gerekmektedir. Örneğin, kullanıcı girişi olan bir web sitesinde, kullanıcı adı ve şifre girmesi durumunda bu bilgilerin veritabanındaki bilgilerle mukayese edilmesi gerekmektedir. Bu ise ancak sunucuda olabilecek bir durumdur.



Yinede siz her şeye rağmen veriyi sunucuda doğrulamak isterseniz, *TextBox* kontrolüne girilecek verilerde *TextChanged* olayını kullanmanız gerekmektedir. *TextChanged* olayı, kullanıcının veriyi girip diğer kontrole geçmesi durumunda devreye girerek çalışan bir olaydır. İşte burumda veriyi rahatlıkla doğrulayabilirsiniz.

Örneğin bir önceki Öğrenme Faaliyeti-1’de yapmış olduğunuz “Mini Kredi Hesaplama Makinesi” web uygulamasını şu şekilde yeniden kodlayınız. Resim 2.1’de görünen program ekranındaki “Çekilecek Miktar” ve “Vade” alanı boş geçildiğinde *Label3* kontrolünde bu alanların boş geçilmemesi gerektiğini bildiren bir uyarı görünecektir.

MINİ KREDİ HESAPLAMA MAKİNESİ

Çekilecek Miktar: 0 Vade: 0 Kredi Türü: Seçiminiz ▼

Aylık Ödeme Miktarı: [Label1] Toplam Ödenecek Miktar: [Label2] Hesapla

[Label3]

Resim 2.1: Program ekranı

“Hesapla” butonuna tıkladığınızda çalışacak kodlar Resim 1.8’de yer almaktadır. Öncelikle bu kodları yazmalısınız. Bu kodları yazdıktan sonra aşağıdaki adımları uygulayınız.

Öncelikle “Çekilecek Miktar” alanı için kullanılan *TextBox1* kontrolüne tıklayın. Özellikler (Properties) penceresindeki *AutoPostBack* özelliğini *true* olarak değiştirin. Böylece *TextBox1* için tanımlanacak bir olayda sayfanın sunucuya gönderilmesini sağlamış oldunuz. Daha sonra *TextBox1* kontrolüne fareyle çift tıklayarak kodlama ekranına geçin. Kodlama ekranına geçtiğinizde *TextBox1* kontrolü için *TextChanged* olayı oluşmuş olmalıdır. Bu olay *TextBox*’da ki metnin değişmesi olayıdır. *AutoPostBack* özelliğinin *true* olduğu her durumda *TextBox* kontrolündeki veri değiştiğinde bu olay meydana gelecektir. Aşağıdaki Resim 2.2’de görünen kodu ekleyiniz.

```
protected void TextBox1_TextChanged(object sender, EventArgs e)
{
    if (TextBox1.Text == "")
    {
        Label3.Visible = true;
        Label3.Text = "HATA. Lütfen anapara alanını boş geçmeyiniz.";
        TextBox1.Focus();
    }
    else
        Label3.Visible = false;
}
```

Resim 2.2: TextBox1 kontrolü için gerekli kodlama

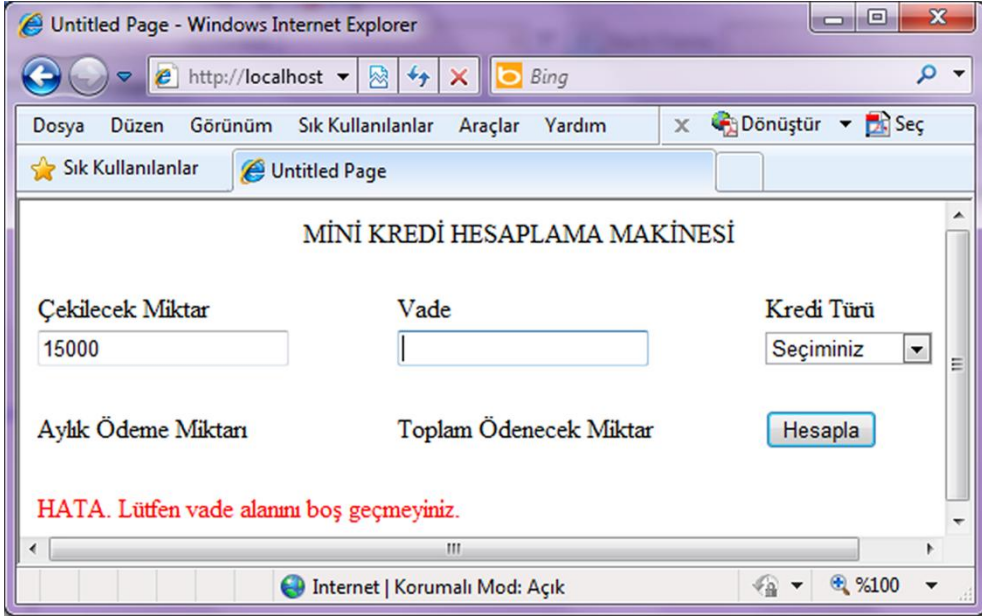
Resim 2.2’de gördüğünüz gibi TextChanged olayı gerçekleştiğinde yani TextBox1 kontrolündeki veri değiştiğinde sayfa sunucuya gidip tekrar yüklenmekte ve bu anda TextBox1 kontrolünün boş olup olmadığı kontrol edilmektedir. Eğer bu kontrol boşsa bu durumda Label3 kontrolünde bir uyarı görüntülenmekte ve TextBox1.Focus() ile de imleç tekrar TextBox1 kontrolüne konumlandırılmaktadır.

Aynı kodlamayı TextBox2 kontrolü içinde aşağıdaki Resim 2.3’te olduğu gibi yapınız.

```
protected void TextBox2_TextChanged(object sender, EventArgs e)
{
    if (TextBox2.Text == "")
    {
        Label3.Visible = true;
        Label3.Text = "HATA. Lütfen vade alanını boş geçmeyiniz.";
        TextBox2.Focus();
    }
    else
        Label3.Visible = false;
}
```

**Resim 2.3: TextBox2 kontrolü için gerekli kodlama**

Kodlamanızı bu hâliyle çalıştırdığınızda aşağıdaki Resim 2.4’te gördüğünüz ekran çıktısıyla karşılaşacaksınız.



**Resim 2.4: Sunucuda doğrulaması yapılan ekran çıktısı**

Gördüğünüz gibi “Vade” olarak tanımlanan *TextBox2* kontrolüne veri girilmediğinde ekrandaki uyarı karşınıza çıkacaktır.

## 2.2. Veriyi Web Tarayıcısında Doğrulama

ASP.NET web form modeli, asıl olarak doğrulamayı web tarayıcıda gerçekleştirir. Bu işlemi yaparken dinamik HTML'yi destekleyen tarayıcılarda doğrulama kontrollerini kullanır. Doğrulama kontrolleri (Validation), tarayıcıda çalışan ve sunucuya gidip gelme gerekliliğinden kaçınan JavaScript kodu oluşturur. Program geliştirici yalnızca ASP.NET doğrulama kontrollerini web form üzerine sürükleyip bırakır, özelliklerini ayarlar (ister properties penceresinden, ister kodlarla) ve gerçekleştirecek doğrulama kurallarını ve görüntülenecek hata iletilerini belirtir.

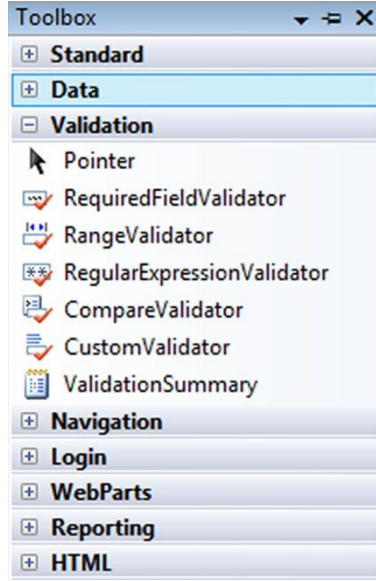
Doğrulama kontrolleri araç kutusunda doğrulama (validation) sekmesinde bulunur ve aşağıdakilerden oluşmaktadır.

- *RequiredFieldValidator*: Kullanıcının kontrole veri girmesine zorlamak için kullanılan denetimdir.
- *CompareValidator*: Girilen bir veriyi sabit bir değerle, başka bir kontrole girilen değerle ya da veritabanındaki bir veriyle kıyaslamak için kullanılır.
- *RangeValidator*: Kullanıcının kontrole veri girerken belirli aralıklarda veri girmesini sağlamak için kullanılır
- *RegularExpressionValidator*: Kullanıcının belirli bir format ya da biçime göre veri girmesini sağlamak için kullanılır.
- *CustomValidator*: Programcının kendi özel doğrulama mantığını tanımlaması için kullanılan denetimdir.
- *ValidationSummary*: Doğrulama denetimleri sonucu oluşan uyarı metinlerini biçimli bir şekilde görüntülemek için kullanılan denetimdir.

Her bir denetim tek bir tanımlı doğrulama türü gerçekleştirebiliyor olsa da birkaç tanesinden oluşan bir bileşimi kullanabilirsiniz. Örneğin, kullanıcının bir metin kutusuna veri girdiğinden ve aynı zamanda girilen değer belirli bir aralığın içinde olduğundan emin olmak istiyorsanız, metin kutusuna *RequiredFieldValidator* ve *RangeValidator* denetimlerini bağlayabilirsiniz.

## 2.3. İstemci Doğrulaması

Bu tür doğrulama ASP.NET tarafından sağlanan ve bir önceki konu başlığında açıklaması yapılan doğrulama kontrolleri ile yapılmaktadır. Bu kontroller tıpkı standart ve diğer kontroller gibi araç kutusundaki (*ToolBox*) *Validation* sekmesinde bulunur (Resim 2.5). Programcı bu kontrolleri sayfasına eklerken 'sürükle bırak' yöntemini kullanabilir. Aşağıda bu kontrollerin uygulamalı açıklamasını bulacaksınız. Lütfen dikkatlice inceleyip uygulayınız.



Resim 2.5: ToolBox da ki Doğrulama (Validation) kontrolleri

### 2.3.1. RequiredFieldValidator Kontrolü

Bu bölümde, sayfanıza bir *RequiredFieldValidator* kontrolü ekleyerek sayfanızdaki metin kutusuna veri girmeye zorlayacaksınız. Kullanıcı böylece metin kutusunu boş bırakamayacaktır. Eğer sayfanızdaki boş bırakılmaması gereken birden fazla metin kutusu varsa her bir metin kutusu için *RequiredFieldValidator* kontrolü kullanmalısınız.

Şimdi yeni bir web uygulaması oluşturunuz. Resim 1.6'da yer alan web form görüntüsünü aşağıdaki gibi değiştiriniz. Bu görüntü için gerekli kodlamayı Resim 1.8'teki gibi yapınız.

| MİNİ KREDİ HESAPLAMA MAKİNESİ   |                                |  |
|---|--------------------------------|--|
| Çekilecek Miktar  | Vade                           | Kredi Türü                             |
| <input type="text" value="0"/>  | <input type="text" value="0"/> | <input type="text" value="Seçiminiz"/> |
| Aylık Ödeme Miktarı   | Toplam Ödenecek Miktar         | <input type="button" value="Hesapla"/> |
| [Label1]  | [Label2]                       |  |
| ** Çekilen Miktar alanını boş geçemezsiniz... ** Vade alanını boş geçemezsiniz... |                                |  |

Resim 2.6: Doğrulama kontrolünün eklendiği ekran görüntüsü

Resim 2.6'daki ekran görüntüsünün en altına, iki adet *RequiredFieldValidator* kontrolü ekleyiniz. Bunlardan ilki "Çekilecek Miktar" alanındaki metin kutusu için diğeri de "Vade" alanındaki metin kutusu için olacaktır.

*RequiredFieldValidator1* kontrolü için özellikler (*properties*) penceresinden aşağıdaki ayarlamaları yapınız.

*ErrorMessage* özelliği; kullanıcıya, hata gerçekleştiğinde yani ilgili metin kutusunu boş bıraktığında gösterilecek uyarıdır. Bu uyarıyı, “\*\* Çekilen Miktar alanını boş geçemezsiniz...” şeklinde değiştiriniz.

*ControlToValidate* özelliği, ilgili doğrulama kontrolünün hangi metin kutusunu denetlediğini belirlediğiniz özelliktir. *RequiredFieldValidator1* kontrolü için *TextBox1* kontrolünü seçerek değiştiriniz.

*Display* özelliği, hata iletisinin görüntülenme biçimini belirleyen özelliktir. Doğrulama kontrolünün, hata iletisini sayfaya eklendiği yerde göstermesi için bu özelliği *static*, hata iletileri için belirlenen ortak bir yerde görüntülenmesi için bu özelliği *dinamic* olarak ayarlayınız. Bu uygulama için *display* özelliğini *dinamic* olarak ayarlayınız.

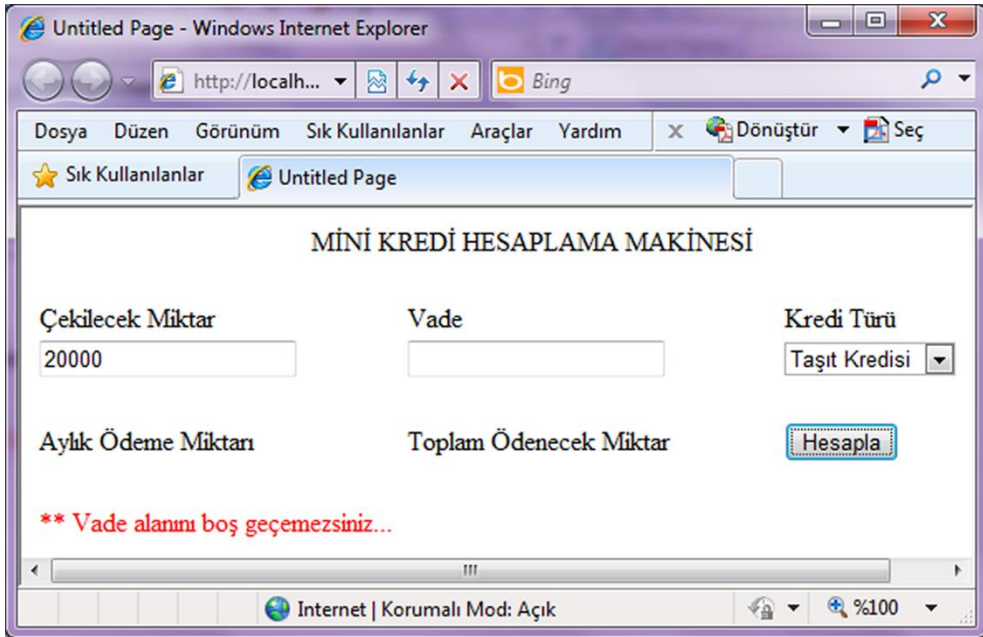
*RequireFieldValidator2* kontrolü için özellikler (*properties*) penceresinden aşağıdaki ayarlamaları yapınız.

*ErrorMessage*: “\*\* Vade alanını boş geçemezsiniz...”

*ControlToValidate*: *TextBox2*

*Display*: *Dinamic*

Uygulamanızı çalıştırdığınızda aşağıdaki Resim 2.7’de görünen ekranla karşılaşmalısınız.



**Resim 2.7: RequiredField kontrolünün çalışması**

Gördüğünüz gibi “Vade” alanındaki metin kutusu boş olduğu için sayfa sunucuya gönderilmeden hata iletisini vermektedir. Metin kutusu boşken “Hesapla” butonuna tıklasanız bile sayfa sunucuya gönderilmeyecektir.

### 2.3.2. RangeValidator Kontrolü

Bu bölümde, sayfanıza bir *RangeValidator* ekleyerek ilgili metin kutusuna girilen verinin, sizin belirlediğiniz bir aralıkta olup olmadığını anlayacaksınız. Kullanıcı böylece sizin belirlediğiniz sınır dışına çıkmayacaktır. Sayfanızdaki bu doğrulama kontrolüyle kontrol edeceğiniz birden fazla metin kutusu varsa her bir metin kutusu için ayrı *RangeValidator* kontrolü kullanmalısınız.

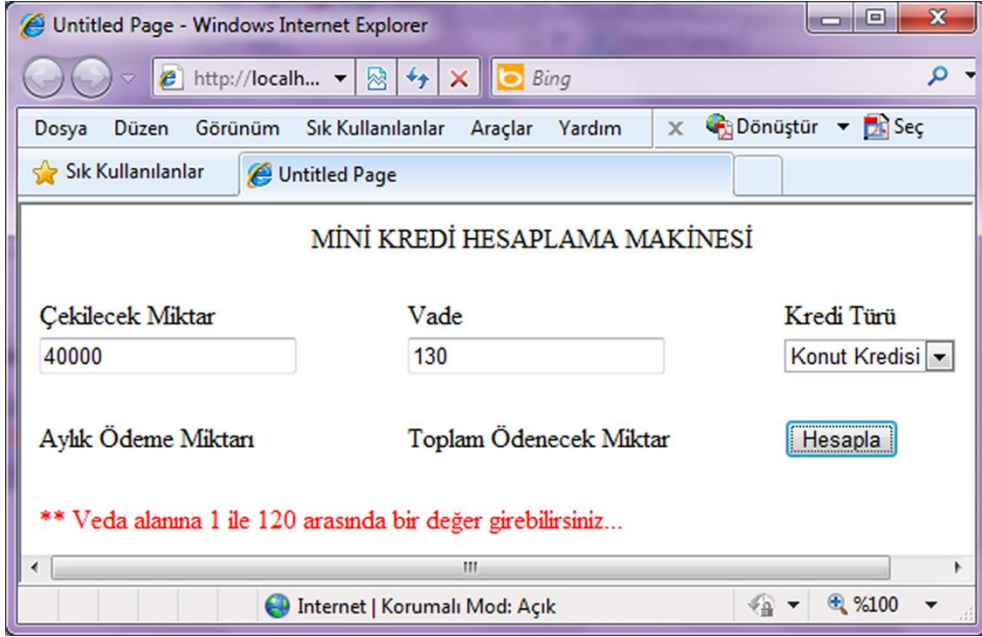
Şimdi yeni bir web uygulaması oluşturunuz. Resim 2.6’da yer alan web form görüntüsünde ki *RequiredFieldValidator2* kontrolünden sonra bir de *RangeValidator* kontrolü ekleyiniz. Bu görüntü için gerekli kodlamayı Resim 1.8’deki gibi yapınız. Bu uygulamada “Vade” alanındaki metin kutusuna girilecek sayının 1 ile 120 arasında olmasını sağlayacaksınız.

Resim 2.8: RangeValidator eklenmiş ekran görüntüsü

Şimdi sayfanıza eklediğiniz *RangeValidator1* kontrolü için özellikler (properties) penceresinden şu ayarlamaları yapınız.

ErrorMessage: “\*\* Vade alanına gireceğiniz sayı 1 ile 120 arasında olmalıdır.”  
ControlToValidate: TextBox2  
Display: Dynamic  
MaximumValue: 120  
MinimumValue: 1

Böylece kullanıcı TextBox2 kontrolüne 1 ile 120 aralığı dışında bir sayı girdiğinde sayfa sunucuya gönderilmeden hata ile karşılaşacaktır. Aşağıda böyle bir hata görmektesiniz.

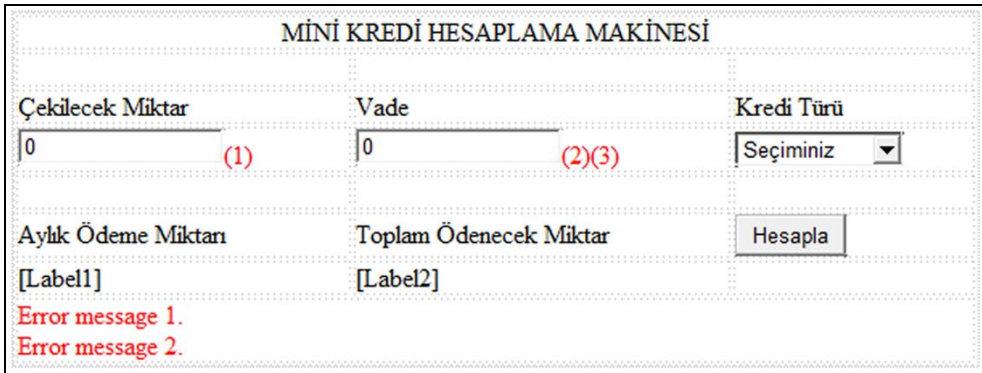


Resim 2.9: RangeValidator kontrolünün çalışması

### 2.3.3. ValidationSummary Kontrolü

Sayfanızda birden fazla doğrulama kontrolü kullanmışsanız bu kontrollerin görüntülemesini yaparken bir araya toplayan kontroldür. Eğer sayfanızda hata iletilerini görüntülemek için bir alanınız yoksa bu durumda, ValidationSummary kontrolü iletilerinizi MessageBox yöntemiyle de görüntülemenizi sağlamaktadır.

Yeni bir web site oluşturunuz. Bu web sitedeki bir sayfayı şu aşağıdaki yönergelerle göre Resim 2.10'da görünen ekran görüntüsünü oluşturunuz.



Resim 2-10: ValidationSummary kontrolü

Resim 2.10'da görünen ekran görüntüsündeki en altta yer alan kontrol (Error message 1. Error message 2) ValidationSummary kontrolüdür. Bu kontrolün Display özelliğini List olarak ayarlayın.

Resim 2.10’da (1) olarak görünen kontrol *RequiredFieldValidator1* kontrolüdür. Bu kontrolün özelliklerini şu şekilde değiştiriniz.

*ErrorMessage*: “(1) Çekilecek Miktar alanını boş geçemezsiniz...”

*ControlToValidate*: TextBox1

*Display*: Dinamic

*Text*: (1)

Resim 2.10’da (2) olarak görünen kontrol *RequiredFieldValidator2* kontrolüdür. Bu kontrolün özelliklerini şu şekilde değiştiriniz.

*ErrorMessage*: “(2) Vade alanını boş geçemezsiniz...”

*ControlToValidate*: TextBox2

*Display*: Dinamic

*Text*: (2)

Resim 2.10’da (3) olarak görünen kontrol *RangeValidator1* kontrolüdür. Bu kontrolün özelliklerini şu şekilde değiştiriniz.

*ErrorMessage*: “(3) Vade alanına gireceğiniz sayı 1 ile 120 arasında olmalıdır.”

*ControlToValidate*: TextBox2

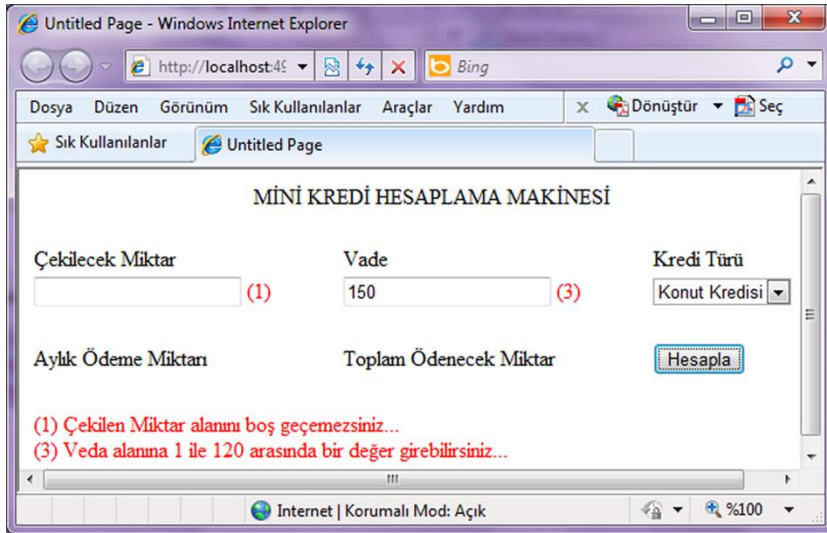
*Display*: Dynamic

*MaximumValue*: 120

*MinimumValue*: 1

*Text*: (3)

Programı çalıştırdığınızda aşağıdaki gibi bir ekran görüntüsüyle karşılaşmış olmalısınız.



**Resim 2.11: ValidationSummary kontrolü çalışması**



## UYGULAMA FAALİYETİ

Klavyeden iki yazılı (*TextBox1* ve *TextBox2*) ve bir sözlü notu (*TextBox3*) girilen bir dersin ortalamasını bulan programı web form olarak yapınız.

| İşlem Basamakları   | Öneriler   |
|---|--|
| ➤ Kullanıcının metin kutularını boş geçmemesi için doğrulama işlemini if yapısını da kullanarak sunucuda gerçekleştiriniz.    | ➤ Kullanıcı butona bastığında oluşan <i>Button_Click</i> olayında metin kutularını kontrol ederek doğrulamayı sağlayabilirsiniz. |
| ➤ Yukarıdaki doğrulamayı bu kez de <i>RequiredFieldValidator</i> kontrolünü kullanarak istemci tarafında gerçekleştiriniz.    | ➤ Her bir metin kutusu için ayrı <i>RequiredFieldValidator</i> kontrolü kullanabilirsiniz.                                       |
| ➤ Metin kutularına girilecek verilerin 0 ile 100 arasında olmasını sağlamak için <i>RangeValidator</i> kontrolünü kullanınız. | ➤ Her bir metin kutusu için olmak üzere üç adet <i>RangeValidator</i> kontrolünü kullanabilirsiniz.                              |
| ➤ Doğrulama sonucu oluşacak hata iletilerini bir arada görüntülemek için <i>ValidationSummary</i> kontrolünü kullanınız.      | ➤ <i>ValidationSummary</i> kontrolünün <i>Display</i> özelliğini <i>List</i> olarak ayarlayabilirsiniz.                          |

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

| Değerlendirme Ölçütleri   | Evet | Hayır |
|---|------|-------|
| 1. Kullanıcı girişlerinin sunucu tarafı doğrulamasını yapabildiniz mi?            |      |       |
| 2. Kullanıcı girişlerinin istemci tarafı doğrulamasını yapabildiniz mi?           |      |       |
| 3. Kullanıcıyı metin kutusuna veri girişi yapması için zorladınız mı?             |      |       |
| 4. Metin kutusuna girilen verinin belli bir aralıkta olmasını sağlayabildiniz mi? |      |       |

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme”ye geçiniz.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. ASP.NET ' in, kullanıcı girişlerini istemci tarafında doğrulamak için programcılara sunmuş olduğu kontroller araç kutusundaki (ToolBox) ilgili sekmede yer almaktadır. Bu sekmenin ismi aşağıdakilerden hangisidir?
  - A) Validation
  - B) Standart
  - C) Veri
  - D) Login
2. Aşağıdakilerden hangileri, kullanıcı girişlerini sunucu tarafında doğrulamanın dezavantajlarındanıdır?
  - I. Sunucuyu gereksiz yere yorar.
  - II. Zaman kaybına yol açar.
  - III. Ağ trafiğini meşgul eder.
  - A) I ve II
  - B) II ve III
  - C) I ve III
  - D) I, II ve III
3. Kullanıcı girişleri esnasında yapılan istemci doğrulamasında, kullanıcının metin kutusuna girdiği sayının 500 ile 800 arasında olması için hangi doğrulama kontrolü kullanılmalıdır?
  - A) RangeValidator
  - B) RequiredFieldValidator
  - C) ValidationSummary
  - D) CompareValidator
4. Aşağıdakilerden hangisi veya hangileri yanlıştır?
  - I. Bir sayfada birden fazla RequiredFieldValidator kontrolü kullanılabilir.
  - II. Bir sayfada birden fazla RangeValidator kontrolü kullanılabilir.
  - III. Doğrulama kontrollerinin görüntülenmesi için CompareValidator kullanılır.
  - A) Yalnız I
  - B) Yalnız II
  - C) Yalnız III
  - D) I ve II

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-3

## AMAÇ

Bir web sitesinin güvenliğini sağlayarak web formlarıyla veritabanındaki verilere erip onlar üzerinde işlemler yapabileceksiniz.

## ARAŞTIRMA

- Gündelik yaşantımızda değişik güvenlik seçeneklerine göre erişim yapıp yapılamayacağını belirleyen durumlar vardır. Örneğin, bir binanın kapalı otoparkına arabanızı park ederken otoparkın kapısı eğer o binadan iseniz otomatik açılmakta; o binadan değilseniz otoparkın kapısı açılmamaktadır. Siz de bu şekilde güvenliğin olduğu durumlardan üçer adet örnek bulunuz. Örneğinizi sınıfta paylaşacak ve öğretmeninize teslim edecek şekilde hazırlayınız.

## 3. GÜVENLİK VE ERİŞİM

Günümüzde hemen hemen pek çok web sitesinde erişime bağlı olarak ortaya çıkacak güvenlik sorunları önem arz etmektedir. Örneğin, okulunuzda “E-Okul” sistemi kullanılmaktadır. Bu sisteme idareciler, öğretmenler, öğrenciler ve veliler erişerek bir takım işlemleri gerçekleştirmektedirler. Ancak her birinin eriştiği alan farklılık arz etmektedir. Zaten öyle de olması gerekir. Çünkü bir öğretmenin eriştiği sayfalarda, derse girdiği sınıfın öğrencilerine ait notları olacaktır;bu sayfaya bir öğrencinin erişmesi büyük bir güvenlik sorunu demektir. İşte kullanıcıları gruplamak ve her bir gruba farklı erişim izinleri vermek bu işin temelinde yatan bir mantıktır.

ASP.NET, sayfalarda oluşacak bu tip erişimler için *Login* kontrollerini sizlerin hizmetine sunmuştur. Form tabanlı güvenliğin temelini teşkil eden bu kontroller sizleri değişik kullanıcı gruplarına bazı kurallar tanımlamak suretiyle, erişimlerini ayarlama işini yapmanıza yardımcı olmaktadır.

### 3.1. Form Tabanlı Güvenliği Yönetme

Form tabanlı güvenlik, bir kullanıcının kimliğini, kullanıcıya bir ID ve bir parola soran oturum açma formu görüntüleyerek doğrulamanızı sağlar. Bu doğrulama sonucu kullanıcı sisteme erişim gerçekleştirirse bile kendine tanınan haklar çerçevesinde diğer sayfalara erişir.

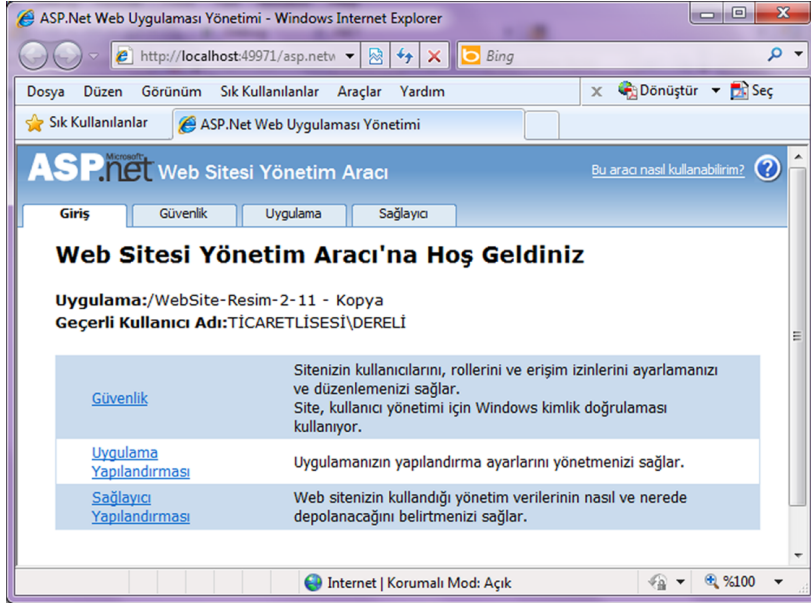
ASP.NET form tabanlı güvenliğini kullanabilmek için iki yöntem mevcuttur. Bunlardan ilki *web.config* yapılandırma dosyasına kodlamaları elle yapmak; ikincisi ise *ASP.NET Web*

*Site Yönetim Aracı* görsel ara yüzünü kullanmaktır. Sizler ikinci yöntemi kullanarak kullanıcıların erişim haklarını düzenleyeceksiniz.

Örnek: Bir okulun web sayfası yapım aşamasındadır. Bu okulun web sayfasında idareci, öğretmen ve öğrencilerin kullanabilecekleri sayfalar farklı olarak plânlanmaktadır. Sisteme giriş yapan kullanıcı, hangi grupta (idareci, öğretmen, öğrenci) ise o grubun sayfasına yönlendirilmek istenmektedir. Buna göre kullanıcı gruplamasını yaparak her bir gruba ait sayfayı oluşturunuz.

Burada tam olarak yapılmak istenen form tabanlı güvenlik işlemidir. Bunun için ASP.NET'in kullanıcı kontrollerini ve kendi web yapılandırıcısını kullanacaksınız.

- İlk olarak yeni bir web sitesi oluşturunuz.
- Bu web sitesine Idareciler, Öğretmenler, Öğrenciler olmak üzere üç adet klasör oluşturunuz ( Proje penceresindeki -solution Explorer- proje adı üzerine farenin sağ tuşuna tıklayınız. Gelen menüden Folder seçeneğini tıklayınız. Klasörün adını değiştiriniz.). Klasörleri oluşturmanızın amacı; her bir grubun sayfalarını kendi klasörlerine koymaktır. Kullanıcıların erişimlerini de klasörlere yönlendirmek suretiyle yapacaksınız.
- Website > ASP.NET Configuration menü seçeneğini tıklayarak yapılandırma işlemlerini yapacaksınız. Bu işlem sonucunda karşınıza aşağıdaki Resim 3.1'de görünen ekran gelecektir.



**Resim 3.1: ASP.NET Web Site Yönetim Aracı giriş ekranı**

- Karşınıza gelen Resim 3.1'deki ekranda yer alan Güvenlik sekmesini ya da linkini tıklayınız. Karşınıza aşağıdaki Resim 3.2'de görünen ekran gelecektir.



**Resim 3.2: ASP.NET Web Site Yönetim Aracı Güvenlik ayarlamaları**

- Resim 3.2’deki ekranda yer alan Kullanıcılar bölümündeki “Kimlik doğrulama türünü seçin” linkini tıklayınız. Bu ekranda karşınıza, kullanıcıların web sitesine nasıl erişeceklerini soran bir ayarlama gelmektedir. Bunun için iki seçenek sunulmaktadır: *İnternette*, *Yerel Bir Ağdan*. Siz internet seçeneğini seçiniz. Çünkü kullanıcılarımız web sayfasına internet üzerinden erişeceklerdir. Daha sonra “Bitti” butonuna tıklayınız ve işleminizi onaylayınız. Bu işleminden sonra tekrar güvenlik sayfasına dönmüş olmalısınız.
- Artık karşınıza gelen güvenlik sayfasındaki *Kullanıcılar* bölümü aşağıdaki Resim 3.3’te görüldüğü gibi değişmiş olmalıdır.



**Resim 3.3: ASP.NET Web Sitesi Yönetim Aracı Kullanıcılar bölümü**

- Şimdiye kullanıcıları oluşturacaksınız. Bunun için Resim 3.3’teki “Kullanıcı Oluştur” linkini tıklayınız. Karşınıza aşağıdaki Resim 3.4’te görünen kullanıcı oluşturma ekranı gelecektir.

Kullanıcı Oluştur

Yeni Hesabınız için Kaydolun

Kullanıcı Adı:

Parola:

Parolayı Onayla:

E-posta:

Güvenlik Sorusu:

Güvenlik Yanıtı:

Kullanıcı Oluştur

**Resim 3.4: Kullanıcı oluşturma ekranı**

Bu ekranı kullanarak aşağıdaki bilgilere göre üç adet kullanıcı oluşturunuz. Bunun için bilgileri girdikten sonra “Kullanıcı Oluştur” butonunu tıklamanız yeterlidir.

Kullanıcı Adı: idareci1  
Parola: 123456!  
E-posta: idareci1@okul.k12.tr  
Güvenlik Sorusu: Unvanım nedir?  
Güvenlik Yanıtı: idareci

Kullanıcı Adı: ogretmen1  
Parola: 123456!  
E-posta: ogretmen1@okul.k12.tr  
Güvenlik Sorusu: Unvanım nedir?  
Güvenlik Yanıtı: öğretmen

Kullanıcı Adı: ogrenci1  
Parola: 123456!  
E-posta: ogrenci1@okul.k12.tr  
Güvenlik Sorusu: Unvanım nedir?  
Güvenlik Yanıtı: öğrenci

Bu kullanıcıları oluşturduktan sonra tekrar “Güvenlik” sekmesine tıklayarak güvenlik sayfasına dönünüz. Artık Kullanıcılar bölümünde var olan kullanıcıların üç olduğunu göreceksiniz.

- Şimdi ise Resim 3.2’deki ekranda görünen *Roller* bölümündeki “Rolleri Etkinleştir” linkini tıklayarak aşağıdaki Resim 3.5’teki görüntüyü elde ediniz.



**Resim 3. 5: Rollerin etkinleştirilmesi**

Roller, kullanıcıların web sitesinde nasıl davranacaklarını belirleyen yapılardır. Windows temelli bir güvenlik sistemidir. Sizin oluşturmuş olduğunuz web sitesi için şu şekilde belirtilebilir. İdarecilerin İdareciler klasöründeki sayfalara; öğretmenlerin Öğretmenler klasöründeki sayfalara; öğrencilerin Öğrenciler klasöründeki sayfalara erişim yapması ve diğer klasörlerdeki sayfalara erişim yapmaması için kullanacağımız yapı *Roller* yapısıdır. Resim 3.5'teki "Roller Oluştur veya Yönet" linkini kullanarak istediğimiz rolleri oluşturacağız.

- Karşınıza gelen rol oluşturma ekranını kullanarak, *idareci*, *ogretmen* ve *ogrenci* rollerini oluşturunuz (Resim 3.6) .

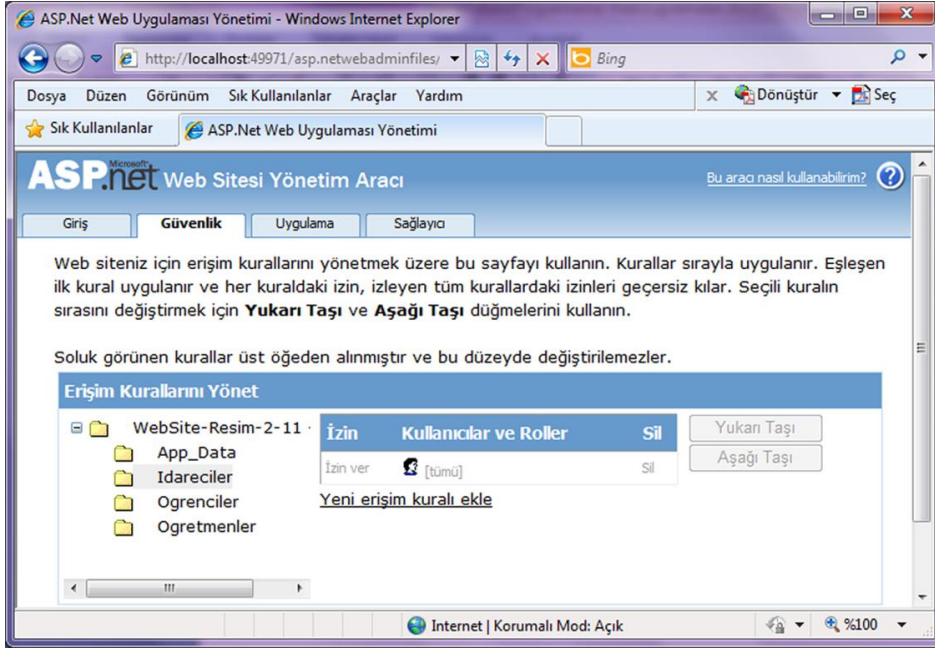


**Resim 3.6: Rol oluşturma ekranı**

Rol oluşturma işleminden sonra "Güvenlik" sekmesine tıklayarak Resim 3.2'deki ekrana dönünüz. Ancak artık üç adet rolünüz olmuştur.

- Bundan sonraki işimiz Resim 3.2'deki ekrandaki "Erişim Kuralları" bölümüyle olacaktır. Artık hangi rolün hangi klasörlere erişeceğini buradan gerçekleştireceğiz. "Erişim kuralları yönet" linkine tıklayarak bu sayfaya geçiniz.
- "Erişim kuralları yönet" linkine tıkladığımızda aşağıdaki Resim 3.7'deki ekran karşınıza gelecektir. Bu ekranda sol tarafta proje klasörleri, sağ tarafta ise bu klasörlere erişim izni olan veya erişim yasağı olan rol isimleri görünmektedir. Şu anda herhangi bir erişim kuralı mevcut olmadığı için roller veya kullanıcılar şu anda seçili olan "İdareciler" klasörüne erişebilirler.

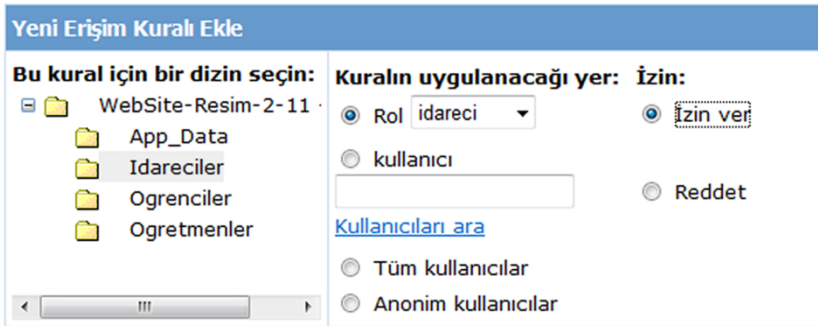




**Resim 3.7: Erişim kuralı yönet ekranı**

Şimdi “Idareciler” klasörüne erişim izni olan veya erişim yasağı olan kullanıcıları ayarlayacağız. Yapacağımız şey şudur: “Idareciler” klasörüne idareci rolüne sahip kullanıcılar erişebilecek; diğerleri erişemeyecektir. Sol taraftan “Idareciler” klasörünü seçiniz. Sağ taraftaki “Yeni erişim kuralı ekle” linkine tıklayınız.

- Bu linke tıkladığınızda aşağıdaki Resim 3.8’deki ekran karşınıza gelecektir. Bu ekranı kullanarak “Idareciler” klasörüne erişim izinleri düzenleyeceksiniz. Bunun için sol tarafta “Idareciler” klasörü seçili iken sağ taraftan *Rol* bölümünden “idareci”; *İzin* bölümünden ise *İzin ver* seçeneğini seçip “Tamam” butonuna tıklayınız.



**Resim 3.8: Yeni erişim kuralı ekleme ekranı**

Aynı şekilde “ogretmen” ve “ogrenci” rolleri ile “Anonim Kullanıcılar” seçeneğini seçip *İzin* bölümünden *Reddet* seçeneğini seçtikten sonra “Tamam” butonuna tıklayınız. Böylece “Idareciler” klasörüne erişim izinlerini düzenlemiş olduk. *Güvenlik* sekmesindeki

“Erişim İzinlerini Yönet” linkine tıklarsanız aşağıdaki Resim 3.9’da görünen ekranla karşılaşmış olursunuz.



**Resim 3.9: Erişim izinleri düzenlenmiş “Idareciler” klasörü**

Aynı işlemleri “Ogrenciler” klasörü için gerçekleştiriniz ve aşağıdaki Resim 3.10’da görünen ekranı elde ediniz.



**Resim 3.10: Erişim izinleri düzenlenmiş “Ogrenciler” klasörü**



**Resim 3.11: Erişim izinleri düzenlenmiş “Ogretmenler” klasörü**

- Şimdide hangi kullanıcının hangi rolü yerine getireceğini belirleyeceksiniz. Bunun için “Güvenlik” sekmesine tıklayınız. Karşınıza gelen ekranda *Kullanıcılar* bölümündeki “Kullanıcıları Yönet” linkini tıklayınız. Karşınıza gelen ekranda kullanıcıların bir listesini göreceksiniz. Bu kullanıcı listesindeki “idareci1” kullanıcılarındaki “Rolleri Düzenle” linkini tıklayarak aşağıdaki Resim 3.12’de görünen ekranı elde ediniz.

| Etkin                               | Kullanıcı adı |                                   |                               |                                 | Roller   |
|-------------------------------------|---------------|-----------------------------------|-------------------------------|---------------------------------|--|
| <input checked="" type="checkbox"/> | idareci1      | <a href="#">Kullanıcı düzenle</a> | <a href="#">Kullanıcı sil</a> | <a href="#">Rolleri düzenle</a> | " idareci1 " kullanıcısının ekleneceği roller: |
| <input checked="" type="checkbox"/> | ogrenci1      | <a href="#">Kullanıcı düzenle</a> | <a href="#">Kullanıcı sil</a> | <a href="#">Rolleri düzenle</a> | <input checked="" type="checkbox"/> idareci    |
| <input checked="" type="checkbox"/> | ogretmen1     | <a href="#">Kullanıcı düzenle</a> | <a href="#">Kullanıcı sil</a> | <a href="#">Rolleri düzenle</a> | <input type="checkbox"/> ogrenci               |
|                                     |               |                                   |                               |                                 | <input type="checkbox"/> ogretmen              |

**Resim 3.12: Kullanıcı düzenleme ekranı**

Bu ekranda gördüğümüz gibi “idareci1” kullanıcısı idareci rolüne atanmış olmaktadır. Aynı işlemi “ogrenci1” ve “ogretmen1” kullanıcısı içinde gerçekleştiriniz. Böylece *ASP.NET Web Site Yönetim Aracındaki* işinizi bitirmiş oldunuz. Bu pencereyi kapatınız ve projenize dönünüz.

- Şu ana kadarki yaptıklarınız için projenizin *App\_Data* klasöründe *ASPNETDB.MDF* isminde bir veri tabanı oluşturulur ve buraya kaydedilir. Ayrıca her bir klasöre erişimi kontrol etmek için her klasör altında bir *web.config* dosyası oluşturulmuştur.
- Şimdide proje penceresinden (Solution Explorer) her bir klasöre bir adet *aspx* sayfası oluşturun. Bu *aspx* sayfalarına hangi gruba aitse o grubu anımsatacak bir metin ekleyin. Örneğin “Idareciler” klasörü altındaki *Default.aspx* sayfasına şu metni ekleyin.

*İdareciler grubu sayfası*

*İdareciler sayfasına hoş geldiniz...*

- Daha sonra projenize *Login.aspx* adında yeni bir web sayfası ekleyin. Bu sayfa, ASP.NET sayfalarında kullanıcıların, *kullanıcı adı* ve *parolalarını* kullanarak sisteme giriş yaptıkları sayfadır.
- Oluşturmuş olduğunuz *Login.aspx* sayfasına araç kutusundaki (ToolBox) *Login* sekmesinde yer alan *Login* kontrolünü ekleyin. Sayfanız aşağıdaki Resim 3.13’ deki gibi görünecektir.

Log In

User Name:  \*

Password:  \*

Remember me next time.

**Resim 3.13: Login kontrolü**

Login kontrolündeki özellikleri aşağıdaki gibi değiştiriniz.

LoginButtonText: Giriş

PasswordLabelText: Şifre

RememberMeText: Bir sonraki oturumda beni hatırla

TitleText: KULLANICI GİRİŞ FORMU

UserNameLabelText: Kullanıcı Adı

- “İdareciler” klasöründeki Default.aspx sayfasını tarayıcıda açın. Henüz bir yetkiniz olmadığı için otomatik olarak web sitenizdeki Login.aspx sayfasına yönlendirileceksiniz. Bu sayfada kullanıcı adı bölümüne “idareci1”, şifre bölümüne “123456!” yazarak “Giriş” butonuna tıklayınız. Bu kullanıcı bu sayfaya girmeye yetkili olduğu için sayfa açılacaktır. Eğer öğrenci1 kullanıcıyla giriş yapsaydınız idareciler sayfasını görüntüleyemeyecektiniz.

Sizde aynı uygulamayı adım adım yaparak aynı sonuçları almaya çalışın.

### 3.2. Veriyi Görüntülemek ve Sorgulamak

Günümüzde artık veriler genellikle veritabanı sistemlerinde kayıtlı olmaktadır. Size kalansa bu veritabanındaki veriyi sorgulamak ve bu sorgu sonucu dönen kayıtları izlemektir. ASP.NET, bir veri tabanındaki veriler üzerinde işlem yapmak için birtakım kontrollerle birlikte gelmiştir. Bu kontroller veri (data) kontrolleri olup DataList, GridView, DetailsView, FormView, Repeater şeklinde sıralayabiliriz. Bunlardan en çok kullanılanı ise GridView kontrolüdür.

GridView kontrolü, veri görüntüleme ve yönetmek için tablo yapısı biçiminde bir kontroldür. Aynı zamanda binlerce veriyi aynı anda çekip bu verileri sayfalar hâlinde de görüntüleme yeteneğine sahiptir. GridView kontrolü, verileri tablo içerisinde salt okunur bir şekilde görüntülenmektedir. Ancak istenirse düzenleme modu ile de veri düzenlenebilmektedir.

Aşağıda Access veri tabanındaki bir tablodan veriler, adım adım sorgulanarak görüntülenmektedir. Her bir adımı dikkatlice inceleyip uygulayınız.

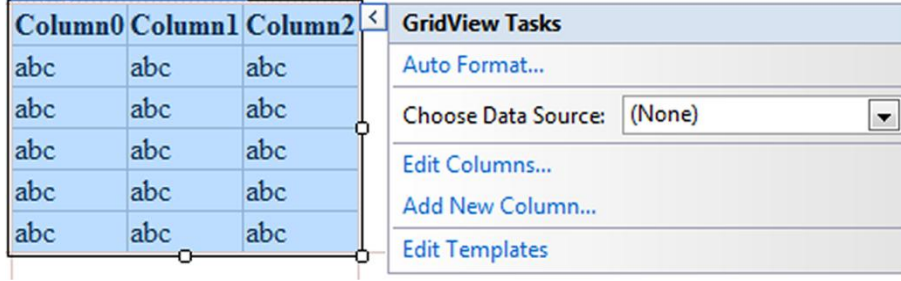
- Örnek olarak aşağıdaki Resim 3.14’te görünen tabloyu Access veri tabanında oluşturunuz. Tablo oluştururken alan isimlerini ve veri türlerini aynı şekilde gerçekleştiriniz. Bu uygulamada anlatılan Access veritabanı dosyasının adı “okul.mdb”; tablonun adı ise “ogrenciler” ‘ dir.
- Daha sonra oluşturduğunuz tabloya biraz veri giriniz.



| Alan Adı | Veri Türü     |
|----------|---------------|
| OkulNo   | Otomatik Sayı |
| Ad       | Metin         |
| Soyad    | Metin         |
| Cinsiyet | Evet/Hayır    |
| DTarihi  | Tarih/Saat    |
| DYeri    | Metin         |
| BabaAdi  | Metin         |
| AnaAdi   | Metin         |
| Sinif    | Metin         |
| Alan     | Metin         |

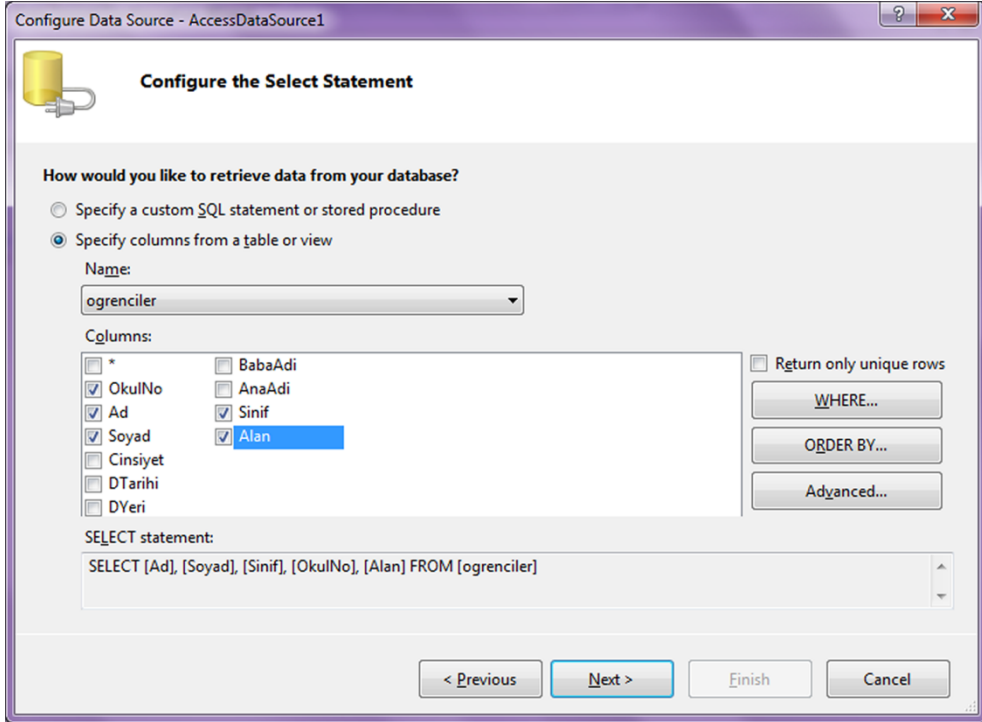
Resim 3.14: Örnek veritabanı tablo alanları ve veri türleri

- Şimdi Nesne Tabanlı Programlama Yazılımını kullanarak yeni bir web sitesi oluşturunuz.
- Oluşturmuş olduğunuz Access veri tabanı dosyasını web sitenizin *App\_Data* klasörüne yerleştiriniz (*App\_Data* klasörü üzerinde farenin sağ tuşuna tıklayın. Gelen seçeneklerden *Add Existing Item* seçeneğine tıklayın.).
- *Default.aspx* sayfasına *GridView* kontrolünü sürükleyin.



**Resim 3.15: GridView kontrolü ve akıllı görev listesi**

- *GridView* kontrolünü sayfanıza eklediğinizde akıllı görev bölmesi de beraberinde açılır. *Choose Data Source* seçeneğinden *New Data Source* seçeneğini seçiniz. Karşınıza gelen pencereden *AccessDataSource* seçeneğini seçin ve “OK” butonuna tıklayın.
- Karşınıza database seçme (Choose a Database) ekranı gelmiş olmalıdır. “Browse” butonu yardımıyla web sitenizdeki “okul.mdb” veri tabanı dosyasını seçiniz ve “Next” butonuna basınız.
- Karşınıza aşağıdaki Resim 3.16’da görünen veri sorgulama ekranı gelmiş olmalıdır. Bu ekranda veri tabanında bir tane tablo olduğu için (ogrenciler) o tablo yüklü olarak gelir. Eğer birden fazla tablo olursa bu defa açılır kutudan istediğiniz tabloyu seçebilirsiniz. Bu ekrandaki alanlardan hangini listelemek istiyorsanız onları seçiniz. (Bu uygulamada Resim 3.16’da da gördüğümüz gibi; OkulNo, Ad, Soyad, Sınıf ve Alan seçilmiştir.)



**Resim 3.16: Görüntülenecek veri tabanı alanlarının seçimi**

“Next” butonuna tıklayarak sonraki Test Query ekranına geçiniz.

- Text Query ekranında sorgunuzun düzgün bir şekilde çalışıp çalışmadığını deneyebilirsiniz. “Test Query” butonuna tıkladığınızda veri tabanından çekilecek veriler listelenecektir. Bu da sorgunuzun düzgün çalıştığını göstermektedir. Siz de sorgunuzu test edin ve çalışıyorsa “Finish” butonuna tıklayarak *GridView* kontrolüne veri bağlama işini sonlandırın.
- Default.aspx sayfasına tekrar dönün. Sayfanızı tarayıcıda test etmek üzere çalıştırın. Karşınıza veri listesi aşağıdaki Resim 3.17’deki görüntüye benzer bir şekilde gelecektir.

| OkulNo | Ad     | Soyad      | Sinif | Alan                  |
|--------|--------|------------|-------|-----------------------|
| 1      | Özkan  | Özçelik    | 11-A  | Bilişim Teknolojileri |
| 2      | Gizem  | Güleç      | 11-A  | Bilişim Teknolojileri |
| 3      | Mehmet | Ersandıkçı | 11-A  | Bilişim Teknolojileri |
| 4      | Ceyhan | Ayaydın    | 11-A  | Bilişim Teknolojileri |
| 5      | Burçin | Abacı      | 11-A  | Bilişim Teknolojileri |
| 6      | Gurur  | Kaya       | 11-A  | Bilişim Teknolojileri |
| 7      | Simge  | Atar       | 11-A  | Bilişim Teknolojileri |
| 8      | Gökmen | Güngüder   | 12-A  | Bilişim Teknolojileri |
| 9      | Kerem  | Cinel      | 12-A  | Bilişim Teknolojileri |

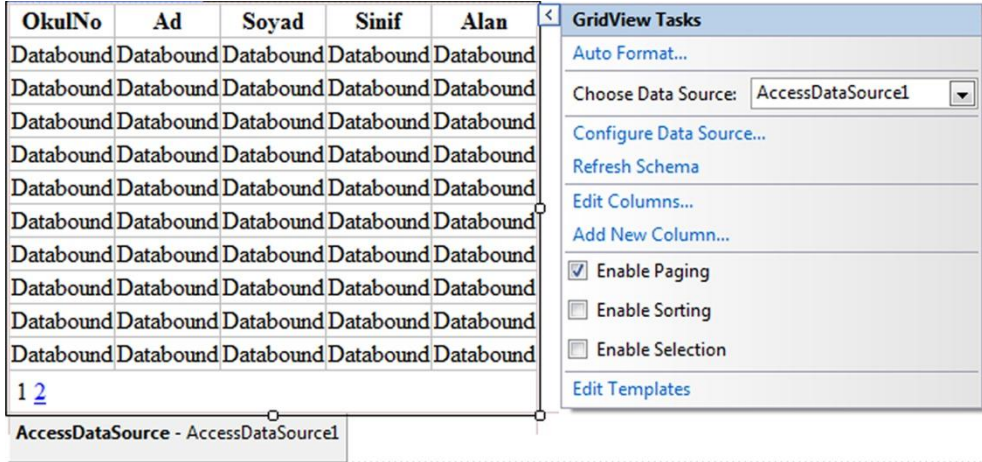
Resim 3.17: GridView kontrolünün çalışması

### 3.3. Veriyi Sayfalarda Görüntüleme

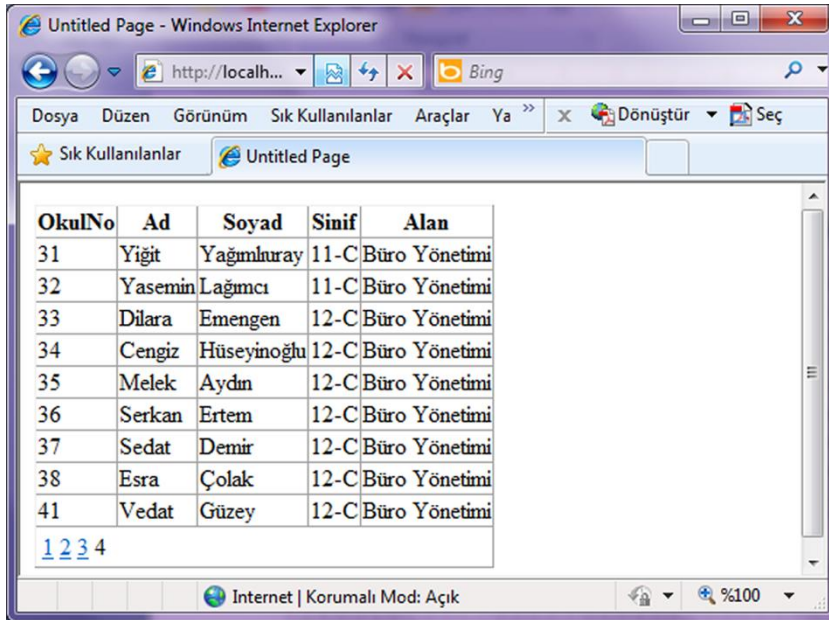
Veri tabanından veri sorgulamanın ve sayfalarda görüntülemenin ne kadar kolay olduğunu bir önceki konuda anladınız. Sorguladığımız veri, muhtemelen birkaç satırdan ibaretti. Ancak daha büyük web sitelerinde yüzlerce hatta binlerce veri çekilmektedir. Örneğin, mail adresinizi düşünün. Mail adresinize gelen mailler, belli bir sayıya ulaştığında sayfalara bölünerek görüntülenir. Bu işlem, hem okunabilirliği kolaylaştırır hem de veriyi görüntüleme işini daha hızlı bir hale getirir.

Aşağıda bir önceki konu başlığında (3.2.Veriyi Görüntülemek ve Sorgulamak) anlatılan uygulamadaki verilerin, sayfalara nasıl bölündüğünü anlatan bir uygulama mevcuttur.

Bir önceki uygulamada oluşturmuş olduğunuz Default.aspx sayfasına geliniz. GridView kontrolünün aşağıdaki Resim 3.18’te görüldüğü gibi akıllı görev listesini açınız. Bu listedeki Enabled Paging seçeneğini işaretleyiniz (Resim 3.19). Sayfanızı test ettiğinizde 10’ar adet satırlar halinde verileri görüntüleyecektir. Ancak siz isterseniz bu sayıyı GridView kontrolünün özelliklerinden değiştirebilirsiniz.



Resim 3.18: GridView kontrolünün akıllı görev listesi



Resim 3.19: GridView kontrolündeki verinin sayfalara bölünmesi

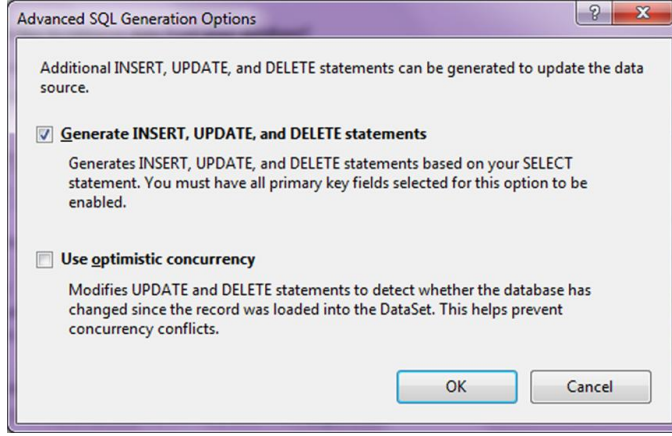
### 3.4. Veriyi Düzenleme

Veri tabanındaki veriler sadece görüntülenmek için sorgulanmazlar. Aynı zamanda bu verilerden bazıları değiştirilebilir veya silinebilir. Bu gibi durumlarda da *GridView* kontrolü etkin bir şekilde kullanılabilir. GridView kontrolüyle veri düzenlemek için aşağıdaki uygulamayı adım adım gerçekleştiriniz.

Nesne Tabanlı Programlama Yazılımı ile yeni bir web uygulaması oluşturunuz. Aynı şekilde “okul.mdb” olarak oluşturduğunuz veri tabanını, web sitenizin *App\_Data* klasörüne taşıyınız. Default.aspx sayfasına GridView kontrolünü sürükleyin ve “okul.mdb” veri



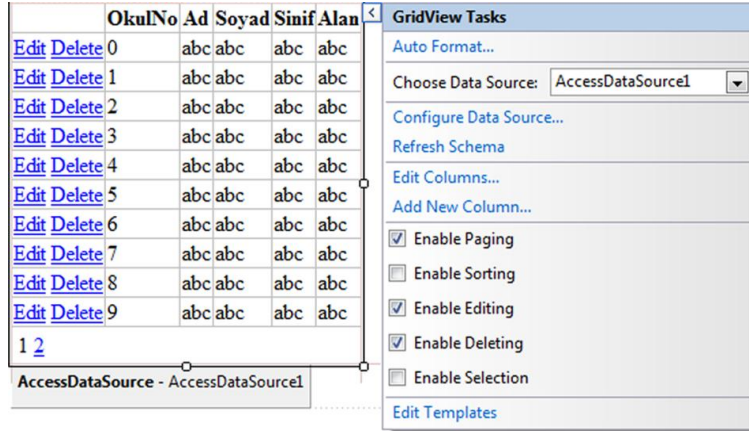
tabanını bu kontrole bağlayın. Yalnız veri bağlama işlemini yaparken Resim 3.16’da görünen *Configure the Select Statement* ekranına geldiğinizde, bu ekrandaki “Advanced” butonuna tıklayın. Karşımıza aşağıdaki Resim 3.20’de görünen ekran gelecektir.



**Resim 3.20: Veriyi düzenlemek için gerekli ayarlama**

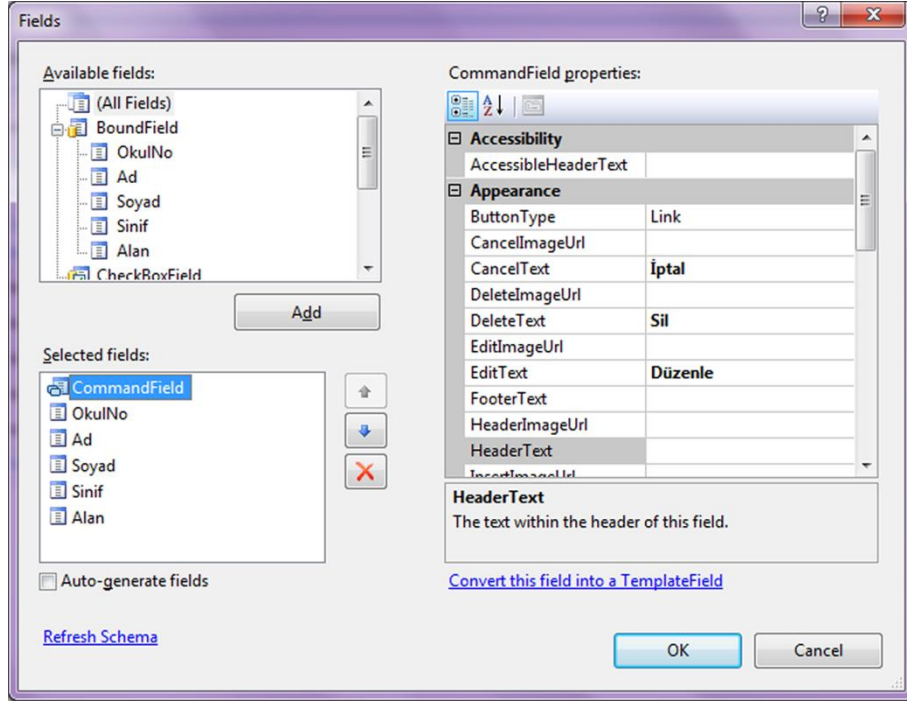
Bu ekrandaki *Generate INSERT, UPDATE and DELETE statements* ifadesini işaretleyiniz. “OK” butonu ile işinizi sonlandırınız. Böylece verinin bağlı olduğu kontrolde ekleme, güncelleme ve silme işlemleri yapabilirsiniz.

Sorgunuzu test edip çalıştırdıktan sonra Default.aspx sayfasına tekrar dönün. GridView kontrolünün akıllı görev listesini aşağıdaki Resim 3.21’deki gibi görüntüleyin.



**Resim 3.21: GridView kontrolü akıllı görev listesi**

Bu ekranda da gördüğümüz gibi verileri düzenlemek (değiştirmek) için *Enable Editing*, verileri veri tabanından silmek içinde *Enable Deleting* seçeneğini seçiniz. Her bir seçeneği seçtiğinizde *GridView* kontrolünün en başında “Edit” ve “Delete” linklerinin görüldüğünü fark edeceksiniz. İsterseniz bu linkleri butona, yazıları da istediğiniz gibi değiştirebilirsiniz. Bunun için akıllı görev listesinde yer alan (Resim 3.21) *Edit Columns* seçeneğini tıklayınız. Karşımıza aşağıdaki (Resim 3.22) ekran gelecektir.



**Resim 3.22: GridView sütunu düzenleme ekranı**

Bu ekran toplamda üç bölümden oluşmaktadır. Birinci bölüm (Available field); veri tabanından çekilen alanların tamamıyla birlikte GridView kontrolünde kullanılabilecek diğer alanların bulunduğu bölümdür.

İkinci bölüm (Selected fields); o anda GridView kontrolünde seçili olarak görüntülenen alanları gösterir. Birinci bölümden seçilen alanlar buraya eklenir. İstenirse buradaki bazı alanları silerek görüntülenmesini engelleyebilirsiniz.

Üçüncü bölüm (field properties) ise seçili olan alanın özelliklerinin ayarlandığı bölümdür. Siz “Edit” ve “Delete” isimlerini değiştirmek için ikinci bölümdeki (selected fields) CommandField alanını seçiniz. (Resim 3.22 de görüldüğü gibi) Üçüncü bölümde bu alanın özellikleri listelenecektir. Burada şu değişiklikleri yapınız;

CancelText: İptal  
DeleteText: Sil  
EditText: Düzenle  
UpdateText: Güncelle

“OK” butonuna basarak işlemi sonlandırınız. Böylece linklerin değiştiğini görebilirsiniz.

|                                | OkulNo | Ad           | Soyad      | Sinif | Alan                  |
|--------------------------------|--------|--------------|------------|-------|-----------------------|
| <a href="#">Düzenle Sil</a>    | 1      | Özkan        | Özçelik    | 11-A  | Bilişim Teknolojileri |
| <a href="#">Düzenle Sil</a>    | 2      | Gizem        | Güleç      | 11-A  | Bilişim Teknolojileri |
| <a href="#">Düzenle Sil</a>    | 3      | Mehmet       | Ersandıkçı | 11-A  | Bilişim Teknolojileri |
| <a href="#">Düzenle Sil</a>    | 4      | Ceyhan       | Ayaydın    | 11-A  | Bilişim Teknolojileri |
| <a href="#">Düzenle Sil</a>    | 5      | Burçin       | Abacı      | 11-A  | Bilişim Teknolojileri |
| <a href="#">Güncelle İptal</a> | 6      | Gurur sdfhsk | Kaya       | 11-A  | Eğilim Teknolojileri  |
| <a href="#">Düzenle Sil</a>    | 7      | Simge        | Atar       | 11-A  | Bilişim Teknolojileri |
| <a href="#">Düzenle Sil</a>    | 8      | Gökmen       | Güngüder   | 12-A  | Bilişim Teknolojileri |
| <a href="#">Düzenle Sil</a>    | 9      | Kerem        | Cinel      | 12-A  | Bilişim Teknolojileri |

**Resim 3.23: GridView kontrolüyle veri güncellemesi**

Sadece yazı düzenlemesi değil aynı zamanda satırların yüksekliği, sütunların genişliği, stil tanımlamaları, metinleri hizalanması gibi pek çok işlemi bu ekranı kullanarak gerçekleştirebilirsiniz.

### 3.5. Formlar Arasında Gezinme

Şu ana kadar geliştirdiğiniz bütün web uygulamalarında tek bir sayfa dolayısıyla da tek bir web form kullandınız. Ancak pratikte ASP.NET web uygulamaları birden fazla web formundan oluşmaktadır. Örneğin, internetten alışveriş yaptınız ve o an sepetinize bakmak istiyorsunuz. Linke tıklayıp sepetteki ürünlerin listesini başka bir sayfada görebilirsiniz veya bir ürünün detayına bakmak istediğinizde yine linke tıklarsınız ve başka bir sayfada o ürünün detayını incelersiniz. Bunun gibi örnekler çoğaltılabilir.

Şimdiki yapacağınız uygulama buna benzer bir uygulamadır. Resim 3.19’da görünen listedeki her bir satırın sonuna “Detay” linki koyup ilgili satırdaki kayıtla ilgili bütün bilgileri başka bir web formda görebileceksiniz.

Öncelikle Nesne Tabanlı Programlama Yazılımında yeni bir web form oluşturun. *App\_Data* klasörüne “okul.mdb” veri tabanı dosyamızı taşıyın. Default.aspx sayfasına bir GridView kontrolü ekleyin ve öğrenciler tablosundaki verilerden “OkulNo”, “Ad” ve “Soyad” alanlarını listeleyin (Bu işlemi Resim 3.16’deki ekranda OkulNo, Ad ve Soyad alanlarını seçerek yapabilirsiniz.).

GridView kontrolüne veriyi bağladıktan sonra bu kontrolün akıllı görev listesinden “Edit Columns” seçeneğini tıklayarak Resim 3.22’de görünen ekranı açınız. Bu resimdeki geçerli alanlar (Available fields) bölümünden *HyperLinkField* seçeneğini seçerek “Add” butonuna tıklayınız. Böylece *HyperLinkField* ögesini seçili alanlar (Selected fields) bölümüne almış olursunuz.

Yine Resim 3.22’de yer alan ekrandaki seçili alanlar (Selected field) bölümünde yer alan HyperLinkField ögesini seçerek bu ögenin özelliklerinin özellikler (properties) bölümünde görüntülenmesini sağlayın.

HyperLinkFields ögesinin özelliklerini aşağıdaki şekilde düzenleyin.

Text: Detay >> (her bir satırda görünecek metin)

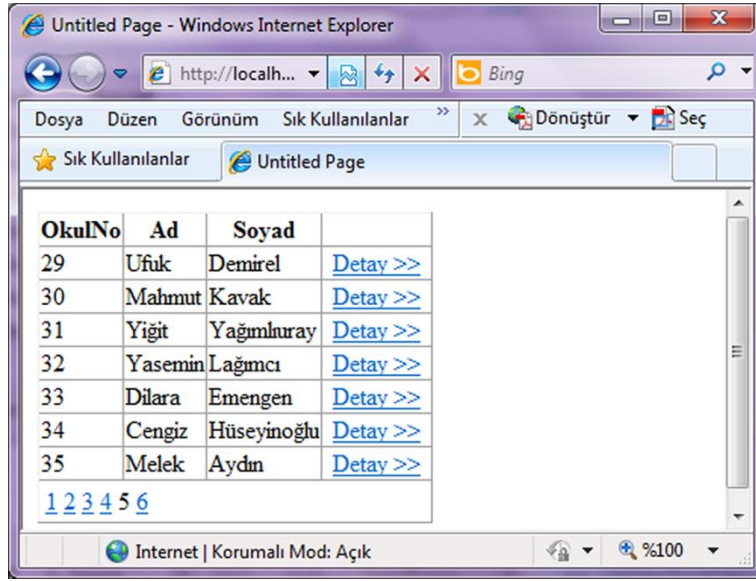
DataNavigateUrlFields: OkulNo

DataNavigateUrlFormat: ~/Detay.aspx?OgrenciNo={0}

ItemStyle > Width: 70px (sütunun genişliği)

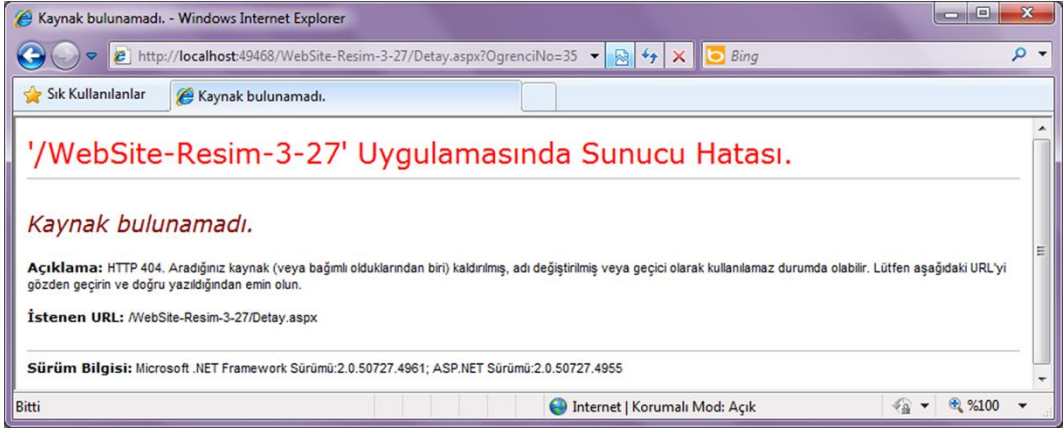
ItemStyle > HorizontalAlign: Center (hücredeki verinin hizalaması)

Bu özelliklerde *Detay.aspx* sayfasına bir link verilmiştir. Yalnız bu sayfaya giderken beraberinde *OkulNo* verisini de götürmektedir. Ayrıca *OkulNo* verisinin adını o sayfaya *OgrenciNo* olarak taşımaktadır. dolayısıyla *Detay.aspx* sayfasına dışarıdan gelen veri *OgrenciNo* ‘dur. Bu ayarlamalar neticesinde Default.aspx sayfanız aşağıdaki Resim 3.24 de ki gibi görünüyordur.



**Resim 3.24: GridView kontrolünde HyperLinkField ögesi**

Resim 3.24’te yer alan “Detay >>” linkine tıkladığınızda *Detay.aspx* sayfasına *OgrenciNo* verisiyle birlikte gidildiğini göreceksiniz (Resim 3.25).



**Resim 3.25: Diğer bir web forma yönlendirme**

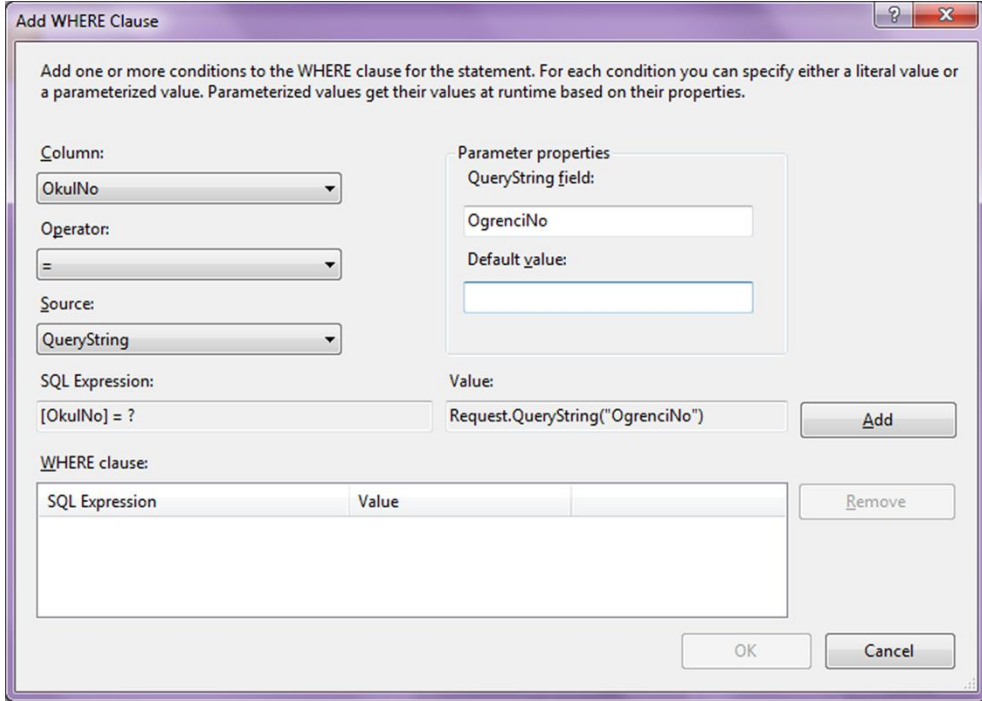
Resim 3.25'te yer alan görüntüdeki adres çubuğuna baktığınızda *OgrenciNo* bilgisinde beraberinde geldiğini göreceksiniz. Şimdi de aşağıdaki adımları uygulayarak *Detay.aspx* sayfasını tasarlayacaksınız. Bu sayfada ilgili kayda ait bilgileri görüntülemek için *FormView* kontrolünü kullanacaksınız.

Nesne Tabanlı Programlama Yazılımındaki proje penceresinden (Solution Explorer) *Detay.aspx* sayfasını oluşturun. Bu sayfaya araç kutusundaki (ToolBox) *Data* sekmesinden *FormView* kontrolünü ekleyin. *GridView* kontrolünde yaptığınız gibi “okul.mdb” veri tabanındaki “ogrenci” tablosunu *FormView* kontrolüne tek bir değişiklikle bağlayın. Bu değişikliği Resim 3.16'daki ekranda gerçekleştireceksiniz. Öncelikle tüm sütunları görüntüleyeceğiniz için hepsini seçili hale getirin.

Daha sonra *WHERE* butonuna tıklayarak Resim 3.26'da yer alan ekrandaki değişiklikleri yapınız. Bu değişiklik veri sorgulamada bir şart söz konusu olduğunda yapılacak bir işlemdir. Siz, burada aslında bir şart ortaya koyuyorsunuz. Nedir bu şart? İsteddiğiniz kayıtlarla alakalı (Okul No) bilgileri sorguluyorsunuz. Resim 3.26'daki ayarlamalar bu manada yapılan ayarlamalardır. Bu ayarlamaları;

Columns: OkulNo (*verileri hangi alana göre kıyaslama yaparak sorguladığınızı*)  
Operator: = (*koşulun operatörü*)  
Source: QueryString (*kıyaslama verisini adres satırından aldığınızı*)  
QueryStringField: OgrenciNo (*adres satırından gelen veriyi*)

şeklinde gerçekleştiriniz.



**Resim 3.26: Veri sorgulamada şart kullanımı**

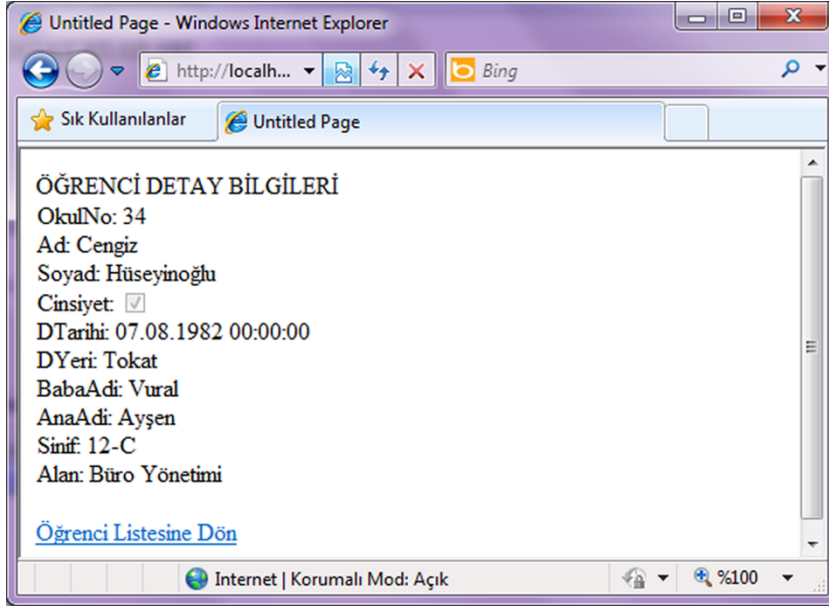
Son olarak “Add” butonu ile ayarlamalarınızın geçerli olmasını sağlayın ve “OK” butonu ile pencereyi kapatın. Böylece işlemi tamamlamış oldunuz. *Detay.aspx* sayfasına aşağıdaki kontrolleri de ekleyin.

*FormView* kontrolü altına *Default.aspx* sayfasına dönmek için bir *LinkButton* kontrolü ekleyin. Bu kontrolün *Text* özelliğini “Öğrenci Listesine Dön” olarak, *PostBackUrl* özelliğini de “~/Default.aspx” olarak düzenleyin.

*FormView* kontrolünün üzerinde bir *Label* kontrolü ekleyin. Bu kontrolün *Text* özelliğini, “ÖĞRENCİ DETAY BİLGİLERİ” olarak düzenleyin.

Uygulamanızı *Default.aspx* sayfasında iken çalıştırın. İlk olarak karşınıza Resim 3.24’te görünen ekran gelecektir. Herhangi bir satırdaki “Detay >>” linkine tıkladığınızda karşınıza o satırda yer alan öğrenciye ait diğer bilgiler gelecektir (Resim 3.27).

*FormView* kontrolü içerisindeki verilere direkt olarak müdahale edemezsiniz. Eğer bu verilere müdahale etmek isterseniz (Örneğin, verilerin daha düzenli bir şekilde görünmesi için tablo içerisine yerleştirmek vb.) bu kontrolün *akıllı görev* listesinden *Edit Template* seçeneğinden gerçekleştirebilirsiniz.



**Resim 3.27: FormView ile öğrenci detay bilgileri**

## UYGULAMA FAALİYETİ

Alanlara ait öğrencilerin listelendiği listede yer alan öğrenciye tıkladığında öğrenci detaylarının gösterildiği bir web uygulaması geliştirmiz.

| İşlem Basamakları  | Öneriler  |
|--|---|
| ➤ Yeni bir web sitesi oluşturun.   | ➤ Nesne Tabanlı Programlama Yazılımı programı size yardımcı olabilir          |
| ➤ Yapacağınız web uygulamasında “okul.mdb” veri tabanını kullanarak Default.aspx sayfasında öğrencilerin alanlarının listesini oluşturun.  | ➤ Liste için GridView kontrolünü kullanabilirsiniz.                           |
| ➤ Her bir satırda “Detay” isminde bir link oluşturunuz. Bu linke tıklayınca OgrenciListesi.aspx sayfasında ilgili alanın öğrenci listesi görüntülensin.                            | ➤ Öğrenci listesi için yine GridView kontrolünü kullanabilirsiniz.            |
| ➤ OgrenciListesi.aspx sayfasındaki öğrenci listesini sayfalara bölerek görüntüleyiniz.   | ➤ GridView kontrolünün Paging modunu kullanabilirsiniz.                       |
| ➤ Web sitenizde “OgrenciDetay.aspx” adında başka bir sayfa oluşturunuz.  | ➤ File > New > File menü seçenekleri ile bu işlemi gerçekleştirebilirsiniz.   |
| ➤ OgrenciListesi.aspx sayfasından ilgili öğrenci seçildiğinde, öğrencilerin ayrıntılı diğer bilgileri OgrenciDetay.aspx sayfasında görüntülensin.                                  | ➤ GridView kontrolüne HyperLinkField ögesi ekleyerek gerçekleştirebilirsiniz. |
| ➤ OgrenciListesi.aspx sayfasında gerekli ayarlamaları yaparak seçilen kayıt silinsin veya bilgileri güncellensin.  | ➤ GridView kontrolünün düzenleme ve silme modundan faydalanabilirsiniz.       |
| ➤ Aynı uygulamayı; öğretmenlerin, öğrenci listesi ve öğrenci detay bilgilerini, idarecilerinde öğrencileri silebilecek veya bilgilerini güncelleyebilecek şekilde yeniden yapınız. | ➤ Form tabanlı güvenlik sistemini kullanabilirsiniz.                          |



## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

| Değerlendirme Ölçütleri  | Evet | Hayır |
|--|------|-------|
| 1. Oturum açma web formu oluşturabildiniz mi?  |      |       |
| 2. ASP.NET sitesi için güvenliği yapılandırabildiniz mi?                                 |      |       |
| 3. Form tabanlı güvenlik uygulaması gerçekleştirebildiniz mi?                            |      |       |
| 4. Veri tabanındaki veriyi görüntülemek için form oluşturduunuz mu?                      |      |       |
| 5. Bir web formundaki veriyi yönetilebilir bloklar hâlinde görüntüleyebildiniz mi?       |      |       |
| 6. GridView denetimini kullanarak veri tabanındaki satırları değiştirebildiniz mi?       |      |       |
| 7. GridView denetimindeki bir satırı seçerek bir web formundan diğerine geçebildiniz mi? |      |       |
| 8. Çalışma anında bir veri kaynağına GridView denetimi bağlayabildiniz mi?               |      |       |

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme”ye geçiniz.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. ASP.NET ile form tabanlı güvenlik ayarlamaları yapmak için kullanılan görsel ara yüz yazılımının adı aşağıdakilerden hangisidir?
  - A) ASP.NET Web Sitesi Yönetim Aracı
  - B) ASP.NET Web Konfigürasyon Aracı
  - C) ASP.NET Form Tabanlı Güvenlik Aracı
  - D) ASP.NET Web Yetkilendirmesi Aracı
2. ASP.NET ile form tabanlı güvenlik ayarlamaları yapmak için kullanılan görsel arayüz yazılımını açmak için aşağıdaki menü seçeneklerinden hangisi kullanılır?
  - A) Build > ASP.NET Configuration
  - B) View > ASP.NET Configuration
  - C) Web Site > ASP.NET Configuration
  - D) Tools > ASP.NET Configuration
3. Verileri, HTML tablo yapısı şeklinde görüntülemek için kullanılan Data kontrollünün adı aşağıdakilerden hangisidir?
  - A) FormView
  - B) GridView
  - C) Login
  - D) DataList
4. GridView kontrolüyle ilgili olarak aşağıdakilerden hangisi doğrudur?
  - I. Verileri düzenlemek için kullanılır.
  - II. Yeni bir kayıt eklemek için kullanılır.
  - III. Verileri sayfalara bölmek için kullanılır.
  - A) I ve II
  - B) II ve III
  - C) I ve III
  - D) I, II ve III

**Aşağıdaki cümlelerde boş bırakılan yerlere doğru sözcükleri yazınız.**

5. ASP.NET ile web tabanlı güvenlik sistemi için yapılan ayarlamalar ..... veri tabanı dosyasında tutulur.
6. ASP.NET web tabanlı güvenlik uygulamasında kullanıcı girişleri için oluşturulan sayfada ..... kontrolü kullanılmaktadır.

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-4

## AMAÇ

Bir web servis oluşturup kullanabileceksiniz.

## ARAŞTIRMA

- Günlük hayatımızda yaptığımız işlemlerin bazılarında aynı işlemleri farklı platformlar üzerinde gerçekleştiririz. Örneğin, bankamatikten yapacağımız işlemleri aynı zamanda internet bankacılığı sayesinde internet üzerinden de gerçekleştirebiliriz. Sizde buna benzer üç adet örnek veriniz. Örnekleri sınıfınızda arkadaşlarınızla paylaşacak veya öğretmeninize teslim edecek şekilde hazırlayınız.

## 4. WEB SERVİSLERİ

Hayatımıza yeni yeni girmesine rağmen web servisler kullanımı giderek artan bir yapıya sahiptir. Örneğin; artık günümüzde bankalar, müşterilerin kredi başvurularını değerlendirirken ortak bir sistem üzerinden onların daha önceden her hangi bir ödememe durumlarının olup olmadığını kontrol edebilmekteler. Aynı zamanda bankalar, müşterilerinin kredi başvurularını değerlendirirken onların adres, nüfus kayıt örnekleri gibi bilgilerini Nüfus ve Vatandaşlık İşleri'nin web sayfasından, web servisleri kullanarak alırlar.

### 4.1. Web Servisi

Web servisler, adından da anlaşılacağı üzere web üzerinden servis veren program parçacıklarıdır. Web üzerinden verilen bu servisler standart HTTP protokolü ile olmaktadır. Bu da herkesin bu protokol vasıtası ile bir web servisine ulaşabileceğini göstermektedir. Web servisleri evrensel erişimi olan bir bileşen gibi düşünülebilir. Uygun erişim haklarına sahipseniz bilgisayarınız internete bağlı olduğu sürece dünyanın herhangi bir yerinden web servislerini kullanabilirsiniz. Çünkü web servisleri, veri iletimi için standart ve kabul edilmiş bir iletişim dili olan HTTP (Web tarafından HTML sayfalarını göndermek ve almak için kullanılan iletişim kuralı) ve XML tabanlı veri biçimini kullanır. HTTP ve XML, ASP.NET dışındaki diğer programlama ortamları tarafından da kullanılabilen standartlaşmış teknolojilerdir. Bu şu demektir: Nesne Tabanlı Programlama Yazılımını kullanarak web servisleri oluşturabilirsiniz ve Java gibi tamamen farklı ortamlarda çalışan istemci uygulamaları bunları kullanabilir. Bunun tam olarak tersini de düşünebilirsiniz.

Web servisler gerek B2B (kurumlar arası işlemler için kullanılan web sistemi) gerekse de B2C (kurumların, müşterilerin işlemleri için sunduğu web sistemi) sistemlerinde yaygın

olarak kullanılan ve kullanıldığında işlemleri ziyadesiyle kolaylaştıran sistemlerdir. Örneğin, büyük bir ithalat ihracat firması ve bu firmanın çok sayıda bayisinin olduğunu düşünelim. Bu bayilerle tüketici arasındaki mal alım satımının döviz ile olduğunu kabul edelim. Bu durumda bütün bayiler arasındaki kur miktarının tutarlı olması gerekmektedir. Bu sebepten ana bayinin, bütün bayilerin kullanması için bir döviz kuru belirlemesi gerekir. Bayilerde satış işlemlerinin yapıldığı uygulama, bu ortak döviz kurunu kullanarak satış işlemi yapmalıdır. Dağıtık yapıdaki bu bayilerde kullanılan uygulamalar aynı döviz kuruna nasıl erişecektir? Sorununun cevabı web servisleri olacaktır. Ana bayi tarafından dinamik olarak belirlenen döviz kuru, bir web servisi olarak HTTP üzerinden kullanıma açılacak ve diğer bayiler de bu servisi kullanıp işlemlerine devam edecektir. Bu örnekte de görüldüğü gibi web servisleri genellikle son kullanıcıya hitap etmez. Dağıtık yapıdaki uygulamaların bir parçası olarak geliştirilirler.

## 4.2. SOAP (Simple Object Access Protocol) Rolü

SOAP (Basit Nesne Erişim Protokolü), kullanıcıların web servislerine istekler göndermek ve onlardan yanıtlar almak için kullandıkları iletişim kuralıdır. SOAP, HTTP üstünde oluşturulmuş basit bir iletişim kuralıdır. SOAP, bir kullanıcının bir XML web servisi üzerinde harekete geçirmek isteyeceği yöntemlerin adlarını belirtmek; parametrelerini ve dönüş değerlerini ve bunların türlerini tanımlamak için bir XML söz dizimi tanımlar. Bir istemci web servisini çağırmak için, bu XML söz dizimini kullanarak yöntemi ve parametrelerini belirtmesi gerekir.

### 4.2.1. Web Servisi Tanımlama Dili

Bir SOAP iletisinin gövdesi XML'dir. Bir istemci uygulaması bir web yöntemini çağırdığında web sunucu, yöntemin parametrelerini kodlamak üzere belirli bir dizi etiketi kullanmasını bekler. Bir istemci hangi etiketin ya da XML şemasının kullanılacağını nereden bilir? Yanıtı şudur: bir XML web servisinin, sorulduğunda bir tanımlamayı kendisinin sağlaması beklenir. Web servisinin cevabı, web servisi tanımlayan bir başka XML belgesidir. Bu belge Web Servisi Tanımlama (Web Service Description) olarak bilinir. Bu belge için kullanılan XML şeması standartlaştırılmıştır ve Web Servis Tanımlama Dili (Web Services Description Language - WSDL) olarak adlandırılır.

### 4.2.2. Web Servislerinin İşlevsel Olmayan Gereksinimleri

Web servislerinin dağıtılmış hizmetleri bir araya getirmek için önemli bir teknoloji haline gelmesinden kısa bir süre sonra SOAP ve HTTP'nin tek başına çözemediği sorunlar olduğu anlaşıldı. Bu sorunlar şu şekilde sıralanabilir:

Güvenlik, ilk sorun olarak görülmektedir. Aşağıdaki örnekleri inceleyiniz.

- Bir web servisi ile bir kullanıcı arasında giden SOAP iletilerinin, internette giderken içine girilmediğinden nasıl emin olabilirsiniz?
- Aldığınız SOAP iletisinin size iddia edildiği gibi bir web servisi ya da kullanıcı tarafından gönderildiğinden ya da bilgi elde etmeye çalışan bazı sahte siteler tarafından gönderilmediğinden nasıl emin olabilirsiniz?
- Bir web servisine girişi sadece belirli kullanıcılarla nasıl sınırlandırabilirsiniz?

Bu sayılanlar; ileti bütünlüğü, gizliliği ve kimlik doğrulamayla ilgili sorunlardır. İnternette yararlanılan dağıtılmış uygulamalar oluşturacaksanız bunları göz önünde bulundurmalısınız.

1990'ların başında, dağıtılmış sistemler oluşturmak için araçlar sağlayan birçok satıcı sonradan OASIS olarak bilinen bir kurum oluşturdular. Bu kurum, yukarıdaki eksiklikler ortaya çıkmaya başladıkça WS-Security olarak bilinen bir çözüm üretti. WS-Security bildirimine abone olan satıcılar, bu bildirim gerekliliklerini karşılayan kendi uygulamalarını sağladılar. Bunu genellikle çeşitli şifreleme düzenekleri ve sertifika kullanarak yaptılar.

İlke, karşımıza çıkan başka bir sorundur. WS-Security bildiri gelişmiş bir güvenlik sağlamasına rağmen geliştiriciler bunu uygulamak için hala bir kod yazma gereği duyarlar. Örneğin, bir web servisi diğerlerine göre daha zayıf olan ve kolaylıkla kırılabilen bir şifreleme kullanabilir. Bu web servisine oldukça gizli bilgi gönderen bir kullanıcı, daha yüksek seviyede güvenlik için ısrar edebilir. Bu ilkeye bir örnektir. Bunun için WS-Policy bildiri bir web servisin ilkelerini tanımlamak ve bildirmek için genel amaçlı bir model sağlar.

Yönlendirme ve adresleme, işlevsel olmayan başka bir durumdur. Örneğin, web servis isteklerinin yeniden yönlendirilmesi, yöneticinin bakım yapması için ilgili bilgisayarı kapatması gerektiğinde faydalıdır. Aksi durumda bu bilgisayara gönderilen istekler eş bir bilgisayara yeniden yönlendirilebilir. WS-Addressing bildiri, web servis isteklerini yönlendirmek için bir çerçeve tanımlamaktadır.

#### **4.2.3. Windows Communication Foundation 'ın Rolü**

Farklı makineler üzerindeki uygulamaların birbirleriyle iletişim halinde olmasını sağlayan NET Framework 3.0 altyapısını, NET Framework 2.0 mimarisinden üstün kılan ve Servis Yönelimli Mimari için uygulamalar geliştirmek amacıyla geliştirilmiş yeni bir web servistir. WCF, Windows üzerinde servis geliştirmenizi ve onu yayınlamanızı sağlayan bir sistemdir. WCF'nin temel amacı servislerin birbirleri ile iletişimini ve birlikte çalışabilmesini sağlamaktır.

Aslında ASP.NET teknolojisini kullanarak da web servis geliştirebilirsiniz. Ancak normal ASP.NET web servisleri, doğrudan çeşitli WS-Security, WS-Policy, WS-Addressing gibi bildirimleri desteklemez. Bu nedenle de web servis geliştirmek için WCF'yi kullanmanız önerilir.

#### **4.2.4. Web Servisi Oluşturma**

Bütün bu anlattıklarımız ışığında WCF (Windows Communication Foundation) yöntemini kullanarak bir web servisi oluşturup bu web servisi başka projelerimizde kullanacaksınız. Bu web servisimiz, masaüstü uygulamasından internet sayfasında kullanılan kullanıcı adı ve şifreyle internet sayfasına erişebilmeyi sağlamaktadır.

İlk adım olarak Nesne Tabanlı Programlama Yazılımını Net Framework 3.0 versiyonunda yeni bir web site oluşturunuz.

İkinci olarak proje ismine farenin sağ tuşuyla tıklayın. *Add New Item* seçeneğine tıklayıp yeni öğe oluşturma penceresini açınız. Bu pencereden *WCF Service* öğesini seçip isminide *Service1.svc* olarak belirleyiniz. *Add* butonuna tıklayınız. Proje penceresine (Solution Explorer) bakarsanız. *Service1.svc* dosyası ile *App\_Code* klasörü projemizde oluşacaktır. Bu klasör içerisine bakarsanız iki adet dosya görürsünüz. Bunlar *IService1.cs* ve *Service1.cs*'dir. Burada *IService1.cs*, WCF web servisinde *sözleşme* (contract) olarak isimlendirilir. Web serviste kullanacağınız fonksiyonların isimlerini burada belirtirsiniz. Fonksiyonların gerçekleştireceği işlevler için gerekli kodlamaları ise *Service.cs* dosyasında yapacaksınız. Kısacası web servisinizin ana yapısını bu dosyalar oluşturacaktır.

Üçüncü adım olarak web servisinizin sözleşmesi olarak nitelendirilen *IService1.cs* dosyanızı tıklayarak kod ekranında açınız. Bu ekranda *DoWork()* fonksiyonu yüklü olarak gelen fonksiyondur. Bu fonksiyonun yerine aşağıdaki kendi fonksiyonunuzu aşağıdaki Resim 4.1'deki gibi yapınız.

```
using System;
using System.Collections.Generic;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

// NOTE: If you change the interface name '
[ServiceContract]
public interface IService1
{
    [OperationContract]
    bool giris(string isim, string sifre);
}
```

**Resim 4.1: IService.cs dosyası**

Bu resimde sonucu *boolean* tip olan *giris* isimli bir fonksiyon tanımlanmıştır. Bu fonksiyonda *string* tipte iki adet parametre kullanılmıştır. Bu parametrelere göre fonksiyonun sonucu belirlenecektir.

Dördüncü adım olarak *Service1.cs* dosyasını aşağıdaki Resim 4.2'de görüldüğü gibi oluşturunuz (*Service1: IService1* yazısının üstüne gelip tıkladığınızda fonksiyon ismi otomatik olarak *IService1.cs* dosyasından getirilecektir.).

```

using System;
using System.Collections.Generic;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.Text;

// NOTE: If you change the class name "Service1"
public class Service1 : IService1
{
    #region IService1 Members

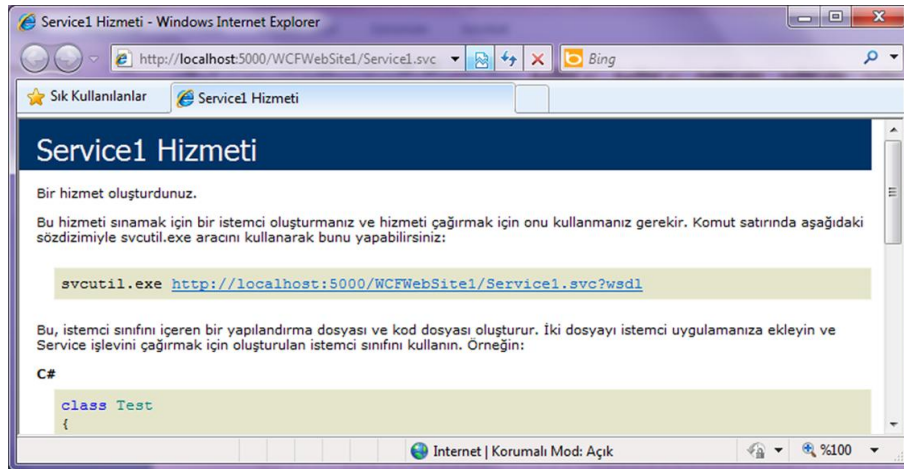
    public bool giris(string isim, string sifre)
    {
        if (isim == "meh" && sifre == "12345")
        {
            return true;
        } return false;
    }

    #endregion
}

```

**Resim 4.2: Service1.cs dosyası**

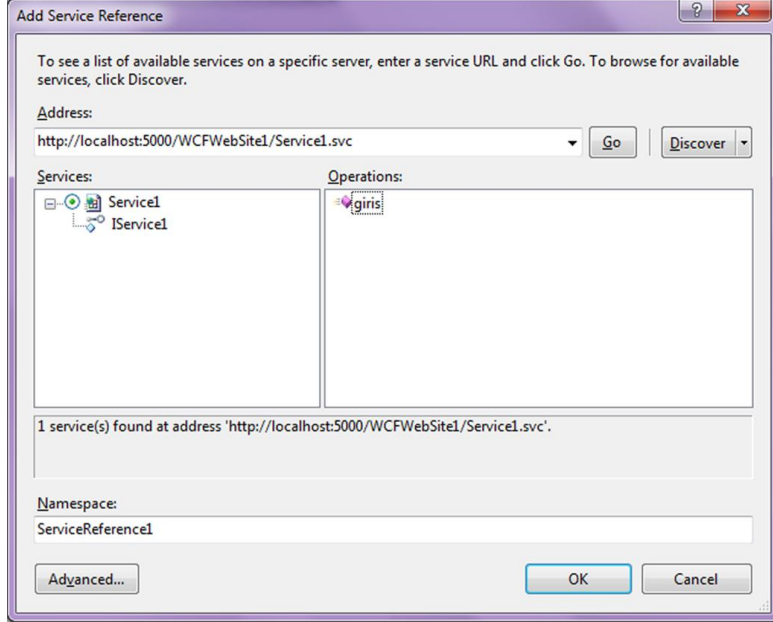
Böylece web servisimizin kodlama aşaması tamamlanmıştır. Ancak hiçbir web servis derlenmeden oluşturulamaz. Bunun için proje penceresindeki (solution explorer) Service1.svc dosyasını tarayıcıda açarak derleyiniz. Bu dosya tarayıcıda aşağıdaki Resim 4.3'teki gibi görünürse işlem tamamlanmış olacaktır. Ancak bunun için proje özelliklerinden *Use Dynamic Ports* özelliğini *False* olarak ayarlayın ve *Port number* özelliğininide 5000 olarak değiştirin.



**Resim 4.3: Web servis oluşturma penceresi**

Şimdide yapılan bu web servisi oluşturacağınız bir Windows uygulamasında kullanacaksınız. Bunun için WCF\_Uygulama isminde bir Windows projesi oluşturunuz.

Form1 açıkken Project > Add Service Reference menü seçeneklerini tıklayarak aşağıdaki Resim 4.4'te yer alan pencereye ulaşınız.



**Resim 4.4: Windows projesine web servis referans ekleme ekranı**

Yaptığınız web servisi projenize referans olarak gösterebilmeniz için Resim 4.3'teki resimde yer alan ekranın adres satırındaki bilgiyi (*http://localhost:5000/WCFWebSite1/Service1.svc*) *address* satırına giriniz ve 'Go' butonuna basınız. Belli bir süre sonra servisler ve bu servislerde bulunan tanımlı fonksiyonlar listelenecektir. Buda web servisimizle bağlantı kurulduğunu göstermektedir.



Şimdide oluşturduğunuz Windows projesindeki forma iki adet metin kutusu (textbox) ve bir adet buton ekleyiniz. Butonun tıklama olayına aşağıda yer alan kodlamayı yapınız.

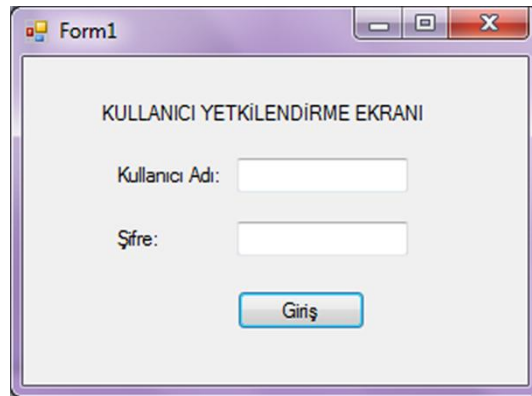
```
namespace WCF_Uygulama
{
    public partial class Form1 : Form
    {
        ServiceReferencel.Service1Client WebServiseBaglan = new
WCF_Uygulama.ServiceReferencel.Service1Client();

        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            bool sonuc = WebServiseBaglan.giris(textBox1.Text,
textBox2.Text);
            if (sonuc == true)
            {
                MessageBox.Show("Tebrikler Giriş Başarılı");
            }

            else MessageBox.Show("Hatalı giriş yaptınız, Lutfen
tekrar deneyiniz");
        }
    }
}
```

Bu kodlamada; web servisimize nasıl bağlanıldığına, web servisindeki fonksiyonun nasıl çalıştırıldığına dikkat ediniz. Programınızı çalıştırdığımızda aşağıdaki Resim 4.5'teki gibi bir ekranla karşılaşırınız.

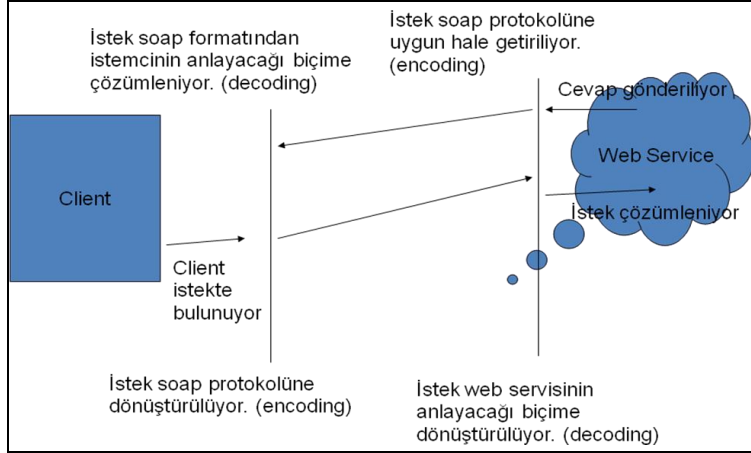


**Resim 4.5: Form görüntüsü**

Bu formda yanlış kullanıcı adını veya şifreyi yanlış yazınız ve sonuçlarını görünüz. Birde kullanıcı adını ve şifreyi doğru yazarak sonuçlarını görünüz.

### 4.3. Web Servisleri, İstemciler ve Proxy'ler

Bir web servisinin, istekleri alıp geri göndermek amacıyla düzenekler sağlamak için SOAP kullandığını gördünüz. SOAP, aktarılan veriyi biçimlendirmek için, web sunucuları ve tarayıcılar tarafından kullanılan http protokolü üstünde çalışan XML'i kullanır. XML web servislerini güçlü kılanda budur. SOAP konuşan bir istemci web servisiyle iletişim kurabilir. Peki, bir istemci nasıl SOAP konuşur? Windows uygulamanıza Add-Service Reference menüsünden servisin adresini gösterdiğinizde aslında arka plânda bir araç yardımı ile servisin tanımlama dokümanı (WSDL) talep edilmiş ve bu dokümana göre servise erişim için kullanılacak Proxy sınıfları oluşturulmuş olur. Nesne Tabanlı Programlama Yazılımı editörü sizin için bu işlemleri arka plânda gerçekleştirmektedir.



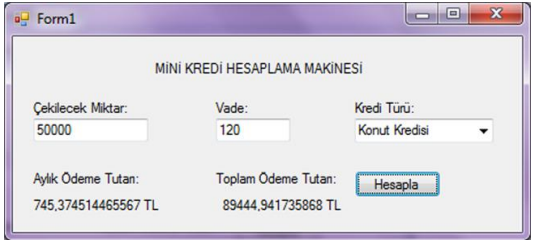
Resim 4.6 : İstemci ve Web Servis arasındaki haberleşme

Proxy, SOAP kullanmanın karmaşıklığını gizler ve web servisin yayınladığı yöntemlere dayanan basit, programatik bir ara birim sunar. İstemci uygulaması, web yöntemini, Proxy'deki aynı adlı yöntemi harekete geçirerek çağırır. Proxy, bu yerel çağrılarını SOAP isteklerine dönüştürür ve web servisine gönderir. Proxy yanıtı bekler, verinin yanıtını bekler ve herhangi bir basit yöntem çağrısındaki gibi istemciye geçirir.

Burada sizin yaptığınız uygulamada bu yaklaşım kullanılmıştır. Bu en kolay yaklaşım biçimidir.

## UYGULAMA FAALİYETİ

Bu modülün Öğrenme Faaliyeti-1 başlıklı bölümünde geliştirdiğiniz “Mini Kredi Hesaplama” programını web servis şeklinde oluşturup bu programı aynı zamanda Windows uygulaması şeklinde de çalıştıracak şekilde yeniden kodlayınız.

| İşlem Basamakları  | Öneriler   |
|--|--|
| ➤ Nesne Tabanlı Programlama Yazılımı programında en az NET Framework 3 destekli bir web site oluşturunuz                           | ➤ File > New > Web Site menü seçeneklerini kullanabilirsiniz.  |
| ➤ Oluşturduğunuz web sitesine yeni bir WCF Servis ögesini “Servis.svc” isminde ekleyiniz.  | ➤ File > New > File menü seçeneklerini kullanabilirsiniz.  |
| ➤ Web servise kontrat dosyasına (IService.cs dosyası) kullanacağınız fonksiyonunuzu tanımlayınız.                                  | ➤ <i>double KrediAylıkTutar(int AnaPara, int tur, int vade);</i> şeklinde bir tanımlama yapabilirsiniz.                                |
| ➤ Web servis kontrat dosyasında tanımladığınız fonksiyonunun (Örnek: KrediAylıkTutar) kodlamasını “Service.cs” dosyasında yapınız. | ➤ Resim 1.8’de yer alan kodlamayı, bir değer döndürecek şekilde yeniden kodlayınız.  |
| ➤ Web servisinizi derleyerek hizmetin oluşmasını sağlayınız.   | ➤ Service.svc dosyasını tarayıcıda görüntüleyerek oluşmasını sağlayabilirsiniz.  |
| ➤ Yeni bir Windows uygulaması oluşturunuz.   | ➤ Nesne Tabanlı Programlama Yazılımı size yardımcı olabilir.   |
| ➤ Bu Windows uygulamasına, oluşturduğunuz web servis hizmetini refererans gösteriniz.  | ➤ Project > Add Service Reference menü seçeneklerini kullanabilirsiniz.  |
| ➤ Form görüntüsünü ve kodlamanızı tamamlayarak web servisle bağlantınızı gerçekleştirip sonucu görüntüleyiniz.                     | ➤ Form görüntüsü Resim 4.6’daki gibi olabilir.<br> |

Resim 4.6: Örnek form görüntüsü

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

| <b>Değerlendirme Ölçütleri</b>  | <b>Evet</b> | <b>Hayır</b> |
|---|-------------|--------------|
| 1. Bir web servis oluşturabildiniz mi?  |             |              |
| 2. Bir web servisin tanımını görüntüleyebildiniz mi?                                  |             |              |
| 3. Karmaşık verileri, web yöntem parametreleri olarak verip değer döndürebildiniz mi? |             |              |
| 4. Bir uygulamaya web başvurusu ekleyebildiniz mi?                                    |             |              |
| 5. Bir web yöntemini çağırabildiniz mi?   |             |              |

## DEĞERLENDİRME

Değerlendirme sonunda “Hayır” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “Evet” ise “Ölçme ve Değerlendirme”ye geçiniz.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. WCF (Windows Communication Foundation) web servis oluşturabilmek için minimum kullanılması gereken Net Framework versiyonu aşağıdakilerden hangisidir?  
A) Net Framework 1.1  
B) Net Framework 2.0  
C) Net Framework 3.0  
D) Net Framework 3.5
2. Bir Windows projesine web servis referansı vermek için aşağıdaki menü seçeneklerinden hangisi kullanılır?  
A) Project > Add Service Reference  
B) Building > Add Service Reference  
C) Format > Add Service Reference  
D) Tools > Add Service Reference
3. Web servislerle ilgili olarak aşağıdaki seçeneklerden hangisi doğrudur?  
I. Web üzerinden çalışan program kodlamasıdır.  
II. Bütün kodlamalar tamamlandıktan sonra derlenerek oluşturulur.  
III. XML tabanlı veri biçimini kullanan yapılardır.  
A) I ve II  
B) II ve III  
C) I ve III  
D) I, II ve III

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise **D**, yanlış ise **Y** yazınız.

4. ( ) Bir Windows projesine, web servis referans verildiğinde bu projeye *Service Reference* klasörü eklenir.
5. ( ) Bir WCF web servis oluşturulduğunda, bu web serviste kullanılacak fonksiyonların listesini barındıran dosya, projedeki *App\_Code* klasörüne eklenir.

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise “Modül Değerlendirme”ye geçiniz.

# MODÜL DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. İstemci bilgisayardan talep edilen ASP.NET destekli bir internet sayfasını, sunucu bilgisayardaki istemciye hangi yazılım tarafından gönderir?
  - A) ASP.NET Yorumlayıcı
  - B) NET Framework
  - C) Web Sunucu (IIS)
  - D) Nesne Tabanlı Programlama Yazılımı
2. ASP.NET web formu geliştirmek için kullanılan yazılım araçlarında proje geliştirme ekranında yer alan Solution Explorer penceresinde, aşağıdaki işlemlerden hangileri gerçekleştirilemez?
  - I. Yeni bir ASP.NET dosyası oluşturma
  - II. Web form dosyasını tarayıcıda ön izleme
  - III. ASP.NET sayfasına web servis referansı verme
  - IV. Projedeki bir dosyası başka bir klasöre taşıma
  - A) I ve II
  - B) II ve III
  - C) II ve IV
  - D) I ve III
3. Bir ASP.NET sayfasını web tarayıcıya göndermek için aşağıdakilerden hangileri yapılır?
  - I. Build > Build Solution menü seçenekleri kullanılır.
  - II. File > View in Browser menü seçenekleri kullanılır.
  - III. Debug > Start Debugging menü seçenekleri kullanılır.
  - A) I ve II
  - B) I ve III
  - C) II ve III
  - D) I, II ve III
4. Bir ASP.NET sayfasına tablo eklemek için kullanılan *Insert Table (Table > Insert Table* menü seçenekleri) penceresinden aşağıdaki hangi ayarlama yapılamaz?
  - A) Tablonun kenarlık rengi ayarlaması
  - B) Tablonun genişlik ayarlaması
  - C) Tablodaki verilerin hizalanması
  - D) Tablodaki metin rengi
5. DropDownList kontrolünden seçilen öğeyi bulmak için kullanılan kodlama aşağıdakilerden hangisidir?
  - A) DropDownList1.SelectedIndex
  - B) DropDownList1.SelectedMember
  - C) DropDownList1.SelectedDisplay
  - D) DropDownList1.SelectedText

6. Bir ASP.NET sayfasına css stilleri eklemek için hangi menü seçeneği kullanılır?
- A) Style > New Style
  - B) Format > New Style
  - C) Edit > CSS > New Style
  - D) View > Add > New CSS
7. Bir ASP.NET sayfasındaki “TextBox5.Focus();” ifadesi ile yapılmak istenen aşağıdakilerden hangisidir?
- A) TextBox5 kontrolü veri yazmaya kapatılmıştır.
  - B) TextBox5 kontrolündeki veri silinir.
  - C) TextBox5 kontrolünü gizleyen bir satırdır.
  - D) İmleci TextBox5 kontrolüne konumlandırır.
8. Aşağıdakilerden hangisi veya hangileri RequiredFieldValidator ile RangeValidator kontrollerinin ortak özelliklerindedir?
- I. MaximumValue
  - II. ControlToValidate
  - III. ErrorMessage
- A) I ve II
  - B) II ve III
  - C) I ve III
  - D) I, II ve III
9. ASP.NET Web Sitesi Yönetim Aracı sisteminden aşağıdakilerden hangileri yapılabilir?
- I. Sistemde “kalderon” isimli yeni bir kullanıcı oluşturulabilir.
  - II. Sistemdeki “GizliBelgeler” isimli klasöre “demli” isimli kullanıcı yasaklanabilir.
  - III. III.”demli” isimli kullanıcı, başka kullanıcılarla aynı gruba alınabilir.
- B) I ve II
  - C) II ve III
  - D) I ve III
  - E) I, II ve III
10. Veriyi ASP.NET sayfasında görüntülemek için kullanılan GridView kontrolü ToolBox’da hangi bölümde yer almaktadır?
- A) Validation
  - B) Standart
  - C) Data
  - D) Navigation
11. GridView kontrolündeki görüntülenecek alan bir veriyi sağa hizalamak için aşağıdaki seçeneklerden hangisi kullanılır?
- A) EditColumns
  - B) ConfigColumns
  - C) WebConfig
  - D) Properties

12. Kullanıcıların web servislerine istekler göndermek ve onlardan gelen talepleri yanıtlamak için aşağıdaki yöntemlerden hangi kullanılır?
- A) HTTP
  - B) SOAP
  - C) WCF
  - D) Reference

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.



# CEVAP ANAHTARLARI

## ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

|   |   |
|---|---|
| 1 | A |
| 2 | B |
| 3 | D |
| 4 | A |
| 5 | C |
| 6 | A |

## ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

|   |   |
|---|---|
| 1 | A |
| 2 | D |
| 3 | A |
| 4 | C |

## ÖĞRENME FAALİYETİ-3'ÜN CEVAP ANAHTARI

|   |              |
|---|--------------|
| 1 | A            |
| 2 | C            |
| 3 | B            |
| 4 | C            |
| 5 | ASPNETDB.MDF |
| 6 | Login        |

## ÖĞRENME FAALİYETİ-4'ÜN CEVAP ANAHTARI

|   |   |
|---|---|
| 1 | C |
| 2 | A |
| 3 | D |
| 4 | D |
| 5 | D |

## MODÜL DEĞERLENDİRMENİN CEVAP ANAHTARI

|    |   |
|----|---|
| 1  | C |
| 2  | B |
| 3  | C |
| 4  | D |
| 5  | A |
| 6  | B |
| 7  | D |
| 8  | B |
| 9  | D |
| 10 | C |
| 11 | A |
| 12 | B |

# KAYNAKÇA

- SHARP, John, **Adım Adım Microsoft Visual Studio C# 2008**, Arkadaş Yayınevi, Ankara, 2009.
- ALGAN, Sefer, **Her Yönüyle C#**, Pusula Yayınevi, İstanbul, 2010.
- <http://www.yazilimokulu.net/Makaleler/100-bilgi-dogrulama-kontrolleri1-requiredfieldvalidator-kontrol.html> (20.08.2011 / 11.50)
- [http://www.turgaysahtiyen.com/post/ASPNet-te-Form-Dogrulama-\(Validation\).aspx](http://www.turgaysahtiyen.com/post/ASPNet-te-Form-Dogrulama-(Validation).aspx) (21.08.2011 / 15.40)
- <http://www.csharpnedir.com/articles/read/?id=546&title=Asp.Net%202.0%20Kullan%C4%B1c%C4%B1%20Y%C3%B6netiminde%20Login%20Kontrollerinin%20Kullan%C4%B1m%C4%B1> (23.08.2011 / 14.30)
- <http://www.becerikli.net/threads/98971-ASP.Net-de-Login-Kontrolleri-Olu%C5%9Fturma> (24.08.2011 / 19.20)
- [http://www.argelab.net/makaleler/asp\\_net\\_uyelik\\_sistemi/index.html](http://www.argelab.net/makaleler/asp_net_uyelik_sistemi/index.html) (25.08.2011 / 12.50)
- <http://www.yazilimokulu.net/Makaleler/203-data-kontrolleri--gridview-kontrol.html> (26.08.2011 / 18.00)
- <http://www.yazilimokulu.net/Makaleler/206-gridview-kontrolu-ile-guncelleme-ve-silme-islemleri.html> (27.08.2011 / 11.00)
- <http://www.ummugulsum.net/what-is-wcf.html/> (05.09.2011 / 12.10)
- <http://www.sumeyyekok.com/?p=237> (05.09.2011 / 15.40)
- <http://www.sanalkurs.net/wcf-nedir-nasil-kullanilir--3849.html> (06.09.2011 / 10.00)