

T.C.
MİLLİ EĞİTİM BAKANLIĞI



MEGEP

(MESLEKİ EĞİTİM VE ÖĞRETİM SİSTEMİNİN
GÜÇLENDİRİLMESİ PROJESİ)

BİLİŞİM TEKNOLOJİLERİ

**GÖRSEL PROGRAMLAMADA
İNTERNET UYGULAMALARI**

ANKARA 2007

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, Mesleki ve Teknik Eğitim Okul ve Kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğretim materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere Mesleki ve Teknik Eğitim Okul ve Kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak, amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlıkta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşılabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

İÇİNDEKİLER

AÇIKLAMALAR	iii
GİRİŞ	1
ÖĞRENME FAALİYETİ-1	3
1. WEB FORM APPLICATION	3
1.1. IIS (Internet Information Services), URL, HTML Ve HTTP Deyimleri	3
1.1.1. IIS (Internet Information Services)	3
1.1.2. URL	5
1.1.3. HTML	5
1.1.4. HTTP	5
1.2. Microsoft FrontPage Server Extensions (FPSE)	5
1.3. Web Form ve Web Sayfası Kavramları	5
1.4. aspx Uzantısı	6
1.5. “http://localhost” Adresi ve “c:\inetpub\wwwroot” Klasörü	6
1.6. “Web Form Designer” Penceresi ve Source Sekmesi	7
1.7. “Web Server Controls” Nesneleri	8
1.7.1. Label	9
1.7.2. TextBox	9
1.7.3. Button	9
1.7.4. LinkButton	10
1.7.5. ImageButton	10
1.8. Validation Controls	10
1.8.1. RequiredFieldValidator	10
1.8.2. CompareValidator	11
1.8.3. RangeValidator	12
1.8.4. Custom Validator	14
1.8.5. RegularExpressionValidator	14
1.8.6. ValidationSummary Nesnesi	15
1.9. “AutoPostBack ve EnableViewState” özellikleri	17
1.9.1. AutoPostBack	17
1.9.2. EnableViewState	19
1.10. Basit HTML Etiketleri (HTML, body, title, form...)	20
1.10.1. HTML	20
1.10.2. HEAD	21
1.10.3. BODY	21
1.10.4. TITLE	22
1.10.5. FORM	22
1.10.5.4. Menüler	23
1.10.6. Center Etiketi	24
1.10.7. Başlık Etiketleri	24
1.10.8. , <i>, <s>, <u>, <big>, <small>, <tt> Etiketleri	25
1.11. “Set As Start Page” Seçeneği	25
UYGULAMA FAALİYETİ	27

ÖLÇME VE DEĞERLENDİRME.....	28
ÖĞRENME FAALİYETİ-2	29
2. “ADO.NET”İ WEB UYGULAMASINDA KULLANMA	29
2.1. DataGrid Nesneleri.....	29
2.2. Kod Penceresinde Nesneleri Veriye Bağlama	33
2.3. DataBind komutu	35
2.4. DataBindings Diyalog Kutusu	35
UYGULAMA FAALİYETİ.....	37
ÖLÇME VE DEĞERLENDİRME.....	38
ÖĞRENME FAALİYETİ-3	39
3. WEB SERVİSLERİ	39
3.1. “Distributed Web Applications” Deyimi	39
3.2. SOAP (Simple Object Access Protocol)	40
3.3. WSDL (Web Services Description Language)	42
3.4. “*.asmx ve disco” Uzantısı	44
3.5. “Add Web Reference”	46
UYGULAMA FAALİYETİ.....	53
ÖLÇME VE DEĞERLENDİRME.....	54
CEVAP ANAHTARLARI.....	58
ÖNERİLEN KAYNAKLAR.....	60
KAYNAKÇA	61

AÇIKLAMALAR

KOD	482BK0073
ALAN	Bilişim Teknolojileri
DAL/MESLEK	Veri Tabanı Programcılığı
MODÜLÜN ADI	Görsel Programlamada İnternet Uygulamaları
MODÜLÜN TANIMI	İnternet uygulamaları ve web servisleri yapabilme becerilerinin kazandırıldığı öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	Görsel Programlamada Veri Tabanı modülünü almış olmak.
YETERLİK	Görsel programlamada İnternet uygulamaları yapmak
MODÜLÜN AMACI	Genel Amaç Gerekli ortam sağlandığında, internet uygulamaları ve web servisleri yapabileceksiniz. Amaçlar Bu modül ile gerekli ortam sağlandığında 1. Web Form Application ile çalışabileceksiniz. 2. ADO.NET’i internet uygulamanızda kullanabileceksiniz. 3. Web servisleri ile çalışabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Görsel programlama için gerekli donanıma sahip bilgisayar, yedekleme için gerekli donanım, raporlama için yazıcı, sayfa için internet bağlantısı, kâğıt ve kalem hazır bulundurulmalıdır.
ÖLÇME VE DEĞERLENDİRME	Her faaliyet sonrasında o faaliyetle ilgili değerlendirme soruları ile kendi kendinizi değerlendireceksiniz. Öğretmen modül sonunda size ölçme aracı (uygulama, soru-cevap) uygulayarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek değerlendirecektir.

GİRİŞ

Sevgili Öğrenciler,

Günümüzde hızlı bir şekilde yaygınlaşan, alışverişten ticarete, bilimsel araştırmalardan, eğlenceye, sivil toplum kuruluşlarının örgütlenmesinden, siyasi partilerin propagandasına kadar hayatın her alanını içine alan internet, insanlık tarihi açısından yeni bir dönüm noktasıdır. Nitekim internetin “3. Devrim” ya da “Enformasyon Devrimi” olarak isimlendirilmesi, sahip olduğu büyük dönüştürücü potansiyele işaret etmektedir. Soğuk savaşın etkili olduğu 1960’ların ikinci yarısında, askeri güvenlik kaygısıyla geliştirilen internetin ancak 1990’ların başında yaygınlaşmaya başlaması hem teknolojik hem de toplumsal ve siyasal dönüşüm potansiyelini anlamak açısından önemlidir. (Cogito, 2002;30)

Günümüzde internet yaşamın tüm boyutlarını etkilemeye başlamış, gündelik yaşamın önemli bir bölümünü oluşturan faaliyetlerin (şimdilik) birçoğunu kendi sanal sınırları içerisine dâhil edebilmiştir. Söz gelimi, internet sayesinde herhangi bir faturayı yatırmak için evinizden ayrılıp bankaya kadar gitmenize gerek kalmamıştır. Nitekim, sanayi toplumunda ciddi efor ve zaman gerektiren birçok günlük faaliyet artık günümüzde, reklamlarda da belirtildiği gibi, “tek tuş” ile halledilebilir olmuştur.

Bu modülün kapsadığı web form uygulamaları, web sayfasında veri tabanı kullanma ve web servisleri konularıyla internette sadece sörf yapıp sayfalar arasında gezmek yerine, kapsamlı internet sayfaları ve web servisleri hazırlayabileceksiniz.

ÖĞRENME FAALİYETİ-1

AMAÇ

Gerekli ortam sağlandığında Web Form Application ile çalışarak, form tasarım penceresinde gerekli nesnelere ekleyebilecek, programı çalıştırarak tarayıcıda görebileceksiniz.

ARAŞTIRMA

Bu faaliyet öncesinde yapmanız gereken öncelikli araştırmalar şunlardır:

İnternet sayfalarını inceleyerek dosya uzantılarına göre neler yapabildiklerini inceleyiniz.

Kullandığınız bilgisayarın işletim sistemini ve neler yapabileceğini araştırınız.

Araştırma işlemleri için internet ortamını kullanınız.

1. WEB FORM APPLICATION

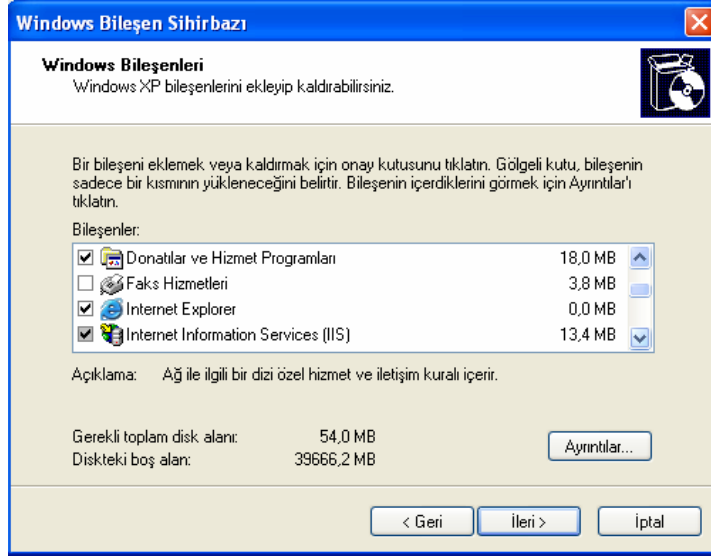
1.1. IIS (Internet Information Services), URL, HTML Ve HTTP Deyimleri

1.1.1. IIS (Internet Information Services)

Frontpage, FTP, ASP, veri tabanı bağlantıları desteği ile birlikte web ve FTP desteği sağlar.

IIS (Internet Information Services - İnternet Bilgi Servisi), Microsoft'un ASP (Active Server Pages-Etkin Sunucu Sayfaları) yazılımları geliştirmek için hazırladığı Web servisidir. IIS, yerel bir server kurarak, ASP dosyalarının sunucuya gönderilmeden bilgisayarımızda çalıştırılmasını sağlar.

IIS kurmak için Denetim Masası → Program Ekle Kaldır → Windows Bileşenleri Ekle Kaldır tıklanarak veya Windows XP Professional CD'sini takınca çıkan pencereden Seçime Bağlı Windows Bileşenlerini Yükle seçeneği tıklanıldığında Şekil 1.1'deki Windows Bileşen Sihirbazı penceresi gelecektir.



Şekil 1.1: Windows Bileşen Sihirbazı

Burada Internet Information Services (IIS) seçilerek (ayrıntılar kısmında istenmeyen bazı özellikler iptal edilebilir veya istenenler eklenebilir) İleri butonuyla sihirbaza devam edilir. Kurulum başarı ile tamamlandığında sabit diskinizde “inetpub” adında bir klasör oluşacaktır.

Tarayıcımızı (Internet Explorer, Netscape vb.) açarak adres çubuğuna <http://localhost> yazarak Şekil 1.2’deki iki asp sayfasını izleyebiliyorsak IIS çalışıyor demektir.



Şekil 1.2: localstart.asp dosyası

Eğer IIS Visual Studio 2005’den sonra kurmuşsanız muhtemelen hata ile karşılaşacaksınız. Bunu gidermek için .Net Framework’u kaldırarak tekrar kurmanız gerekecek. Visual Studio DVD veya CD içerisinde .Net Framework kurulum dosyasını bulabilirsiniz.

Artık ASP dosyalarınızı tarayıcıdan izlemek için dosyalarınızı inetpub klasöründeki wwwroot klasörüne kopyalayarak tarayıcınızın adres çubuğuna http://bilgisayar_adi/dosya_adi veya http://localhost/dosya_adi yazarak çalıştırabilirsiniz.

1.1.2. URL

URL (Uniform Resource Locators), Web Browser'lar (tarayıcılar) içinden bir Web servisine ya da diğer bazı internet servislerine yönlendirme yapılabilmesini sağlayan bir komut formatıdır. URL'ler bir bakıma, internet üzerinde erişebileceğimiz servisleri belirtmek, tanımlamak için kullandığımız adreslerdir

Her Web sayfası sadece kendine ait bir URL'ye sahiptir. Örneğin http://www.yyeml.k12.tr/default.asp bir URL'dir. Bu URL üç bölümden oluşur:

Protokol (http://, ftp://, news: vb.); alan adı (protokolden, ondan sonraki bölü işaretine kadar olan kısım); dosya yolu (ilk bölü işaretinden sonraki kısım).

1.1.3. HTML

HTML (Hyper Text Markup Language), bir belge biçimleme dilidir. Tag (etiket) kullanılarak text metinler biçimlenirken aynı zamanda belgeler arasında veya belgenin içinde köprüler sağlamak ile ilgili komutlar içerir. HTML dosyaları html veya htm uzantısı ile saklanır. HTML, HTTP ve ilgili diğer protokolleri kullanabilmek için renkli ve güzel kullanıcı ara yüzleri hazırlamamızı sağlar.

1.1.4. HTTP

HTTP (Hyper Text Transport Protocol) günümüzde internet ortamı üzerinde Web sayfalarının çeşitli istemcilere iletimi sırasında bir bütünlük sağlanması açısından oluşturulmuş bir standartlar dizisidir. Günümüzde en yaygın olarak bilinen ve kullanılan protokoldür.

1.2. Microsoft FrontPage Server Extensions (FPSE)

Microsoft FrontPage Server Extensions, FrontPage ile oluşturulmuş bir Web sitesini yönetmeyi, düzenlemeyi ve göz atmayı desteklemek için Microsoft Internet Information Services ile çalışan bir grup programdır. FrontPage istemcisi çalıştıran kullanıcılar, böylece kişisel bir Web sayfası ya da bir şirket İnternet veya intranet sitesi oluştururken, profesyonel görünümlü Web siteleri hazırlayabilirler. FrontPage Server Extensions, yönetici olarak size site güvenliğini yönetmek, içeriğinizi alt Web'ler de düzenlemek ve site kullanımını denetlemek için kullanabileceğiniz araçlar sağlar.

1.3. Web Form ve Web Sayfası Kavramları

ASP.NET ile programlama yaparken Web formlarını kullanırız, Web formları iki bileşenden oluşur, görsel kısım ve kod kısmı. Görsel kısmı .aspx dosyaları teşkil eder ve içlerinde HTML elemanlarını ve Web form kontrollerini barındırır. Kod kısmı ise ayrı bir dosyada bulunabileceği gibi .aspx dosyamızın içinde de yer alabilir.

Web istemcisinin, bir Web Servisine bağlandıktan sonra tek seferde transfer ettiği verilerden oluşan html sayfasıdır. Bu sayfadaki bilgiler hem grafik/resim bilgileri, hem normal text, hem ses, hemde başka merkezlere ve başka dokümanlara linkler olabilir.

1.4. aspx Uzantısı

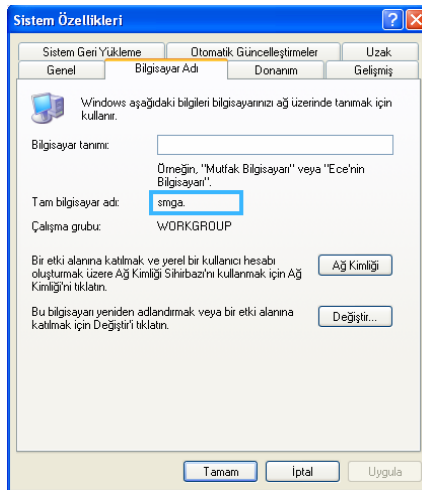
ASP.NET (Active Server Pages), klasik HTML sayfalarına dinamik bir yapı kazandırmak amacıyla Microsoft firması tarafından ortaya çıkarılmış platform ve programlama dilinden bağımsız sunucu taraflı bir Web teknolojisidir. İlk adı ASP olan ASP.NET, Microsoft'un yeni sunucu ailesi ve İnternet teknolojilerini .NET platformu altına alması kararından sonra ASP.NET adını aldı. ASP ile oluşturulan dinamik sayfalar, "asp" uzantısını alıyor. ASP.NET ile oluşturulan dinamik sayfalarsa, "aspx" uzantısını alıyor. Uzantıların farklı olması aynı sunucuda hem ASP'nin, hem de ASP.NET'in sorunsuzca çalışabilmesini sağlıyor.ASP.NET ile oluşturulan Web formları aspx uzantılıdır.

1.5. “http://localhost” Adresi ve “c:\inetpub\wwwroot” Klasörü

Asp veya aspx uzantılı dosyalar yukarıda da belirttiğimiz gibi sunucu taraflı çalışan dosyalardır. Bu dosyaları çalıştırmak demek, dosyaları sunucuya gönderip orada çalıştırdıktan sonra html kodları şeklinde geri gelmesi demektir.

Bilgisayarımıza yüklediğimiz IIS, asp veya aspx dosyalarını sunucuya göndermeden, bilgisayarımızda kurulan yerel sunucu'da çalıştırmayı sağlar. IIS kurulduktan sonra tarayıcımızın adres çubuğuna http://localhost veya http://bilgisayar_adi yazdığımız zaman c:\inetpub\wwwroot klasöründeki index veya default sayfası tarayıcımızda açacaktır (html veya asp).

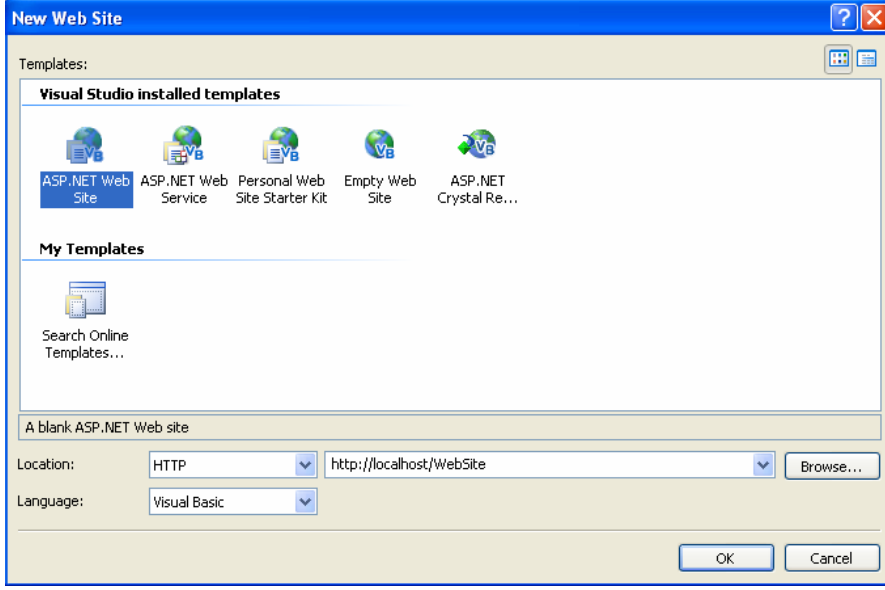
Çalıştırılacak aspx dosyasını wwwroot klasörüne kopyaladıktan sonra http://localhost/dosyaadi.aspx yazarak sayfamızı açabiliriz.



Şekil 1.3: Bilgisayar adı

1.6. “Web Form Designer” Penceresi ve Source Sekmesi

Microsoft Visual Studio çalıştıktan sonra File →New Web Site menüsüne tıklayarak Şekil:1.4'teki pencereye ulaşırız.



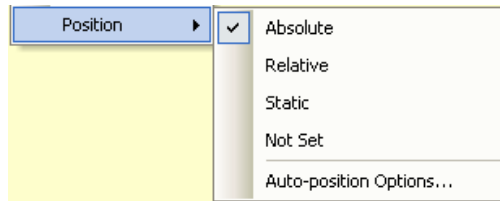
Şekil 1.4: Yeni Web sayfası

Location satırında http seçince, karşısında http://localhost/WebSite gelecektir. Bu, yeni sayfamızın c:\inetpub\wwwroot klasöründe saklanacağı anlamına gelmektedir (aspx dosyalarını çalıştırabilmek için bu klasörde saklanması gerekiyor). Eğer istenirse location kısmında FTP veya File system seçenekleri de bulunmaktadır. OK tıklandığında Şekil:1.6'daki Web Form Designer penceresine ulaşılmış olur.

Web formuna Windows formundan farklı olarak doğrudan metin eklenebilir. Formatting araç kutusu kullanılarak metinde biçimlendirme değişiklikleri yapılabilir.

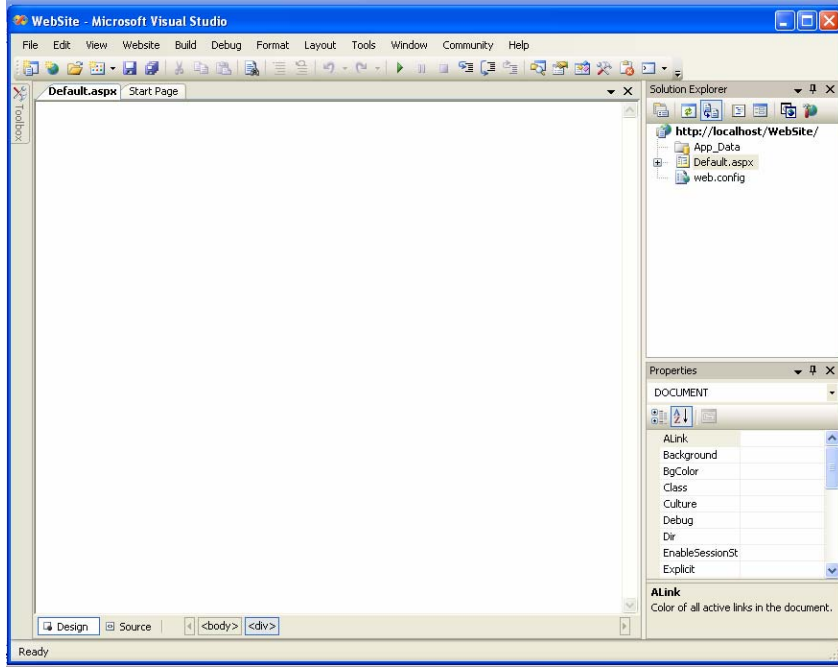
Pencerenin sol alt köşesinde bulunan Design ve Source butonlarını kullanarak dizayn görünümünden html kodları görünümüne veya html kodları görünümünden dizayn görünümüne geçebiliriz.

Dizayn görünümündeyken Windows Formundan farklı olarak ekrana yazı yazabiliriz. Kontrol elemanlarını form üzerinde istediğimiz yere taşıyabilmek için Layout menüsünden Şekil 1.5'teki seçenekler kullanılır.



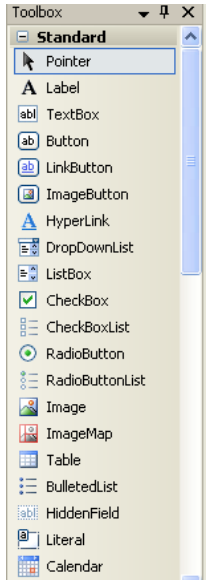
Şekil 1.5: Position seçenekleri

Web formumuzun, Properties penceresinde DOCUMENT seçili iken sayfa başlığı (Title), zemin rengi (BgColor), arka plan resmi(BackGround), link renklerini (ALink, Link, VLink) seçebiliriz.



Şekil 1.6: Web Form Designer

1.7. “Web Server Controls” Nesneleri



Şekil 1.7: Toolbox

Araç çubuğundan (View → Toolbox) istediğimiz kontrolleri seçerek formumuza ekleyebiliriz.

Bu kontrollerden biri seçili iken Properties penceresinde aşağıdaki özellikleri ayarlayabiliriz:

Access Keys :Kontrolle klavye ile ulaşabilmek için bir tuş atar. Bu tuşa Alt tuşu ile birlikte basıldığında kontrol seçilmiş olur.

TabIndex :Sekme sırasını ayarlar. 0'dan başlar sırası ile küçükten büyüğe doğru ilerler.

ToolTip : Fare işaretimiz bir kontrolün üzerinde bekletildiğinde ToolTip metni görüntülenir. Genellikle o kontrol ile ilgili bilgi vermek için kullanılır.

1.7.1. Label

Web formumuzun istediğimiz yerde yazıları gösterebilen bir etikettir. Label kontrolüne dışarıdan değer girilemez.

Örnek:

Formumuzda butona tıklanıldığında sayfamıza yeni bir label ekleyerek Text özelliğine “Bilişim Teknolojileri” yazısını eklemeye çalışalım.

Button1’in klik olayına;

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As
_System.EventArgs) Handles Button1.Click
    Dim etiket As New Label
    etiket.Text = "Bilişim Teknolojileri"
    Page.Controls.Add(etiket)
End Sub
```

yazalım.

1.7.2. TextBox

Web formumuzda kullanıcı tarafından bilgi girilmesi için kullanılan metin kutusudur.

Örnek:

Formumuzda butona tıklanıldığında textbox’a girilen string türündeki bilgiyi büyük harfe çevirip diğer textbox’a eklemeye çalışalım.

Button1’in klik olayına;

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As
_System.EventArgs) Handles Button1.Click
    TextBox1.Text = TextBox2.Text.ToUpper
End Sub
```

yazalım.

1.7.3. Button

Genellikle bir işlemi yerine getirmek için tıklanılarak kullanılan kontrol elemanıdır. Örneğin formu temizlemek veya göndermek için kullanılabilir.

Örnek:

Formumuzda butona tıklanıldığında TextBox içerisindeki karakterleri silmeye çalışalım.

Button1’in klik olayına;

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As
_System.EventArgs) Handles Button1.Click
    TextBox1.Text = ""
End Sub
yazalım.
```

1.7.4. LinkButton

Web formdaki buton kontrolünün, html koduyla yazılmış köprü içeren yazı şeklinde bir kontroldür.



Şekil 1.8: LinkButton ile Buton görünüşleri

Button kontrolündeki örneğin aynısını LinkButton içinde yapabiliriz.

1.7.5. ImageButton

Web formda resim gösteren ve resme tıklanıldığında yanıt veren bir kontroldür. Resim eklemek için özellikler penceresinden ImageUrl seçeneğinden resmi seçip, ImageAlign seçeneğinden resmin hizalamasını yapabiliriz.

1.8. Validation Controls

Web sayfasında kullanılan girişlerin onaylanması validation kontrolleri kullanılarak yapılabilir. Girilen tarihin geçerli olup olmadığını veya değerlerin belirli bir aralıkta olup olmadığını kontrol eder. Geçerli olup olmadığını test eder. Validation kontrolleri kullanıcının hata bilgilerinin nasıl görüntüleneceğini düzenlemesine izin verir.

1.8.1. RequiredFieldValidator

Web sayfasında belirli kontrollerin yanına RequiredFieldValidator kontrolünü ekleyerek bu kontrollerin gereklilik testini yapabilirsiniz. Örneğin kullanıcının bir kayıt formunu göndermeden önce Adı alanını doldurması gerektiğini belirleyebilirsiniz.

ControlToValidate özelliği ile hangi kontrolün gereklilik testi yapılacağını belirleriz.

ErrorMessage özelliği ile gereklilik ihlal edildiğinde sayfada görünmesini istediğimiz mesaj yazılır.

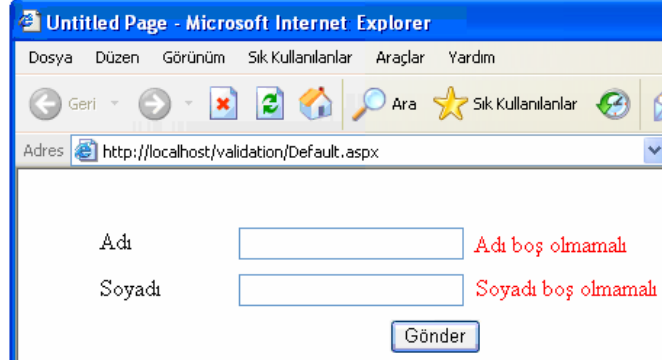
Örnek olarak sadece ad ve soyad içeren çok basit bir kullanıcı kayıt formu yapalım. Bunun için Web Formumuza iki adet Label, iki adet TextBox, iki adet RequiredFieldValidator ve bir adet Buton kontrolü ekleyelim.

TextBox kontrolünün özelliklerinde

Kontrol	ID
TextBox1	txtad
TextBox2	txtsoyad

RequiredFieldValidator özellikleri

Kontrol	ControlToValidate	ErrorMessage
RequiredFieldValidator1	txtad	Adı boş olmamalı
RequiredFieldValidator2	txtsoyad	Soyadı boş olmamalı



Şekil 1.9: Adı ve soyadı bilgisi kayıt için gereklidir

1.8.2. CompareValidator

Web sayfasında kullanıcının sayısal, tarih, string veri türlerinde girişlerinin geçerlilik testini CompareValidator kontrolünü kullanarak yapabilirsiniz. Örneğin kullanıcı kayıt sayfasında doğum tarihi bilgilerini almak istiyorsanız, CompareValidator kontrolüyle girilen tarihlerin geçerli tarih formatında olmasını veya belirli bir tarihten önce veya sonra girilmemesini isteyebilirsiniz.

ControlToValidate özelliği ile hangi kontrolün geçerlilik testi yapılacağını belirleriz.

ErrorMessage özelliği ile geçerlilik ihlal edildiğinde sayfada görünmesini istediğimiz mesaj yazılır.

ValueToCompare özelliği ile bizim girdiğimiz bir değere göre karşılaştırma yapılır.

ControlToCompare özelliği ile sayfadaki aynı veri türünde bilgi içeren başka bir kontrol elemanı ile karşılaştırma yapılır.

Type özelliği ile geçerlilik testi yapılacak kontrolün veri türü belirlenir.

Operatör özelliği ile karşılaştırma operatörü seçilir.

Operatör	Açıklama
DataTypeCheck	Sadece veri türünü karşılaştırır
Equal	Eşittir
GreaterThan	Büyük
GreaterThanEqual	Büyük eşit
LessThan	Küçük
LessThanEqual	Küçük eşit
NotEqual	Eşit değil

Yapacağımız örnekte kullanıcı adı, yeni şifre ve doğrulamak için tekrar yeni şifre girmesini isteyeceğiz. CompareValidator kontrolüyle girilen iki şifrenin aynı olmasını sağlayacağız. Bunun için web formumuza üç adet TextBox, üç adet Label ve bir adet CompareValidator kontrolü ekleyelim.

TextBox özellikleri;

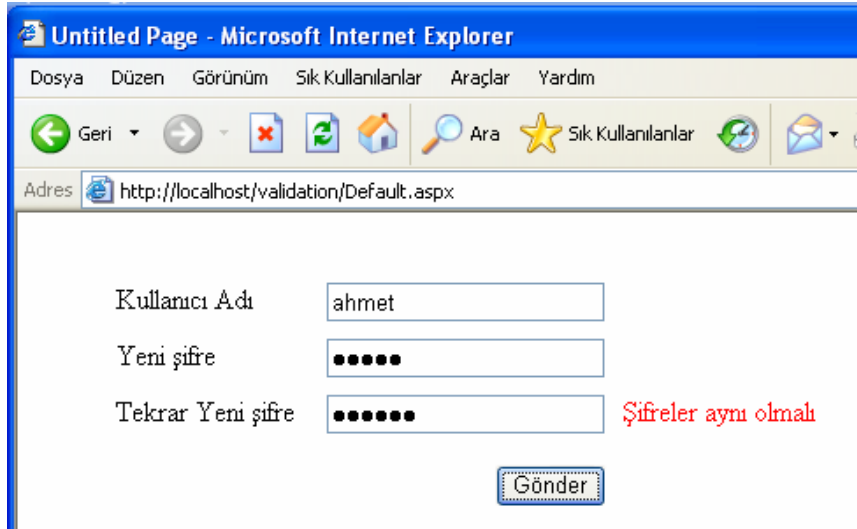
Kontrol	ID	TextMode
TextBox1	txtuser	SingleLine
TextBox2	txtsifre	Password
TextBox3	txttekrar	Password

CompareValidator özellikleri

ControlToValidate	ControlToCompare	ErrorMessage	Type	Operator
txttekrar	txtsifre	Şifreler aynı olmalı	String	Equal

şeklinde değiştiriyoruz.

Web sayfasını çalıştırdığımızda şifreler aynı girilmediği zaman Şekil 1.10'daki hata mesajı görüntülenecektir.



Şekil 1.10: Şifrelerin aynı girilmesi istenecek

1.8.3. RangeValidator

Web sayfasında kullanıcı girişlerinin belirli bir aralıkta olmasını; örneğin iki sayı arasında, iki tarih arasında veya alfabetik karakterler arasında olmasını RangeValidator kontrolü ile sağlayabiliriz.

ErrorMessage özelliği ile gereklilik ihlal edildiğinde sayfada görünmesini istediğimiz mesaj yazılır.

Type özelliği ile geçerlilik testi yapılacak kontrolün veri türü belirlenir.

MaximumValue özelliği ile seçilen kontrole girilebilecek en büyük değer belirlenir.

MinimumValue özelliği ile seçilen kontrole girilebilecek en küçük değer belirlenir.

Yapacağımız örnekte öğrenci adı, yazılı ve sözlü notları girecek bir formda notun 0 ile 100 arasında olmasını istiyoruz.

Web formumuza üç Label, üç TextBox ve iki adet RangeValidator ekliyoruz.

TextBox özelliklerini,

Kontrol	ID
TextBox1	txtogrenci
TextBox2	txtyazili
TextBox3	txtsozlu

RangeValidator özelliklerini

Özellik	ControlTo Validate	ErrorMessage	Minimum Value	Maksimum Value	Type
Range Validator1	txtyazili	Yazılı notu 0–100 arasında olmalıdır	0	100	Integer
Range Validator2	Soyad	Sözlü notu 0–100 arasında olmalıdır	0	100	Integer

Öğrenci

Yazılı Yazılı notu 0-100 arasında olmalıdır.

Sözlü Sözlü notu 0-100 arasında olmalıdır.

Şekil 1.11: 0 ile 100 arası değer isteniyor

1.8.4. Custom Validator

ASP.NET 'in web programcısına sunduğu validation kontrolleri yetersiz kaldığında CustomValidator kullanılabilir. CustomValidator 'ın görevi programcı tarafında yazılan bir sub routine (alt program) ile verilir. Denetlemenin sonucu da sub routine içerisinde belirtilir.

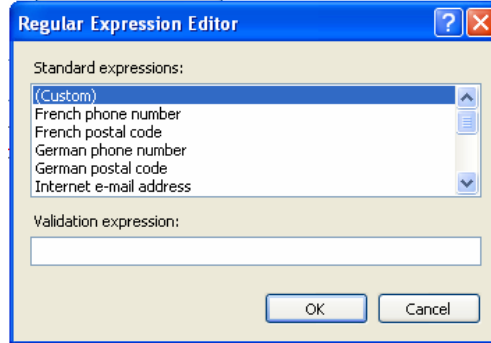
1.8.5. RegularExpressionValidator

Web sayfasında kullanıcı girişlerinin önceden tanımlanmış telefon numarası, posta kodu veya elektronik posta adresi gibi biçimlerle eşlenirliğini test eder.

ControlToValidate özelliği ile hangi kontrolün biçim eşlenirlik testi yapılacağını belirleriz.

ErrorMessage özelliği ile biçim eşlenirliği ihlal edildiğinde sayfada görünmesini istediğimiz mesaj yazılır.

ValidationExpression özelliği ile biçim Şekil 1.12'deki listeden seçilir.



Şekil 1.12: RegularExpression Editor

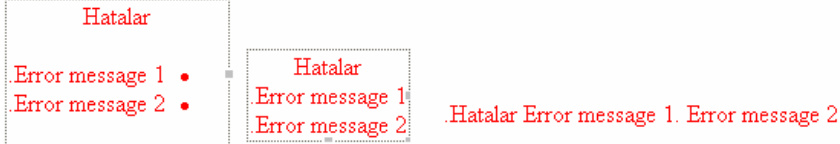
Eğer istenirse bu liste dışından istediğimiz formatı biz belirleyebiliriz. Örneğin T.C. kimlik numarası bildiğiniz gibi 11 rakamdan oluşur. Bunun için `\d{11}` yazarak bu TextBox'a 11 rakam girmeye zorlayabiliriz. Aynı işlemi biraz daha uzun yolla "|" karakterini (OR) kullanarak da

(0|1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)
(0|1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)
(0|1|2|3|4|5|6|7|8|9) yapabiliriz.

1.8.6. ValidationSummary Nesnesi

Hata mesajlarını işlemin sonunda hepsini beraber görüntüleyebileceğimiz bir kontroldür.

Hata mesajlarının görüntülenmesini istediğiniz yere ValidationSummary kontrolünü ekleyerek DisplayMode özelliğinden hataları Şekil 1.13'teki gibi görüntüleyebiliriz.



Şekil 1.13: Sırasıyla BulletList, List, SingleParagraph

HeaderText özelliği ile hatalarımıza bir başlık yazabiliriz.

ShowMessageBox özelliği ile (True/False) hataları mesaj kutusu ile görüntüleyebiliriz.

ShowSummary özelliği ile (True/False) sayfada görüntüleyebiliriz.

Yapacağımız örnekte bir kayıt formunda adı, soyadı, doğum tarihi ve doğum yeri bilgileri için birer label ve textbox ekliyoruz. Her textbox için bir adet RequiredFieldValidator kontrolü, doğum tarihi kontrolü için bir adet CompareValidator kontrolü, hataları bir arada gösterebilmek için ValidationSummary nesnesi ekliyoruz. TextBox özelliklerini,

Kontrol	ID
TextBox1	Ad
TextBox2	Soyad
TextBox3	Dtarihi
TextBox4	Dyeri

validator özelliklerini

Özellik	ControlTo Validate	ErrorMessage	Text	Operator	Type
RequiredField Validator1	Ad	Adı boş olmamalı	*		
RequiredField Validator2	Soyad	Soyadı boş olmamalı	*		
RequiredField Validator3	Dtarihi	Doğum Tarihi boş olmamalı	*		
RequiredField Validator4	Dyeri	Doğum Yeri boş olmamalı	*		
Compare Validator1	Dtarihi	Tarih formatı yanlış	*	DataTypeCheck	Date

şeklinde değiştiriyoruz.

1.9. “AutoPostBack ve EnableViewState” özellikleri

1.9.1. AutoPostBack

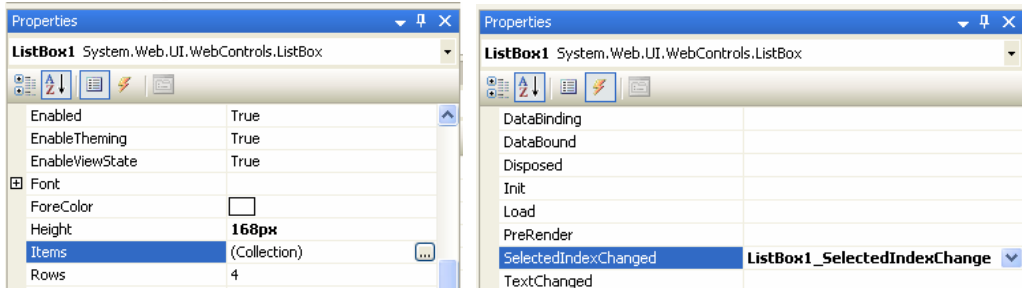
Bazen web sayfalarında, yapılan seçime göre yeni seçeneklerin gösterilmesi gerekir. Örneğin seçilen markanın sahip olduğu modellerin listelenmesi gibi. Bu işi yani kendi kendine tetiklenme işini kolaylaştıran özellik "AutoPostBack" özelliğidir.

Bazı kontrollerin kendilerine has olayları vardır. AutoPostBack özelliği aktif olan kontrolün olayı gerçekleştiğinde otomatik olarak hedefteki Sub'u (alt programı) tetikleyebilir. Ancak bu özelliği çok fazla kullanmak, server'a ek yük getirecektir. Çünkü tüm seçimlerde sayfa yeniden yüklenecektir. Bu nedenle AutoPostBack özelliği sadece gerektiğinde kullanılmalıdır.

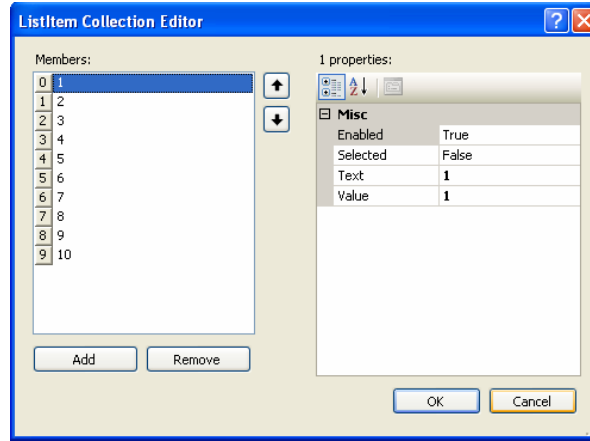
AutoPostBack özelliği ile ilgili yapacağımız örnekte Web formumuza eklediğimiz ListBox kontrolündeki sayılara tıklayarak, seçtiğimiz sayının karesini, küpünü ve faktöriyelini bulmaya çalışacağız.

Bu örnek için öncelikle formumuza bir adet ListBox ve üç adet Label kontrolü ekliyoruz. ListBox kontrolünün AutoPostBack özelliğini **True** olarak ayarlıyoruz. ListBox'a değer girebilmek için ListBox seçili iken Şekil 1.15'teki Properties (özellikler) penceresinden Items özelliğinin sağındaki ... işaretine tıklayarak Şekil 1.16'daki ListItem Collection Editor penceresine ulaşarak istediğimiz değerleri buradan veya Form üzerinde Source (Microsoft Visual Studio 2003 sürümünde HTML) tıklanarak kod yardımıyla girebiliriz.


```
<asp:ListBox ID="ListBox1" runat="server"
AutoPostBack="True" Style="z-index: 100;
left: 32px; position: absolute; top: 24px"
Height="168px" Width="32px">
<asp:ListItem Value="1"></asp:ListItem>
<asp:ListItem Value="2"></asp:ListItem>
<asp:ListItem Value="3"></asp:ListItem>
<asp:ListItem Value="4"></asp:ListItem>
<asp:ListItem Value="5"></asp:ListItem>
<asp:ListItem Value="6"></asp:ListItem>
<asp:ListItem Value="7"></asp:ListItem>
<asp:ListItem Value="8"></asp:ListItem>
<asp:ListItem Value="9"></asp:ListItem>
<asp:ListItem Value="10"></asp:ListItem>
</asp:ListBox>
```



Şekil 1.16: ListBox kontrolü özellikler ve olaylar penceresi

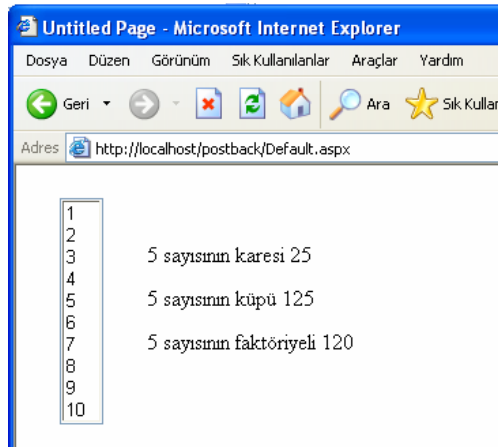


Şekil 1.17:ListItem Collection Editor

Tekrar properties penceresinde  Olaylar (Events) butonuna tıklayarak Şekil 1.15'te görülen SelectedIndexChanged olayına çift tıklayarak kendi olayına veya bu satıra yazacağımız isim ile oluşacak alt programın içeriğine aşağıdaki komutları yazabiliriz.

```
Protected Sub Hesapla(ByVal sender As Object, ByVal e As System.EventArgs) Handles ListBox1.SelectedIndexChanged
    Dim i, fakt As Integer
    Dim sayi As Integer = ListBox1.SelectedItem.Text
    ListBox1.SelectedIndex = -1
    fakt = 1
    For i = 1 To sayi
        fakt *= i
    Next
    Label1.Text = sayi & " sayısının faktöriyeli " & fakt
    Label2.Text = sayi & " sayısının karesi " & Math.Pow(sayi, 2)
    Label3.Text = sayi & " sayısının küpü " & Math.Pow(sayi, 3)
End Sub
```

Programın Web sayfasındaki görünümü Şekil 1.17'deki gibi olacaktır.



Şekil 1.17: AutoPostBack örneği ekran görüntüsü

1.9.2. EnableViewState

Web sayfalarında kullanıcıdan bir form doldurulması istendiğinde, eğer form doldurulduktan sonra hatalar oluşuyorsa kullanıcı forma geri döndüğünde girilen bilgilerin kaybolmaması için EnableViewState özelliğini kullanabiliriz.

EnableViewState özelliği kontrollerde veya direkt olarak @Page direktifi içerisinde kullanılabilir. @Page direktifi içinde kullanıldığında sayfa içerisindeki kontrollerin EnableViewState özelliklerinin değerinin bağlayıcılığı kalmaz ve direktifte ne dendiye o olur.

```
<%@ Page EnableViewState="true" %>
```

EnableViewState özelliğinin değeri "False" olarak belirlenmiş bir TextBox kontrolünün değeri PostBack sonrasında korunmaz. Ancak bu "korunmama" işlemi, listedeki seçili elemanın PostBack'den sonra da seçili olmasını sağlayamaz. EnableViewState özelliğinin değeri "True" olan bir ListBox'un elemanlarını page_load içerisinde doldurur ve buna bir "if not IsPostBack" kontrolü koyarsanız, PostBack sonunda liste boşalır.

Yapacağımız örnekte Web sayfamızdaki Butona tıklayınca EnableViewState özelliği False olan etiketimiz ekrana tekrar yazılmayacak.

```
<%@ Page Language="VB" Debug="True" %>
<script runat="server">
Sub page_load(Source As Object, E As EventArgs)
    If Not IsPostBack Then
        etiket1.Text = "Bu etiketin EnableViewState özelliği
False olarak ayarlandı"
        etiket2.Text = "Bu etiketin EnableViewState özelliği
False olarak ayarlandı"
    End If
End Sub
</script>
<form id="Form1" runat="server">
*
<asp:Label id="etiket1" runat="server"
    style="font: bold 15px Arial;" EnableViewState="False"
/>
<br>
*
<asp:Label id="etiket2" runat="server"
    style="font: bold 15px Arial;" EnableViewState="True"
/>
<br><br>
<asp:Button id="btnGonder" runat="server" Text="Yazıyı yoket"/>
</form>
```



Şekil 1.18: EnableViewState özelliği

Eğer programımızda @page direktifinde EnableViewState="False" yazsaydık kontrollerde belirttiğimiz değerler göz ardı edilecek ve etiketlerin ikisi de PostBack'den sonra görünmeyecek, EnableViewState="True" değerinde de kontrol değerleri ne olursa olsun iki etikette görünecekti.

1.10. Basit HTML Etiketleri (HTML, body, title, form...)

HTML etiketleri, '<' işaretiyle başlayan ve '>' işaretiyle biten özel sözcüklerdir. Bu sözcükler bir metin içinde genellikle iki şekilde yer alır. Bunların birincisi açılış etiketi, ikincisi de kapanış etiketidir. Açılış etiketi komutun etki edeceği sözcüğün (veya resim gibi bir nesnenin) öncesinde, kapanış etiketi de sonrasında yer alır. Kapanış etiketleri ayrıca "/" sembolünü de içerir. Bu sembol bulunmadığı zaman etiketin etkisi başladığı yerden metnin sonuna kadar devam eder. Etiketler tek başlarına kullanılabilecekleri gibi çeşitli özellikleri ile birlikte de kullanılabilir.

Bir sözcüğü etkilemek için birden fazla komut (etiket) kullanılabilir. Ancak sıralamanın nasıl yapıldığı önemlidir. Etkilemeyi yapan ilk etiket en sonda yer almalıdır. Yani bir etiket başka bir etiketi içerecek şekilde kullanılmalıdır ancak etiketler birbirleriyle kesişmemelidirler.

HTML etiketleri büyük veya küçük harflerle yazıldıklarında değişen bir şey olmaz. Tarayıcılar harf büyüklüğüne duyarlı değildir. Bu durum sadece programcının bir tercihidir. Örneğin, sitenin tasarımcısı komutları kolayca görebilmek için büyük harflerle yazmak isteyebilir. Burada komutlarda Türkçe'ye özgü olan karakterlerin kullanılmamasına dikkat edilmesi gerektiğidir. Örneğin <title> veya <TITLE> yerine <TİTLE> yazılması durumunda tarayıcı program komutu anlayamayacaktır.

1.10.1. HTML

HTML kodlarının yorum aralığını başlatmak için bu etiketi kullanırız. Sayfamızın head ve body bölümleri bu etiketin yorum aralığında tanımlıdır.

```
<html>
<head>head etiketinin içeriği</head>
<body>body etiketinin içeriği</body>
</html>
```

1.10.2. HEAD

Yorum alanında sayfamız hakkında bilgilerin bulunduğu bir etikettir. Örneğin, sayfamızın başlık bilgisinin bulunduğu TITLE etiketi ya da arama motorlarına bilgi sağlamak için oluşturulan META etiketi bu etiketin yorum alanında tanımlanır.

```
<html>
<head>
<title>Başlık Etiketleri</title>
</head>
</html>
```

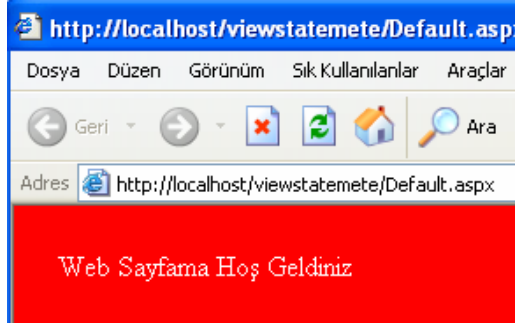
1.10.3. BODY

HTML etiketlerinden en kapsamlı içeriğe sahip etikettir. Web sayfasında izlenen tüm içerik bu etiket alanında tanımlanır.

Kod	Görevi
text = "renk"	Yazı rengini belirler
link = "renk"	Köprülerin rengini belirler
vlink = "renk"	Ziyaret edilmiş köprülerin rengini belirler
alink = "renk"	Aktif olan köprülerin rengini belirler
Bgcolor = "renk"	Arka plan rengini belirler
Background = "resim_dosyası"	Arka planda kullanılacak resim dosyasının adını belirler
Topmargin = değer	Üst kenar boşluğunu belirler
Leftmargin = değer	Sol kenar boşluğunu belirler
Rightmargin =değer	Sağ kenar boşluğunu belirler

```
<html>
<body topmargin=25 leftmargin=25 text=#ffffff bgcolor=#ff0000 >
Web Sayfama Hoş Geldiniz
</body></html>
```

Kodları yazarak Şekil 1.19'daki gibi kırmızı arka plan üzerine beyaz renkle ve üst ve sol kenarlardan 25 pixel boşluk bırakarak "Web Sayfama Hoş Geldiniz" yazacaktır.

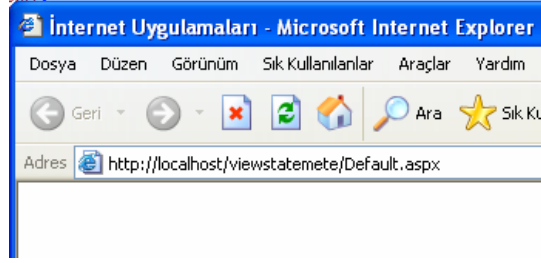


Şekil 1.19: BODY etiketi örneği

1.10.4. TITLE

Web sayfamızın başlık bilgisini içerir. Bu etiket head etiketi içinde kullanılmalıdır.

```
<html>
<head><title>İnternet Uygulamaları</title></head>
</html>
```



Şekil 1.20: TITLE etiketi örneği

1.10.5. FORM

Form etiketi ile Web sayfamızı, sadece bilgi verilen değil aynı zamanda kullanıcıdan bilgi alan dinamik bir sayfa haline hale getirebiliriz.

Form etiketini kullanarak sayfada oluşturabileceğimiz elemanlar şunlardır:

1.10.5.1. Düğmeler (Button)

3 tür düğme vardır. Tıklandığında formu gönderen gönderme düğmesi, tıklandığında formun içeriğini başlangıç değerlerine döndüren sıfırlama düğmesi ve önceden tanımlanmış bir davranışa sahip olmayan genel amaçlı düğmelerdir. Düğmeler ya doğrudan <button> etiketi kullanılarak ya da <input> etiketinde bir parametre olarak kullanılır. Belirtildikleri etiketin type parametresinde bu 3 tür düğme sırasıyla submit, reset, button değerleri verilerek kullanılır.

1.10.5.2. Onay Kutuları (CheckBox)

Kare, içine tik bırakılan kutucukları oluşturmak için kullanılır. <input> etiketinde bir parametre olarak kullanılır.

1.10.5.3. Radyo Düğmeleri (OptionButton)

Sayfamıza yuvarlak işaret kutucuğu bırakmak için kullanılır. Kullanılan kutucuğun onay kutusundan farkı, sadece bir seçenek işaretlenebilmesidir. <input> etiketinde bir parametre olarak type="radio" ve ek olarak value ve name parametreleriyle birlikte kullanılır.

1.10.5.4. Menüler

Kullanıcı bu menüden mevcut seçenekler arasından herhangi birini seçerek bilgi girişinde bulunur. Sayfamıza bu elemanı eklemek için <select> etiketini kullanırız.

1.10.5.5. Metin Girişleri

Metin girdilerini almak için iki tür eleman kullanılır. <input> etiketinde bir parametre olarak type="text" kullanılarak tek satırlık ve <textarea> etiketi ile oluşturulan çok satırlı metin girdi elemanlarıdır.

1.10.5.6. Dosya Seçim Elemanları

<input> etiketinde type="file" parametresi kullanılarak oluşturulan elemanlardır.

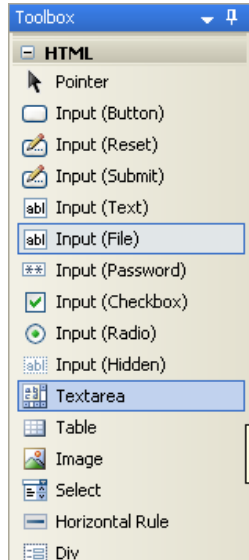
1.10.5.7. Gizli Elemanlar

<input> etiketinde type="hidden" parametresi kullanılarak oluşturulan elemanlardır.

1.10.5.8. Nesnelere

<object> etiketi kullanılarak oluşturulan elemanlardır.

Yukarıda belirtilen elemanları kod yardımıyla sayfamıza ekleyebileceğimiz gibi Microsoft Visual Studio ile Şekil 1.21'deki araç kutusundan HTML elemanlarını çok kolay bir şekilde sayfamıza ekleyebiliriz.



Şekil 1.21: HTML elemanları

1.10.6. Center Etiketi

Etiket aralığında bulunan yazıların tarayıcı programında ortalayarak yazılacağını belirtir.

1.10.7. Başlık Etiketleri

Sayfamıza başlık eklemek istediğimizde <h1> ile h<6> aralığındaki başlık etiketleri seçilir.

```
<html>
<head>
<title>Başlık Etiketleri</title>
</head>
<body>
<h1>Başlık 1</h1>
<h2>Başlık 2</h2>
<h3>Başlık 3</h3>
<h4>Başlık 4</h4>
<h5>Başlık 5</h5>
<h6>Başlık 6</h6></body>
</html>
```



Şekil 1.22: Başlık etiketleri

1.10.8. , <i>, <s>, <u>, <big>, <small>, <tt> Etiketleri

Yazı biçimini belirler.

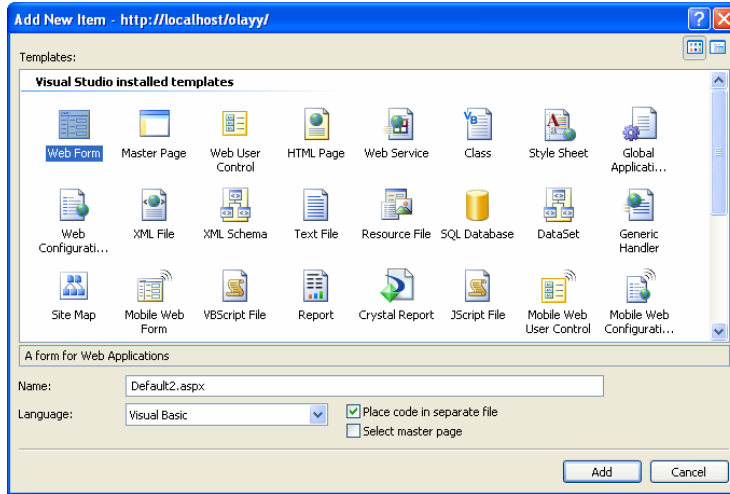
```
<body>
<b>kalın</b>, <i>eğik</i>, <s>üzeri çizili</s>, <u>altı çizili</u>,
<small>küçük</small>, <big>büyük</big>, <tt>daktilo yazısı</tt>
</body>
```

kalın, *eğik*, üzeri çizili, altı çizili, küçük, büyük, daktilo yazısı

1.11. “Set As Start Page” Seçeneği

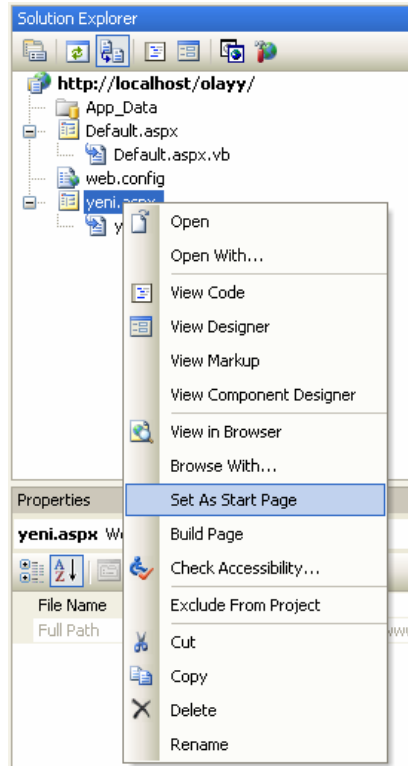
Projemizde birden fazla Web formumuz varsa hangi Web formun ilk olarak yükleneceğine **Set As Start Page** komutu ile karar veririz.

Projemize yeni bir Web formu eklemek için Solution Explorer penceresinden veya Web Site menüsünden Add New Item komutuyla Şekil 1.23'teki pencereye ulaşarak ikinci formu projemize ekleyebiliriz.



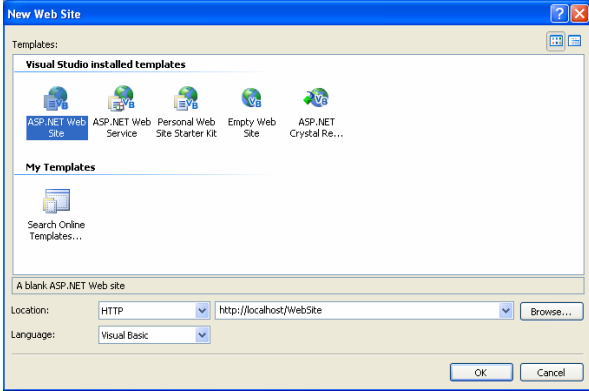
Şekil 1.23: Add New Item

Artık Solution Explorer penceresinde görülen iki Web formumuzdan hangisini istersek üzerine sağ tıklayarak Set As Start Page komutunu vererek proje çalıştığında ilk çalışacak sayfa olarak ayarlayabiliriz.



Şekil 1.24: Set As Start Page

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<ul style="list-style-type: none">➤ Yeni proje açma penceresinden “ASP.NET Web Application” seçiniz.  <ul style="list-style-type: none">➤ Form tasarım penceresinde gerekli nesnelere ekleyiniz.➤ Sayfanın başlık (Title), hızlı ulaşım tuşları (AccessKeys), tab sırası (Tab Index) ve fare ipuçlarını (ToolTip) ayarlayınız.➤ Girilen bilgilerin doğruluğunu “Validation Controls” ile denetleyiniz.➤ Programı test etmek için çalıştırarak, tarayıcıda sonucunu görünüz.	<ul style="list-style-type: none">➤ ASP.NET ile ASP arasındaki farkları araştırınız.➤ Birkaç Web sitesini inceleyerek kullanılan nesnelere, sayfa başlıklarına, Sekme sırasına ve nesnelere üzerinde durulduğunda görüntülenen yazılara dikkat ediniz.➤ Kayıt olunmasını isteyen birkaç siteyi inceleyiniz.

ÖLÇME VE DEĞERLENDİRME

A.OBJEKTİF TESTLER (ÖLÇME SORULARI)

Aşağıdaki çoktan seçmeli soruları cevaplayınız.

1. Web sayfasının başlığı aşağıdaki özelliklerden hangisinde belirtilir?
A) Title
B) Div
C) Form
D) BgColor
2. ToolTip özelliğinin görevi aşağıdakilerden hangisidir?
A) Sekme sırası
B) Kısayol tuşu
C) Açıklama bilgisi
D) Hızlı erişim
3. Aşağıdaki kontrollerden hangisi hata özetlerini verir?
A) Range Validator
B) Compare Validator
C) Validation Summary
D) Custom Validator
4. IIS kurulu bir bilgisayarda Web sunucu tarafı çalıştırılacak bir web sayfası nerede saklanır?
A) wwwroot klasörü
B) Belgelerim\Project klasörü
C) Internet Information Service klasörü
D) İstenilen herhangi bir klasörde
5. Set As StartPage seçeneğinin görevi aşağıdakilerden hangisidir?
A) Visual Studio ilk açıldığında yüklenen sayfadır.
B) Sayfa yüklenirken gerçekleşen olaydır.
C) Visual Studio her açıldığında ekrana gelmesi istenen sayfadır.
D) Birden fazla Web Form var ise çalıştırıldığında hangisi ile başlayacağını belirtir.

DEĞERLENDİRME

Sorulara verdiğiniz cevapları modül sonundaki cevap anahtarıyla karşılaştırınız. Bu faaliyet kapsamında hangi bilgileri kazandığınızı belirleyiniz. Yanlış cevaplandığınız sorularla ilgili konuları tekrar inceleyip öğrenmeye çalışınız. Kaynak ve yardımcı ders kitaplarından faydalanınız. Kütüphanelerden, internetten veya görsel programlar yazan kişilerden araştırma yapınız. Öğrenme faaliyetiyle ilgili olarak öğretmenlerinizden yardım alınız.

ÖĞRENME FAALİYETİ-2

AMAÇ

Gerekli ortam sağlandığında veri tabanı uygulamalarını Web sayfalarında da çalıştırabilecek, Web sayfasında veri bağlı nesnelere çalışabileceklerdir.

ARAŞTIRMA

Bu faaliyet öncesinde yapmanız gereken öncelikli araştırmalar şunlardır:

ASP.NET ile gelen yeniliklerden veritabanı ile ilgilileri araştırınız.

Araştırma işlemleri için internet ortamını kullanınız.

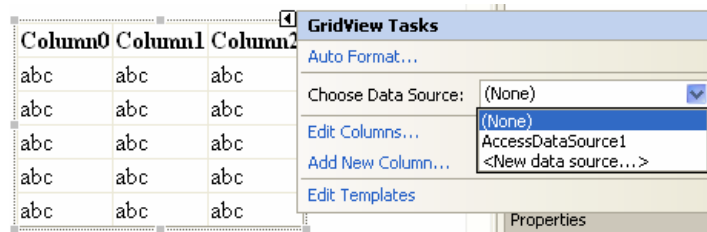
2. “ADO.NET”İ WEB UYGULAMASINDA KULLANMA

2.1. DataGrid Nesneleri

Veri kaynağından alınan bilgiyi, belirlenen bir formatta görüntüleyen bir kontroldür. DataGrid kontrolü, veri kaynağındaki alanları bir tablonun sütunları, kayıtlar ise tablonun satırları olarak gösterilir.

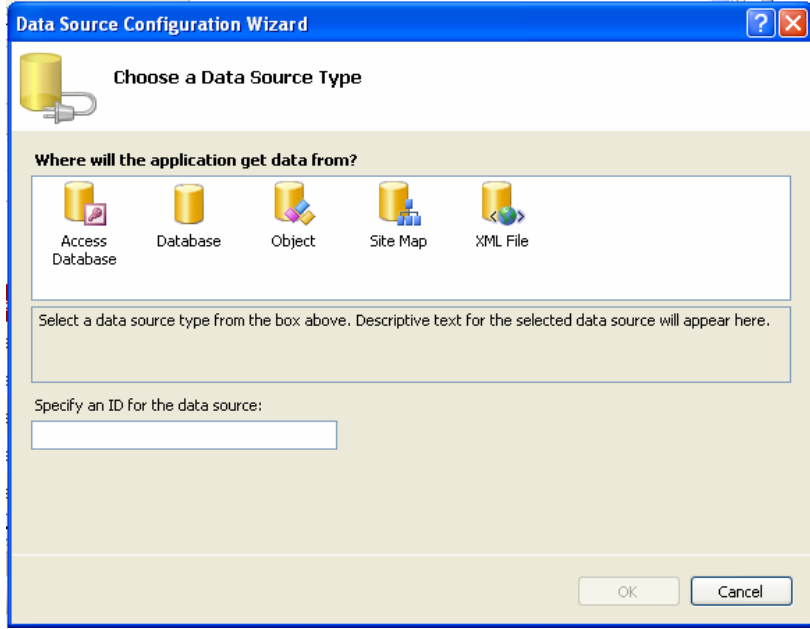
DataGrid, sayfalama, sıralama, seçme, düzenleme ve silme işlemlerine de destek verir.

Araç kutusunun Data başlığından GridView kontrolünü sürükleyerek formumuza eklediğimizde Şekil 2.1'deki görüntüyü elde ederiz.



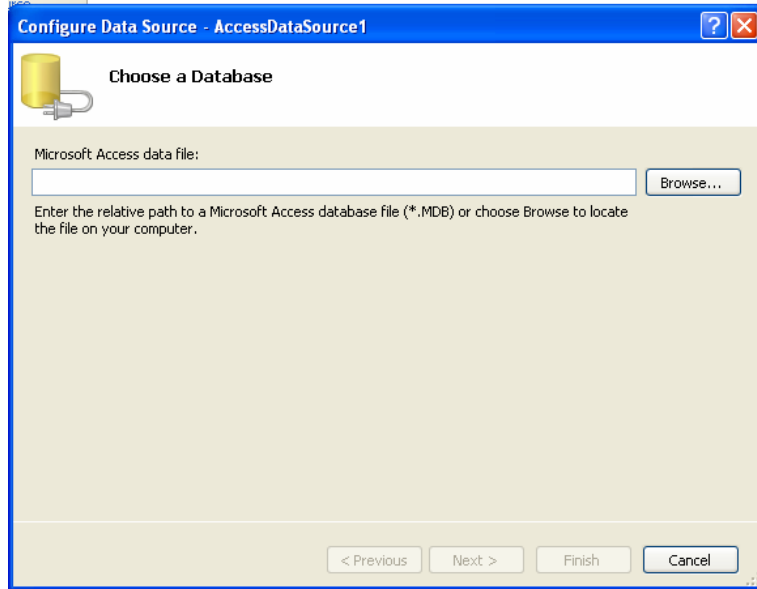
Şekil 2.1: GridView

Önceden formumuzda bir DataSource oluşturmuş ise Choose Data Source seçeneğinden seçeriz. Oluşturmamış ise New Data Source seçeneğini seçerek Şekil 2.2'deki Data Source Configuration Wizard programını başlatırız.

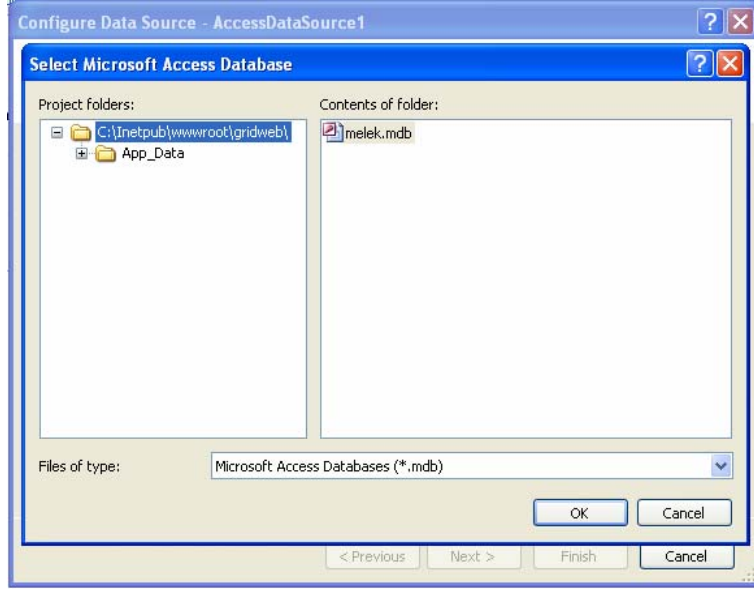


Şekil 2.2: Data Source Configuration Wizard

Kullanacağımız veri türünü seçerek OK tıklanıp işleme devam edilir. Ben Access veri tabanı seçerek devam ettim. Sonraki pencremizde seçtiğimiz veri tabanı türündeki dosyamızı seçeceğiz. Şekil 2.3'ten Browse butonuna tıklayarak Şekil 2.4 penceresine ulaşabiliriz.

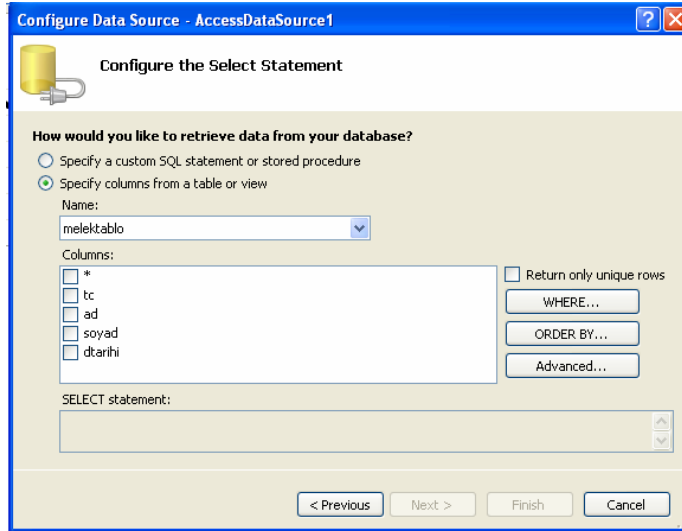


Şekil 2.3: Configure Data Source



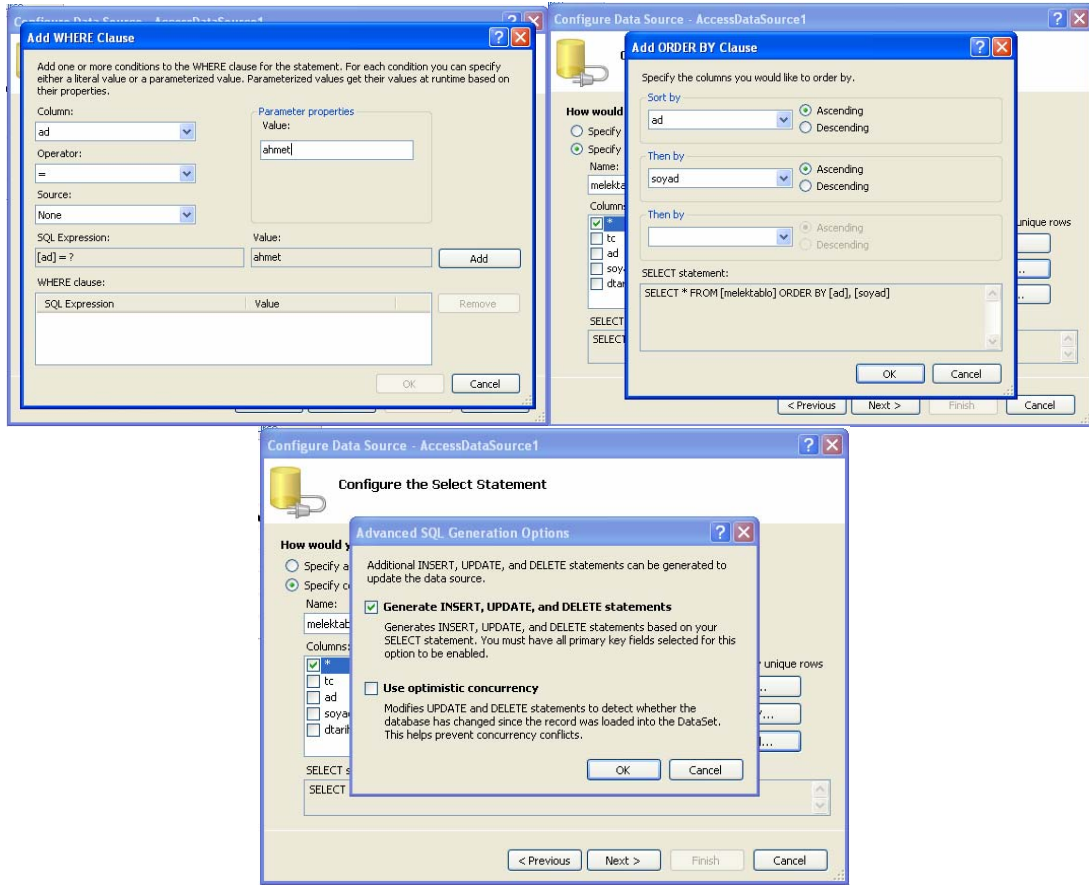
Şekil 2.4: Select Microsoft Access Database

Burada veri tabanı dosyamızı wwwroot içerisindeki projemizin klasörüne kopyalamamız gerekmektedir. Veri tabanı dosyasını seçerek OK butonuna tıklayarak, dosya adı belirtilen pencerede Next butonuna tıklayarak Şekil 2.5'deki veri tabanımızdan hangi alanları seçeceğimize karar verdiğimiz pencereye ulaşabiliriz.



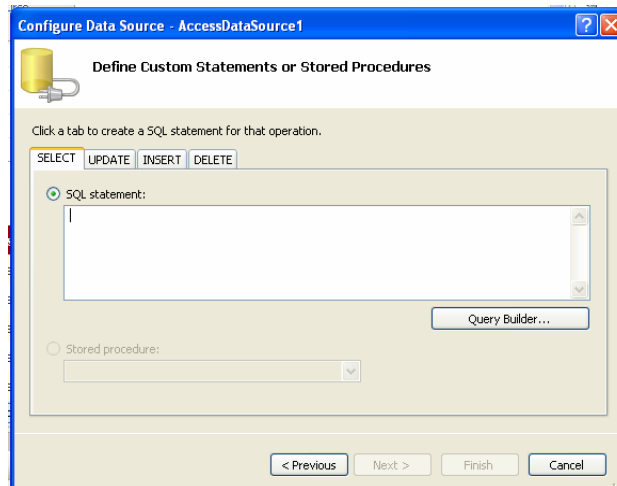
Şekil 2.5: Configure the Select Statment

Burada Specify columns from a table or view seçili iken istediğimiz sütunları işaretleyerek ve WHERE... seçeneğiyle arama, ORDER BY... seçeneğiyle sıralama ve Advanced seçeneği ile ekleme, düzenleme ve silme işlemlerini düzenleyebiliriz.



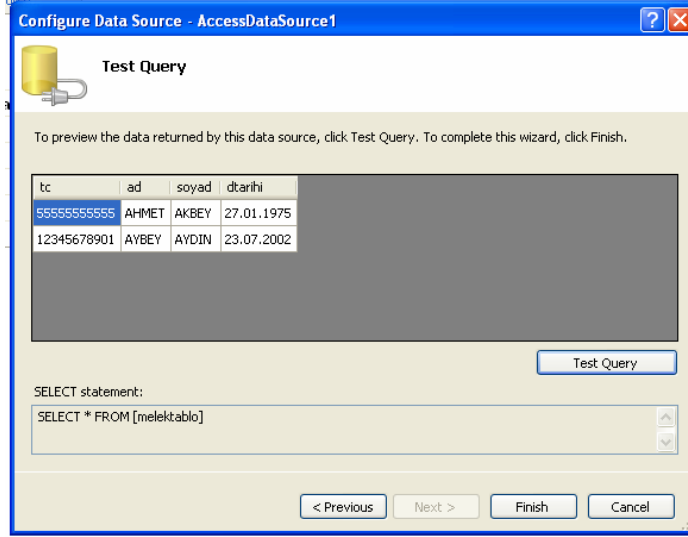
Şekil 2.6: WHERE, ORDER BY ve Advanced

Specify a custom SQL statement or stored procedure seçili iken SQL komutlarını girebileceğimiz pencere veya Query Builder ile oluşturabileceğimiz pencereye ulaşırız.



Şekil 2.7: Specify a custom SQL statement or stored procedure

Son olarak Şekil 2.8'den Test Query ile sorgumuzun sonuçlarını gördükten sonra Finish butonu ile DataSource ve DataGrid kullanıma hazırdır.



Şekil 2.8: Test Query

2.2. Kod Penceresinde Nesnelere Veriye Bağlama

Web Form Designer penceresini kullanarak DataGrid veya benzeri nesnelere veriye bağlamayı önceki konuda gördük. Şimdi nesnelere kod penceresinde oluşturularak veriye bağlamaya çalışacağız.

DataGrid oluşturmak için `<asp:DataGrid id="DataGrid1" runat="server"/>` kodunu kullanırız. Bu kod ile server html tablosu oluşturularak, her kayıt bir satıra ayarlanır. Veri kaynağındaki alan isimleri de sütun başlığı olarak kullanılır.

Sayfanın başında, ADO.NET kullanacağımız için System.Data ve System.Data.OleDb adalarını kodumuza ekledik.

Page_Load olayı içerisinde sql cümlesini oluşturduk. New() yapılandırıcısı (constructor) ile conn isiminde bir OleDbConnection nesnesi oluşturduk ve bağlantı cümlesini parametre olarak kullandık. Ardından yine New() yapılandırıcısı ile adapter isiminde bir OleDbDataAdapter nesnesi oluştururken sql cümlesini ve az önce oluşturduğumuz OleDbConnection nesnesini parametre olarak kullandık. Bu işlemten sonra New() yapılandırıcısı ile bir DataSet nesnesi oluşturup, OleDbDataAdapter nesnesinin Fill metodunu kullanarak bu DataSet nesnesindeki ismini kendimiz belirlediğimiz "melektablo" isimli tabloya sorgu sonucu dönen veriyi doldurduk. Veri kaynağındaki tablo ismi ile DataSet nesnesindeki tablonun isminin aynı olması gerekmez.

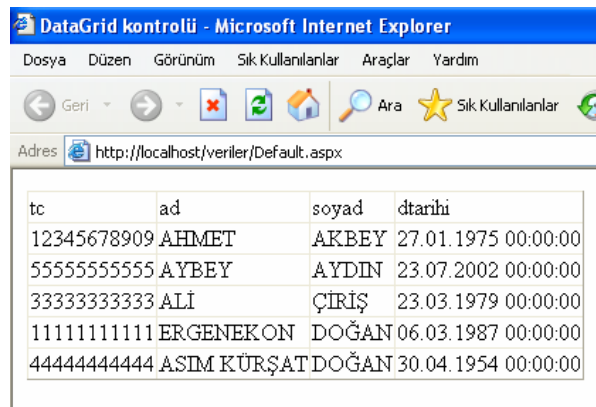
Son olarak, mydatagrid isimli DataGrid kontrolünün DataSource özelliğine, DataSet nesnesinin "melektablo" isimli tablosunu gösterdik. Ardından yine DataGrid kontrolünün

DataBind() metodunu çalıştırıp, Page_Load sub 'ını kapatmadan OleDbConnection nesnemizin close() metodu ile veri tabanı bağlantısını sonlandırdık.

```
<%@ Page Language="VB" Debug="True" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.OleDb" %>

<html><head><title> DataGrid kontrolü </title></head><body>
<script runat="server">
Sub Page_Load(Source As Object, E As EventArgs)
If Not IsPostBack Then
    Dim connstr, sql As String
        connstr = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source="
+ Server.MapPath("melek.mdb") + ""
        sql = "SELECT * FROM melektablo"
    Dim conn As New OleDbConnection(connstr)
    conn.open()

    Dim adapter As New OleDbDataAdapter(sql,conn)
    Dim ds As New DataSet()
        adapter.Fill(ds, "melektablo")
        mydatagrid.DataSource = ds.Tables("melektablo")
        mydatagrid.DataBind()
    conn.close()
End If
End Sub
</script>
<asp:DataGrid id="mydatagrid" runat="server"/>
</body>
</html>
```



tc	ad	soyad	dtarihi
12345678909	AHMET	AKBEY	27.01.1975 00:00:00
55555555555	AYBEY	AYDIN	23.07.2002 00:00:00
33333333333	ALİ	ÇİRİŞ	23.03.1979 00:00:00
11111111111	ERGENEKON	DOĞAN	06.03.1987 00:00:00
44444444444	ASIM KÜRŞAT	DOĞAN	30.04.1954 00:00:00

Şekil 2.9: Kod penceresinden DataGrid'e bağlanan veriler

2.3. DataBind komutu

Kontrolle veri bağlamak için kullanılan fonksiyondur. Önceki örneğe dönecek olursak

```
Dim adapter As New OleDbDataAdapter(sql, conn)
Dim ds As New DataSet()
    adapter.Fill(ds, "melektablo")
    mydatagrid.DataSource = ds.Tables("melektablo")
    mydatagrid.DataBind()
conn.close()
```

tanımlanan ds DataSeti veritabanımızın melektablo tablosu ile doldurulduktan sonra DataGrid kontrolümüzün DataSource özelliğine tanıtılıyor. Bu tanıtma işleminden sonra DataBind fonksiyonu ile DataGrid'i DataSet'e bağlamış oluyoruz. Aynı işlemi veri tabanı olmadan dizileri kullanarak yapabiliriz.

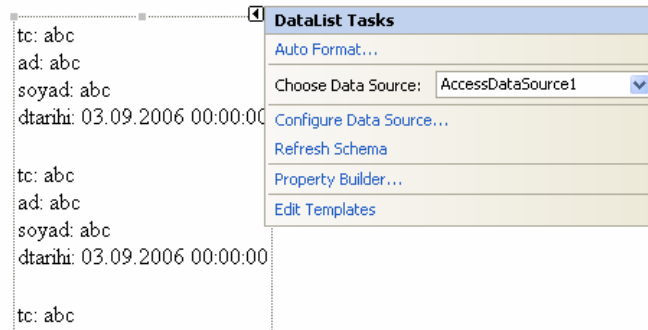
2.4. DataBindings Diyalog Kutusu

Kontrol için özelliklerin listesini görüntüler ve veri kaynağı bağlantı özelliklerini belirtir.

DataBindings diyalog kutusu, Repeater, DataList, DataGrid, ListBox, CheckBoxList ve RadioButtonList kontrolleri gibi çoklu kayıt gösteren kontrollerde kullanılmaz.

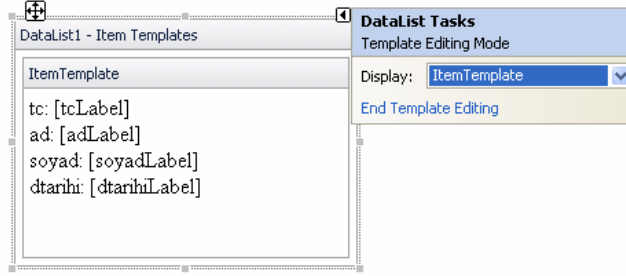
DataBindings diyalog kutusunu açmak için

1. aspx uzantılı sayfamızı dizayn görünümünde açın. (Design View)
2. Araç kutusunun (Toolbox) Data sekmesinden DataList veya FormView kontrolünü sayfaya ekleyin.
3. Kontrolle sağ tıkla ve Show Smart Tag'a tıklayarak kontrol için görevler penceresini aç.
4. Kontrolle DataSource ayarla. Kontrolle DataSource ayarlamadan DataBindings diyalog kutusu açılmayacaktır.



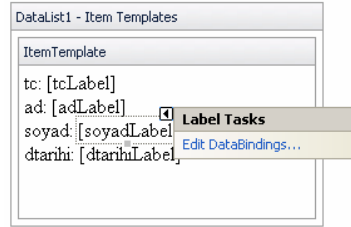
Şekil 2.10: Görevler penceresi

5. Kontrolün Tasks (görevler) penceresinden Edit Templates seçeneğine tıklayınız.



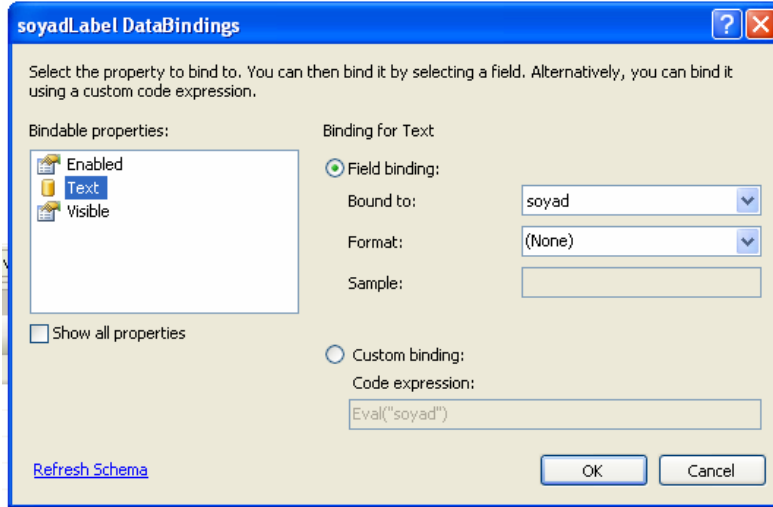
Şekil 2.11: Edit Templates

6. Açılır listeden Item Template seçilir.



Şekil 2.12: Edit DataBindings

7. Item Template alanındaki etiketlerin Label Tasks (görevler) Edit DataBindings e tıklayarak DataBindings Dialog Box açılmış olur.



Şekil 2.13: DataBindings Dialog Box

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<ul style="list-style-type: none">➤ Bir Web uygulamasına veriye bağlanacak nesnelere ekleyiniz.➤ Bir metin kutusunun “DataBindings” özelliğinden metin özelliğine “Simple Binding” ile veri ekleyiniz.➤ “Load” prosedürüne veri bağlantıları kodlarını yazınız	<ul style="list-style-type: none">➤ .NET ile gelen veri tabanı yeniliklerini inceleyiniz.➤ Birkaç Web sitesini inceleyerek kullanılan kontrollerin veri bağlantılarını inceleyiniz.➤ Kayıt olunmasını isteyen birkaç siteyi inceleyiniz.

ÖLÇME VE DEĞERLENDİRME

OBJEKTİF TESTLER (ÖLÇME SORULARI)

Aşağıdaki çoktan seçmeli soruları cevaplayınız.

1. Aşağıdakilerden hangisi bir DataSource tipi değildir?
A) XML dosyası
B) Access Veri Tabanı
C) Site Map
D) XLS dosyası
2. DataSource konfigürasyonunda Select Statement penceresinde ORDER BY.. Butonunun görevi nedir?
A) Verilerin seçilmesini sağlar.
B) Verilerin sıralanmasını sağlar.
C) Verilerin güncellenebilmesini sağlar.
D) Verilerin silinebilmesini sağlar.
3. Kontrole veri bağlamak için kullanılan fonksiyon aşağıdakilerden hangisidir?
A) DataBind
B) DataSource
C) DataAdapter
D) DataTables
4. Aşağıdaki kontrollerin hangisinde DataBindingsDialogBox kullanılabilir?
A) CheckBoxList
B) Label
C) DataList
D) DataGrid
5. Aşağıda verilen komut satırının görevi nedir?

```
Dim adapter As New OleDbDataAdapter(sql, conn)
Dim ds As New DataSet()
adapter.Fill(ds, "melektablo")
```

DEĞERLENDİRME

Sorulara verdiğiniz cevapları modül sonundaki cevap anahtarıyla karşılaştırınız. Bu faaliyet kapsamında hangi bilgileri kazandığınızı belirleyiniz. Yanlış cevaplandığınız sorularla ilgili konuları tekrar inceleyip öğrenmeye çalışınız. Kaynak ve yardımcı ders kitaplarından faydalanınız. Kütüphanelerden, internetten veya görsel programlar yazan kişilerden araştırma yapınız. Öğrenme faaliyetiyle ilgili olarak öğretmenlerinizden yardım alınız.

ÖĞRENME FAALİYETİ-3

AMAÇ

Gerekli ortam sağlandığında Web servis oluşturabilecek, var olan bir Web servisi Web sayfasında veya uygulamada çalışabileceksiniz.

ARAŞTIRMA

Bu faaliyet öncesinde yapmanız gereken öncelikli araştırmalar şunlardır:

Ülkemizde kullanılan birkaç Web servisi inceleyiniz. (örnek T.C. Kimlik Numarası) Web servis ile Web sayfası arasındaki farkı araştırınız.

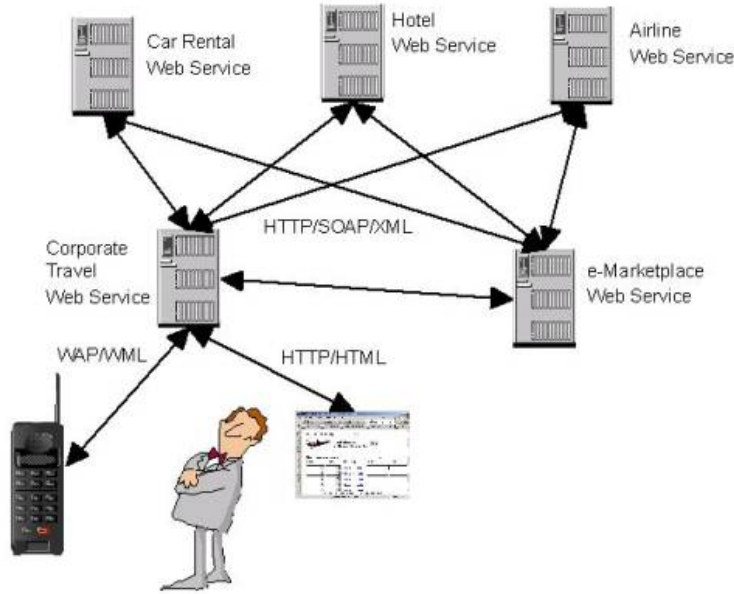
Araştırma işlemleri için internet ortamını kullanınız.

3. WEB SERVİSLERİ

3.1. “Distributed Web Applications” Deyimi

Yazılan programların, geliştirilen protokollerin, hazırlanan Web sitelerinin kısaca bilgisayar ile ilgili yapılan her işlemin amacı, bilgiye daha hızlı ve daha kolay yoldan ulaşmak içindir. Gelişmiş uygulamaların ortak problemi farklı bilgisayarların, farklı ağların ve farklı programların aralarında bilgi alışverişi sırasında yaşadığı engellerdir. XML'in çıkışıyla birlikte bu farklı sistemlerin birbiriyle haberleşmesi büyük bir ölçüde rahatlamıştır. Artık üzerinde çalıştıkları işletim sistemi, kullandıkları veri tabanı, yazılmış oldukları programlama dili fark etmeksizin bir XML dosyasını okuma ve ayrıştırma becerisine sahip her ortam birbiriyle rahatlıkla iletişim kurabilir hale gelmiştir.

1998 yılında Sun ve IBM'in ortak çalışmaları sonucu ortaya çıkardıkları Web Servisleri, farklı ortamların birbiriyle kolayca haberleşmesi konusunda önemli bir adım oldu. Fakat gerek Web Servislerinin oluşturulması ve kullanılmasının ileri derece programlama bilgisi gerektirdiği gerekse pazarlama ve tanıtımına gerektiği önemin verilmemesinden dolayı pek kullanılmadı. Web Servisleri için ikinci önemli adım ise Microsoft'un Visual Studio .NET geliştirme ortamı ile kolayca hazırlanıp kullanılabilir hale getirilmesidir.



Şekil 3.1: Web servisleri ile etkileşim sağlayan iş gezisi uygulaması (Kaynak : IBM)

İşletmelerin diğer işletmelerle olan iş süreçlerini bütünleştirme gereksinimi sonucunda ortaya çıkan ve gelişmekte olan yeni yapının temel taşı Web servisleridir. Web servislerindeki temel amaç işletme bilgi sistemlerindeki program modüllerinin etkileşimini sağlamaktır. Web servisleri Web ortamında yayınlanabilen, aranıp bulunabilen ve çağrılarak erişilebilen modüler uygulama fonksiyonlarıdır. Şekil 3.1 Web servislerini kullanarak bir iş gezisi için gerekli rezervasyonları yapan bir uygulama örneği senaryosunu göstermektedir. Bu örnekte iş gezisi planlayan kişi hava yolu, otel ve araç kiralama rezervasyonları yapacağı var sayılmıştır. İş gezisi organizasyonu için geliştirilen uygulama havayolu, otel, araç kiralama ve e-marketplace firmalarının sağladığı Web servislerini çağırarak gerekli işlemleri tek bir uygulama ile gerçekleştirecektir.

Web Servis nesne ve metotlarını HTTP protokolünü kullanarak bir masaüstü uygulaması ya da bir Web sitesine dahil edebiliriz. HTTP (Hyper Text Transfer Protocol) metin bazlı verilerin taşınmasına olanak sağlar. Web Servislerinde kullandığımız diğer veri tiplerinin (integer, string, array, dataset, object) transferi ise SOAP (Simple Object Access Protocol) sayesinde gerçekleşir.

3.2. SOAP (Simple Object Access Protocol)

SOAP (Simple Object Access Protocol), HTTP üzerinde çalışan fakat bilgiyi bir araya toplamak için XML benzeri özel bir format kullanan protokoldür. SOAP protokolünü kullandığımızda, mesajlarımızdaki bilgileri kodlamak için SOAP standardını kullanırız. SOAP mesajları XML e çok benzeyen bir yapıdadır.

Örnek SOAP mesajı:

İstek

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <HelloWorld xmlns="http://tempuri.org/" />
  </soap:Body>
</soap:Envelope>
```

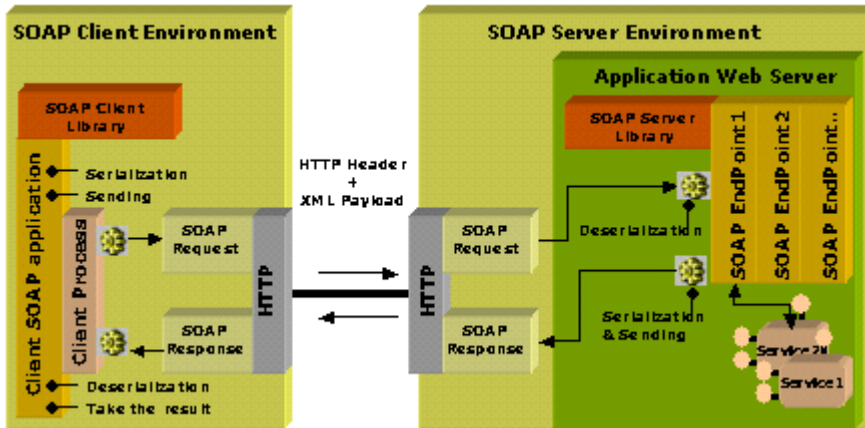
Yanıt

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <HelloWorldResponse xmlns="http://tempuri.org/">
      <HelloWorldResult>string</HelloWorldResult>
    </HelloWorldResponse>
  </soap:Body>
```

</soap:Envelope>

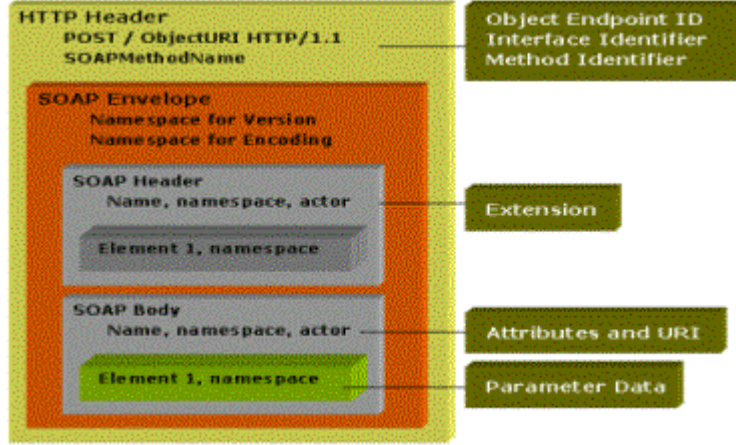
Yukarıdaki SOAP örneğine baktığımızda kök ögenin <soap:Envelope> olduğunu ve istemin <soap:Body> kısmını içerdiğini görürüz.

Şekil 3.2 bir SOAP istemci ve sunucusu arasındaki iletişimi göstermektedir. Bir SOAP uygulaması geliştirmek için istemci ve sunucuya SOAP geliştirme araçları ile birlikte gelen kütüphanelerin yüklenmesi gerekir. Bu kütüphaneler bir XML ayrıştırıcı ve SOAP işlemcisi içerir. İstemci SOAP uygulaması bir SOAP istek mesajı oluşturarak bu isteği SOAP sunucusunda tanımlanmış servis uç noktalarından (end point) birisi tarafından çalıştırılması için gönderir. SOAP sunucu ilgili servisi çalıştırdıktan sonra SOAP yanıt mesajı hazırlar. Hazırlanan SOAP yanıt mesajı istemciye iletilir.



Şekil 3.2: SOAP istemci ve SOAP sunucusu arasındaki iletişim

Şekil 3.3 HTTP protokolü ile gönderilen bir SOAP mesajını göstermektedir. SOAP mesajı HTTP POST metodu veri paketinin içinde gönderilir. Bir SOAP mesajı bir SOAP zarfından (SOAP envelope) oluşur. SOAP zarfı opsiyonel bir SOAP başlığı (SOAP header) ve SOAP gövdesinden (SOAP body) oluşur. SOAP gövdesi çağırılacak metod ve metodun içerdiği parametreleri içerir.



Şekil 3.3: Bir SOAP mesajının yapısı (Kaynak: www.techmetrix.com)

3.3. WSDL (Web Services Description Language)

WSDL (Web Services Description Language-Web Servisleri Tanımlama Dili) yazdığımız ve kullandığımız Web servislerinde parametreler, dönüş değerleri ve internet üzerinde hangi iletim protokolünü kullanacağı gibi ayrıntılar ile beraber bir istemcinin Web servisi ile etkileşimini belirleyen XML tabanlı bir standart dosyasıdır.

WSDL dokümanı Web servisi ile istemci arasındaki iletişim için gereken bilgileri içerir. Yazdığımız kod veya Web servisinin uyguladığı metotlar ile ilgili bilgi içermez. Bir asmx dosyasından WSDL dokümanına ulaşmak için adres satırına "?WSDL" eklemeniz yeterlidir.

Bir uygulamanın bir Web servisini kullanması için, Web servisinin nasıl çağırılacağını, ara yüzünün, hangi protokollerin ve kodlama standartlarının kullanılacağını belirtmesi gerekir. WSDL web servisini tanımlayan bir XML belgesidir. Web servisi tanımlama işlemler, giren ve çıkan mesaj formatları, ağ ve port adresleri gibi bilgileri tanımlar.

Örnek bir WSDL belgesi

```
<?xml version="1.0" encoding="utf-8" ?>
<wSDL:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://tempuri.org/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://tempuri.org/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/">
  <wSDL:types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://tempuri.org/">
      <s:element name="HelloWorld">
        <s:complexType />
      </s:element>
      <s:element name="HelloWorldResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="HelloWorldResult"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wSDL:types>
  <wSDL:message name="HelloWorldSoapIn">
    <wSDL:part name="parameters" element="tns:HelloWorld" />
  </wSDL:message>
  <wSDL:message name="HelloWorldSoapOut">
    <wSDL:part name="parameters" element="tns:HelloWorldResponse" />
  </wSDL:message>
  <wSDL:portType name="ServiceSoap">
    <wSDL:operation name="HelloWorld">
      <wSDL:input message="tns:HelloWorldSoapIn" />
      <wSDL:output message="tns:HelloWorldSoapOut" />
    </wSDL:operation>
  </wSDL:portType>
  <wSDL:binding name="ServiceSoap" type="tns:ServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  </wSDL:binding>
  <wSDL:operation name="HelloWorld">
    <soap:operation soapAction="http://tempuri.org/HelloWorld" style="document"
/>
  </wSDL:operation>
  <wSDL:input>
    <soap:body use="literal" />
  </wSDL:input>
  <wSDL:output>
    <soap:body use="literal" />
  </wSDL:output>
</wSDL:definitions>
```

```

</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:binding name="ServiceSoap12" type="tns:ServiceSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="HelloWorld">
  <soap12:operation soapAction="http://tempuri.org/HelloWorld"
style="document" />
- <wsdl:input>
  <soap12:body use="literal" />
</wsdl:input>
- <wsdl:output>
  <soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:service name="Service">
- <wsdl:port name="ServiceSoap" binding="tns:ServiceSoap">
  <soap:address location="http://localhost/Webservis/Service.asmx" />
</wsdl:port>
- <wsdl:port name="ServiceSoap12" binding="tns:ServiceSoap12">
  <soap12:address location="http://localhost/Webservis/Service.asmx" />
</wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Bir Web servisi tanım belgesi aşağıdaki temel elemanları içerir:

Types	: Mesajlarda kullanılacak veri tiplerini belirtir.
Message	: İletişimde kullanılacak mesajları tanımlar.
PortType	: Web servisinin içerdiği işlemleri (methods) ve ilgili mesajları tanımlar.
Binding	: İşlem ve mesajlarda kullanılacak veri formatlarını tanımlar.
Port	: Binding ve web adresinden oluşan servis noktasını tanımlar. Web adresi servisin çalıştırılacağı URL'dir.
Service	: Kullanılan port'lar kümesidir.

3.4. “*.asmx ve disco” Uzantısı

asmx uzantılı dosyalar, Web servisi kodlarını içeren dosyalardır. asmx dosyası yapı itibari ile aspx dosyalarına benzerler. Ortak özellikleri:

Bir page direktifi ile başlar.

Performansı yüksek tutmak amacıyla ilk çağrıldığında derlenir.

Not defteri veya herhangi bir text editörde düzenlenebilir.

.asmx isminde geçen "m" harfi "method" anlamındadır. Bir Web servisi, asıl işlevi sağlayan bir veya birden fazla Web method'undan oluşur.

Kullanıcılar, WSDL kullanan ve SOAP paketleri halinde bilgi ileten bir Web servisinin sunulduğu url sini biliyorlarsa bu url adresine ?WSDL ekleyerek WSDL dokümanına ve genel tüm özelliklerine ulaşabilirler. Bu sadece web servisini belli bir kurumun standardında çalıştırmada işe yarayacaktır. Ancak web servislerinin sayısı arttıkça ve web servisleri web üzerinde iş takibi açısından genel bir dil haline geldiğinde yukarıdaki olasılık pratik olmaktan çıkacaktır. Web servislerinin nerede bulunduğunu belirten dosyalar .disco uzantılı dosyalardır. Visual Studio bu dosyayı okuyup karşılık gelen web servislerini listeleyebilir. Disco dosyasının yapısı şöyledir:

```
<disco:discovery xmlns:disco="http://schemas.xmlsoap.org/disco"
xmlns:wSDL="http://schemas.xmlsoap.org/disco/wSDL">
<wSDL:contractRef
ref="http://localhost/sellili/service.asmx?WSDL"/>
</disco:discovery>
```

Disco dosyasının avantajları ise sadece Web servisleri tarafından kullanılır ve Web servislerinin tamamı için tek bir disco dosyasına istediğiniz kadar <disco> ögesi ekleyebilirsiniz.

Bir dinamik keşif dosyası oluşturduğunuzda dosyanın keşif faaliyeti sonucunda elinizin altında güncel tüm Web servislerinin bir listesini bulabileceğinizden emin olabilirsiniz. Bir <dynamicDiscovery> ögesi, belli bir dizin altındaki tüm Web servislerini bulmak üzere arama emri gönderir. Örnek vermek gerekirse:

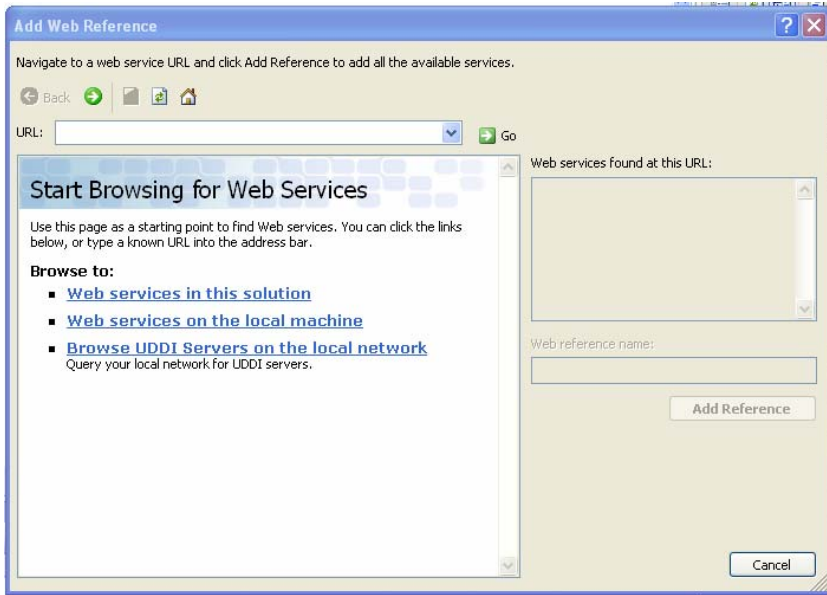
```
<?xml version="1.0" encoding="utf-8" ?>
<dynamicDiscovery
xmlns="urn:schemas-dynamicdiscovery:disco.2000-03-17">
<exclude path="_vti_cnf" />
<exclude path="_vti_pvt" />
<exclude path="_vti_log" />
<exclude path="_vti_script" />
<exclude path="_vti_txt" />
<exclude path="Web References" />
</dynamicDiscovery>
```

UDDI(Universal Description,Discovery and Integration), Microsoft tarafından oluşturulmuş, kurumların kendilerini, sağladıkları servisleri yayınlamak için tanımlamalarını ve bu bilgilerin daha sonra diğer kurumlarca taranıp bulunmasını sağlayan bir standarttır. UDDI Kurum Kayıt Servisi (UDDI Business Registry) kurum ve web servisleri bilgilerini saklayan sunuculardır. Bu sunucular servis sağlayıcılarından gelen bilgileri kendi veritabanlarına kayıt ederek diğer kurumların erişimine açar. Şu anda aktif olarak çalışan kurum kayıt sunucuları uddi.microsoft.com ve uddi.ibm.com 'dur.

3.5. “Add Web Reference”

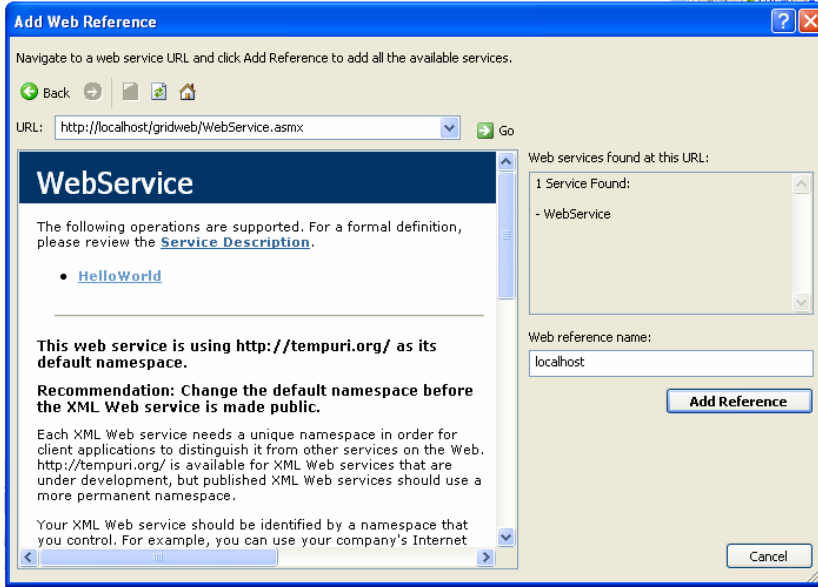
Add Web Reference ile projeye bir WEB Servis ekleyebiliriz. Yapmamız gereken Web servis URL'sini yazmak ve servisi getirmektir. Servis geldiğinde artık bu servisi kullanmak, basit bir class'ı new edip kullanmaktan farksızdır. Normal bir fonksiyonun işlenmesi için kod o fonksiyona dallanır, Web servisin metodunu işleyebilmek için ise Web servis'e dallanır ve cevap gelene kadar bekler. Burada dikkat etmeniz gereken şey metod çağrıldığında o metodun işlem süresinin, sizin ayarladığınız timeout süresinden kısa olmasıdır.

Projemize bir web servisini eklemek için WebSite menüsünden Add Web Reference komutu seçilir.



Şekil 3.4 : Add Web Reference

Add Web Reference penceresinde adres satırına Web servisimizin URL (Unified Resource Locator) adresini yazarak Şekil 3.5'deki Add Web Referance pencerine ulaşırız.

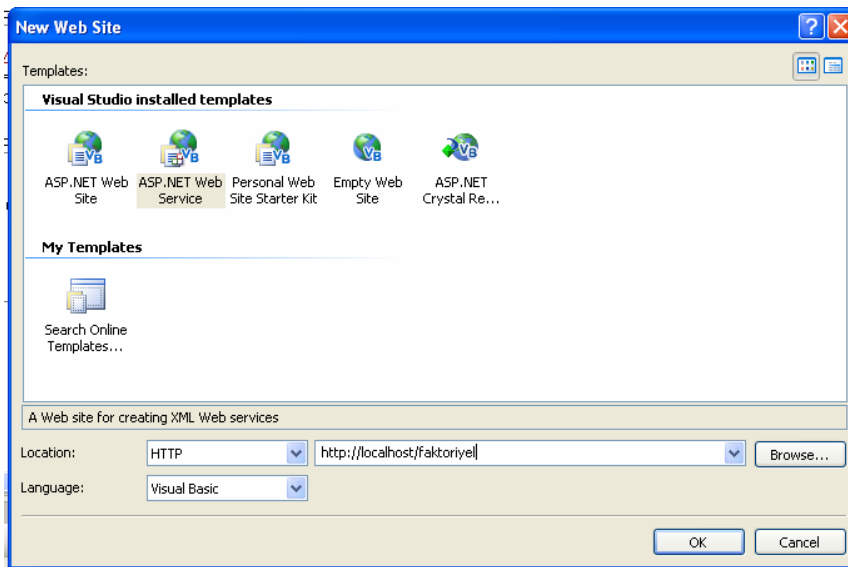


Şekil 3.5: Add Web Reference

Add Reference tıklanıldığında projemizin Solution Explorer penceresinde Web Reference kısmındaki servis ismiyle Web Servisimizi kullanabiliriz.

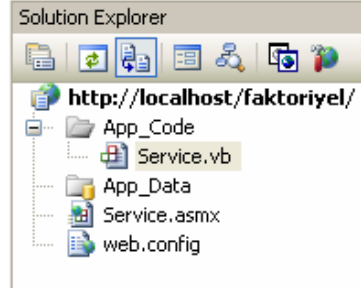
Web Servislerinin tam olarak anlaşılması için integer tipindeki bir sayının faktöriyelini bulan bir Web Servis ve bu servisi kullanan bir Web Form veya Windows uygulaması oluşturmaya çalışalım.

Öncelikle Visual Studio da File menüsünden New Web Site tıklanarak Şekil 3.6'dan ASP.NET Web Service seçilir.



Şekil 3.6: ASP.NET Web Service Oluşturma

Solution Explorer kısmına baktığımızda göreceğimiz Service.asmx dosyası Web Servis kodlarını saklayacak dosyadır.

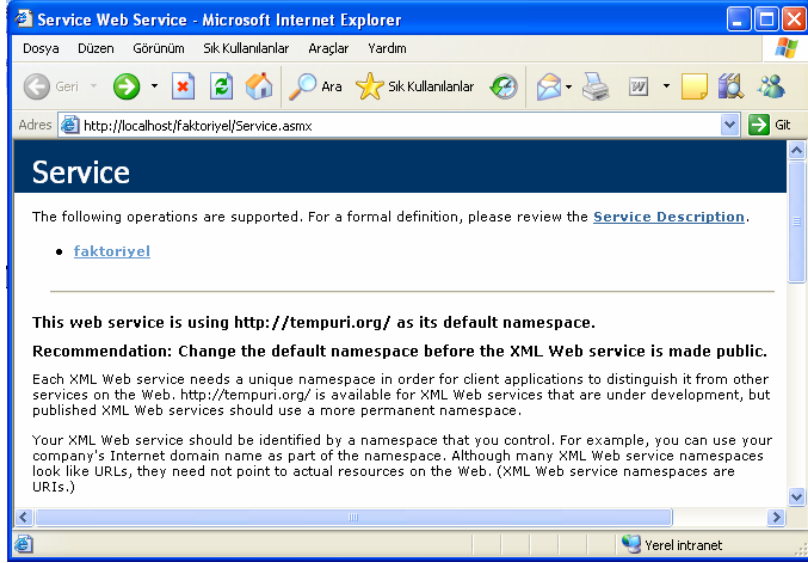


Şekil 3.7:Service.asmx

ASMX modülünü kod editöründe açtığımızda Web Servisi örnek açıklama (Hello World) şeklinde eklenmiştir. Bu örneği silerek <WebMethod()> işaretinden sonra fonksiyonumuzu yazabiliriz.

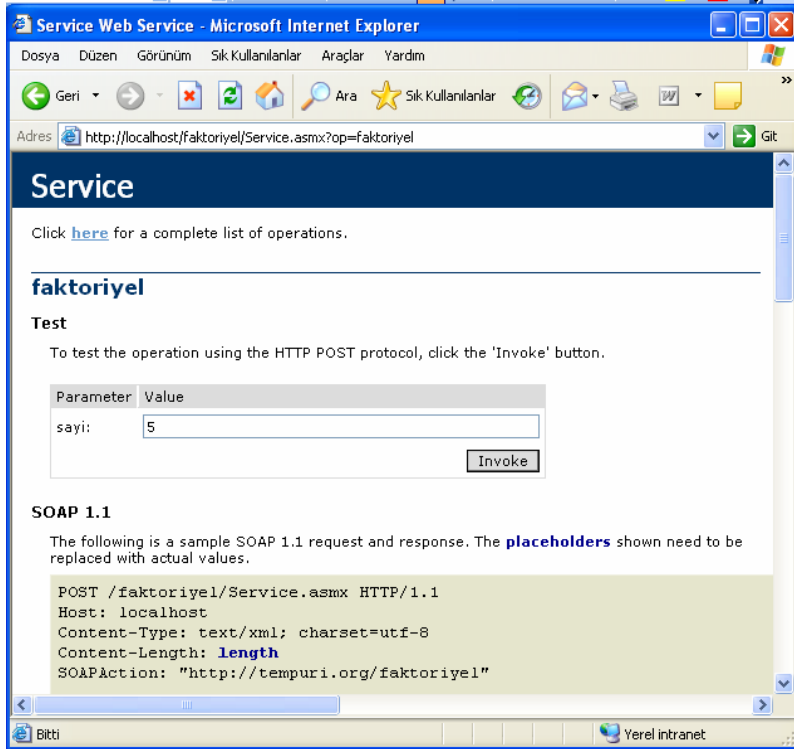
```
<WebService (Namespace:="http://tempuri.org/")> _
<WebServiceBinding (ConformsTo:=WsiProfiles.BasicProfile1_1)> _
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Public Class Service
    Inherits System.Web.Services.WebService
    <WebMethod()> _
    Public Function faktoriyel (ByVal sayi As Integer) As Long
        Dim i As Integer
        Dim fkt As Long
        fkt = 1
        For i = 1 To sayi
            fkt = fkt * i
        Next
        Return fkt
    End Function
End Class
```

Web Servisimizi test etmek için projeyi çalıştırdığımızda Şekil 3.8'deki tarayıcı penceresini göreceğiz.



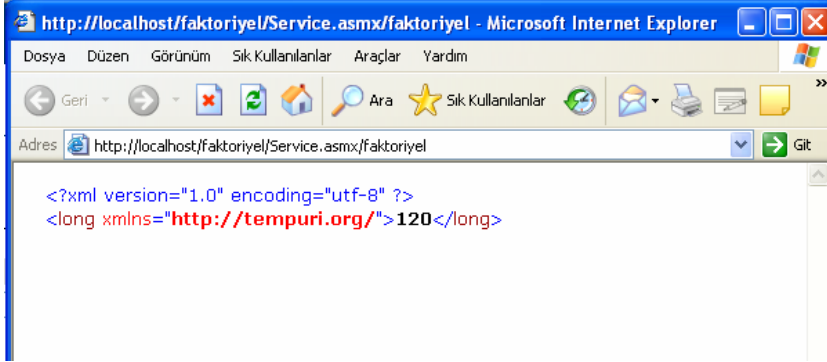
Şekil 3.8: Web servisi test etme

Tarayıcı ekranında **faktoriyel** bağlantısını tıklayarak Web Servisimizi test edeceğimiz test sayfası açılacaktır (Şekil 3.9).



Şekil 3.9: Faktoriyel fonksiyonunu çalıştır

Text kutusuna bir sayı girip Invoke butonuna tıkladığımızda Web Servisince geri döndürülen long tipindeki değeri içeren bir XML sayfası tarayıcıda görüntülenir.

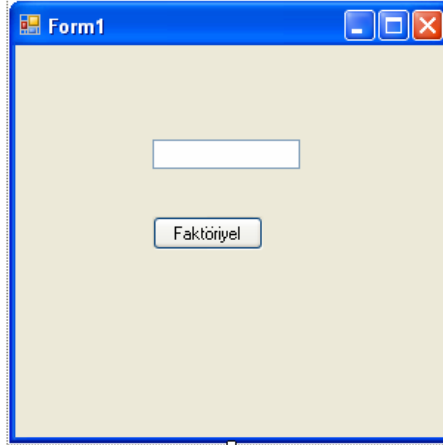


Şekil 3.10: Web servisin döndürdüğü XML sayfası

Artık Web Servisimiz kullanılmaya hazırdır.

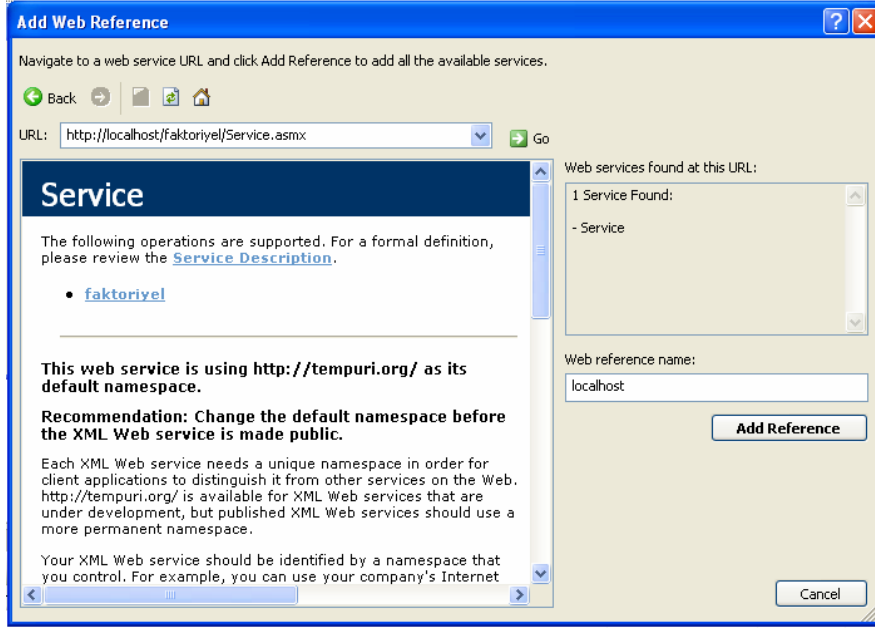
Oluşturduğumuz web servisini bir Windows uygulaması ile kullanabileceğimiz gibi bir Web sayfası ile de kullanabiliriz.

Web Servisinin nasıl çalıştığını görmek için bir Windows Application projesi açalım (File→New Project →Windows Application). Form üzerine faktöriyeli alınacak sayının girileceği bir TextBox, Buton ve sonucun görüntüleneceği bir Label ekleyelim.



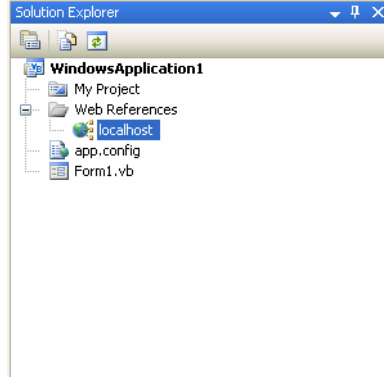
Şekil 3.11: TextBox, Button ve Label ekliyoruz.

Project →Add Web Reference (Web Sayfasından kullanılacaksa Web Site →Add Web Reference) komutu seçilerek Şekil 3.12'deki Add Web Reference diyalog kutusuna ulaşılır ve adres satırına (http://localhost/faktoriyel/Service.asmx) girilerek Add Reference butonuna tıklanır (Adres bilinmiyorsa **Web services on the local machine** tıklanarak servis listesi alınabilir).



Şekil 3.12: Add Web Reference

Solution Explorer penceresinde Web Reference kısmında dünya şeklindeki localhost ile Web Servis projemize eklenmiş olur.



Şekil 3.13: Uygulamaya Web servis eklendi

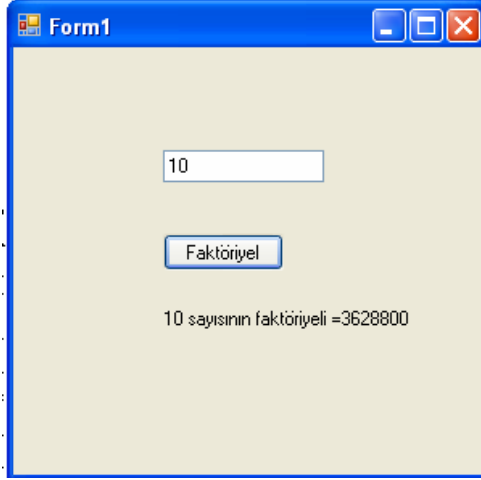
Şimdi Button1 in click olayını yazarak Web Servisimizi kullanalım;

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim txtfak As Integer
    Dim fak As Long
    Dim theservice As New localhost.Service
    txtfak = TextBox1.Text
    fak = theservice.faktoriyel(txtfak)
    MessageBox.Show(txtfak & "faktöriyeli =" & fak)
    Label1.Text = txtfak & " sayısının faktöriyeli =" & fak
End Sub
```

`Dim theservice As New localhost.Service`
kodu ile bir servis nesnesi örneği oluşturuyoruz.

`fak = theservice.faktoriyel(txtfak)`
kodu ile txtfak değişkenini Web servisin faktoriyel fonksiyonuna göndererek geri dönen sayısı fak değişkenine aktarıyoruz.

Projemizi çalıştırdığımızda Şekil 3.14'teki formu görürüz. Web Servisin çalışması gerekmediğine dikkat ediniz.

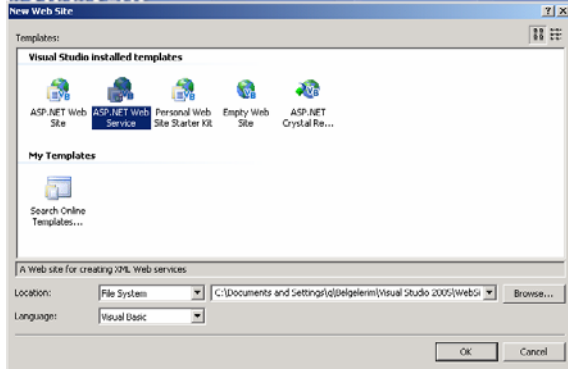


Şekil 3.14: Uygulamada çalışan web servis

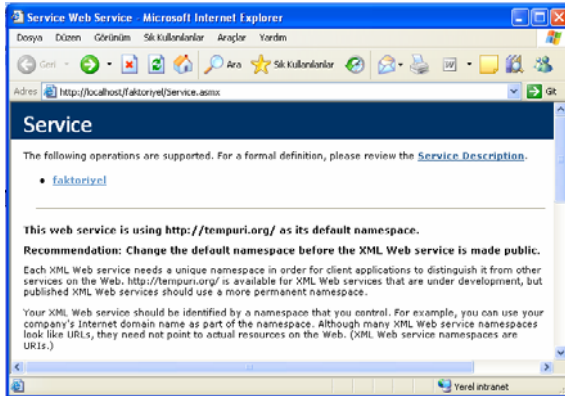
UYGULAMA FAALİYETİ

İşlem Basamakları

- Yeni proje açma penceresinden “Web Service Project” seçerek Web Servis oluşturunuz.



- Görsel arabirim olmadan, kod penceresinde gerekli programı yazınız.
- Projeyi çalıştırılınca açılan “Service help page” penceresinden istenen hizmeti seçerek test ediniz.



- Var olan bir servis sınıfını başka uygulamada çalıştırınız.

Öneriler

- T.C. Kimlik Numarası, Emekli Sandığı ve birkaç gazetenin Web sayfasını inceleyerek Web Servis hakkında bilgi ediniz.
- Oluşturduğumuz Web Servisi kullanabilmek için çalışması gerekmediğine dikkat ediniz.
- Web Servisi herhangi bir uygulama programıyla kullanabildiğimiz gibi bir Web sayfası ile de kullanabiliriz.

ÖLÇME VE DEĞERLENDİRME

A. OBJEKTİF TESTLER (ÖLÇME SORULARI)

Aşağıdaki çoktan seçmeli soruları cevaplayınız.

1. Aşağıdakilerden hangisi web servislerine olan ihtiyacı meydana getirmiştir?
A) İşletmelerin diğer işletmelerle olan iş süreçlerini bütünleştirme gereksinimi
B) NET teknolojisinin gelişmesi
C) Farklı web sayfalarının bir araya toplanma gereksinimi
D) Veri tabanı programlarının yeterli ihtiyacı karşılayamaması
2. HTTP üzerinde çalışan, fakat bilgiyi bir araya toplamak için XML benzeri özel bir format kullanan protokol aşağıdakilerden hangisidir?
A) UDDI
B) DISCO
C) SOAP
D) ASMX
3. Web servislerinde parametreler, dönüş değerleri ve internet üzerinde hangi iletim protokolünü kullanacağı gibi ayrıntılar ile beraber bir istemcinin Web servisi ile etkileşimini belirleyen XML tabanlı dosya aşağıdakilerden hangisidir?
A) ASMX
B) ASPX
C) DISCO
D) WSDL
4. Aşağıdakilerden hangisi ile Web sayfasına ve bir uygulamaya Web servisi eklenir?
A) Add Reference
B) Add Web Reference
C) Set As Start Page
D) Include in Project
5. Aşağıdaki ifadelerden hangisi doğrudur?
A) Web Servis ile Web sayfası aynı klasörde bulunmalıdır.
B) Var olan bir Web servisi sadece web sayfasında kullanılabilir.
C) Web servisi bir işletme içindeki birimlerin birbirleriyle iletişimini sağlamak amacıyla geliştirilmiştir.
D) Web Servisi kullanabilmek için çalışması gerekmemektedir.

DEĞERLENDİRME

Sorulara verdiğiniz cevapları modül sonundaki cevap anahtarıyla karşılaştırınız. Bu faaliyet kapsamında hangi bilgileri kazandığınızı belirleyiniz. Yanlış cevaplandığınız sorularla ilgili konuları tekrar inceleyip öğrenmeye çalışınız. Kaynak ve yardımcı ders kitaplarından faydalanınız. Kütüphanelerden, internetten veya görsel programlar yazan kişilerden araştırma yapınız. Öğrenme faaliyetiyle ilgili olarak öğretmenlerinizden yardım alınız.

MODÜL DEĞERLENDİRME

A.OBJEKTİF TESTLER (ÖLÇME SORULARI)

1. Web sayfasının başlığı aşağıdaki özelliklerden hangisinde belirtilir?
A) Title
B) Div
C) Form
D) BgColor
2. ToolTip özelliğinin görevi aşağıdakilerden hangisidir?
A) Sekme sırası
B) Kısayol tuşu
C) Açıklama bilgisi
D) Hızlı erişim
3. Aşağıdaki kontrollerden hangisi hata özetlerini verir?
A) Range Validator
B) Compare Validator
C) Validation Summary
D) Custom Validator
4. IIS kurulu bir bilgisayarda Web sunucu tarafı çalıştırılacak bir Web sayfası nerede saklanır?
A) wwwroot klasörü
B) Belgelerim\Project klasörü
C) Internet Information Service klasörü
D) İstenilen herhangi bir klasörde
5. Set As StartPage seçeneğinin görevi aşağıdakilerden hangisidir?
A) Visual Studio ilk açıldığında yüklenen sayfadır
B) Sayfa yüklenirken gerçekleşen olaydır
C) Visual Studio her açıldığında ekrana gelmesi istenen sayfadır
D) Birden fazla Web Form var ise çalıştırıldığında hangisi ile başlayacağını belirtir
6. Aşağıdakilerden hangisi bir DataSource tipi değildir?
A) XML dosyası
B) Access Veri Tabanı
C) Site Map
D) XLS dosyası
7. DataSource konfigürasyonunda Select Statment penceresinde ORDER BY.. Butonunun görevi nedir?
A) Verilerin seçilmesini sağlar.
B) Verilerin sıralanmasını sağlar.
C) Verilerin güncellenebilmesini sağlar.
D) Verilerin silinebilmesini sağlar.
8. Kontrole veri bağlamak için kullanılan fonksiyon aşağıdakilerden hangisidir?
A) DataBind
B) DataSource
C) DataAdapter
D) DataTables
9. Aşağıdaki kontrollerin hangisinde DataBindingsDialogBox kullanılabilir?
A)CheckBoxList
B)Label
C)DataList
D)DataGrid

10. Aşağıdakilerden hangisi web servislerine olan ihtiyacı meydana getirmiştir?
A) İşletmelerin diğer işletmelerle olan iş süreçlerini bütünleştirme gereksinimi
B) NET teknolojisinin gelişmesi
C) Farklı Web sayfalarının bir araya toplanma gereksinimi
D) Veri tabanı programlarının yeterli ihtiyacı karşılayamaması
11. HTTP üzerinde çalışan, fakat bilgiyi bir araya toplamak için XML benzeri özel bir format kullanan protokol aşağıdakilerden hangisidir?
A) UDDI
B) DISCO
C) SOAP
D) ASMX
12. Web servislerinde parametreler, dönüş değerleri ve internet üzerinde hangi iletim protokolünü kullanacağı gibi ayrıntılar ile beraber bir istemcinin Web servisi ile etkileşimini belirleyen XML tabanlı dosya aşağıdakilerden hangisidir?
A) ASMX
B) ASPX
C) DISCO
D) WSDL
13. Aşağıdakilerden hangisi ile Web sayfasına ve bir uygulamaya Web servisi eklenir?
A) Add Reference
B) Add Web Reference
C) Set As Start Page
D) Include in Project
14. Aşağıdaki ifadelerden hangisi doğrudur?
A) Web Servis ile Web sayfası aynı klasörde bulunmalıdır.
B) Var olan bir Web servisi sadece web sayfasında kullanılabilir.
C) Web servisi bir işletme içindeki birimlerin birbirleriyle iletişimini sağlamak amacıyla geliştirilmiştir.
D) Web Servisi kullanabilmek için çalışması gerekmemektedir.
15. Aşağıda verilen komut satırının görevi nedir?

```
Dim adapter As New OleDbDataAdapter(sql, conn)
Dim ds As New DataSet()
adapter.Fill(ds, "melektablo")
```

B. PERFORMANS TESTİ (YETERLİK ÖLÇME)

Modülün Adı	İnernet Uygulamaları	Öğrencinin				
Amaç	Gerekli ortam sağlandığında, internet uygulamaları ve Web servisleri yapabileceksiniz.	Adı...: Soyadı: Sınıfı : Nu....:
AÇIKLAMA: Aşağıda listelenen davranışların her birinde öğrencide gözleyemediyse (0), Zayıf nitelikli gözlediyseniz (1), Orta düzeyde gözlediyseniz (2), ve iyi nitelikte gözlediyseniz (3) rakamın altındaki ilgili kutucuğa X işareti koyunuz.						
GÖZLENECEK DAVRANIŞLAR			0 (kötü)	1 (zayıf)	2 (orta)	3 (iyi)
“WEB FORM APPLICATION”						
1. IIS (Internet Information Services), URL, HTML ve HTTP deyimleri (runat="server")						
2. Microsoft FrontPage Server Extensions (FPSE)						
3. Web form ve web sayfası kavramları						
4. “*.aspx” uzantısı						
5. “http://localhost” adresi ve “c:\inetpub\wwwroot” klasörü						
6. “Web Form Designer” penceresi ve HTML sekmesi						
7. “Web Server Controls” nesnelere (Label, Textbox, Button, LinkButton...)						
8. “Validation Controls”						
9. “AutoPostBack ve EnableViewState” özellikleri						
10. Basit HTML etiketleri (HTML, body, title, form, asp...)						
11. “Set as Start Page” seçeneği						
“ADO.NET”İ WEB UYGULAMASINDA KULLANMA						
1. “Data Grid, Repeater” nesnelere						
2. Kod penceresinde nesnelere veriye bağlama (DataSource ve DataMember özellikleri)						
3. “DataBind” komutu						
4. “DataBindings” diyalog kutusu						
WEB SERVİSLERİ						
1. “Distributed web applications” deyimini						
2. SOAP (Simple Object Access Protocol)						
3. WSDL (Web Services Description Language)						
4. “*.asmx ve disco” uzantısı						
5. “Add Web Reference” penceresi						
TOPLAM PUAN						

DEĞERLENDİRME

Yaptığınız değerlendirme sonucunda eksikleriniz varsa öğrenme faaliyetlerini tekrarlayınız.

Modülü tamamladınız, tebrik ederiz. Öğretmeniniz size çeşitli ölçme araçları uygulayacaktır. Öğretmeninizle iletişime geçiniz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1 CEVAP ANAHTARI

1	A
2	C
3	C
4	A
5	D

ÖĞRENME FAALİYETİ-2 CEVAP ANAHTARI

1	D
2	B
3	A
4	B
5	<ul style="list-style-type: none">➤ adapter adında OleDbDataAdapter nesnesi oluşturularak verilen stringlere göre bağlantıyı oluştur.➤ ds adında bir DataSet oluştur.➤ Melektablo tablosuyla, adapter OleDbDataAdapter aracılığı ile ds DataSetini doldur.

ÖĞRENME FAALİYETİ-3 CEVAP ANAHTARI

1	A
2	C
3	D
4	B
5	C

MODÜL DEĞERLENDİRME CEVAP ANAHTARI

1	A
2	C
3	C
4	A
5	D
6	D
7	B
8	A
9	B
10	A
11	C
12	D
13	B
14	C
15	<ul style="list-style-type: none">➤ adapter adında OleDbDataAdapter nesnesi oluşturarak verilen stringlere göre bağlantıyı oluştur.➤ ds adında bir DataSet oluştur.➤ Melektablo tablosuyla, adapter OleDbDataAdapter aracılığı ile ds DataSetini doldur.

ÖNERİLEN KAYNAKLAR

- DAVIS Harold, **Elektrik Visual Basic.NET Programlama Kılavuzu Sybex**, Alfa, İstanbul, 2002.
- PALA Zeydin, **Visual Basic. Net**, Türkmen Kitabevi, İstanbul, 2003.

KAYNAKÇA

- www.godoro.com
- <http://msdn2.microsoft.com/en-us/default.aspx>
- www.evcil.net
- www.mutasyon.net
- www.bilyaz.com
- www.belgeler.org