

T.C.
MİLLÎ EĞİTİM BAKANLIĞI



MEGEP

(MESLEKİ EĞİTİM VE ÖĞRETİM SİSTEMİNİN GÜÇLENDİRİLMESİ
PROJESİ)

BİLİŞİM TEKNOLOJİLERİ

ETKİLEŞİMLİ WEB UYGULAMALARI 1

ANKARA 2008

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	ii
GİRİŞ	1
ÖĞRENME FAALİYETİ - 1	3
1. YAZILIM KURULUMU	3
1.1. Web Sunucusu Kurulumu	5
1.2. .NET Yazılımı Kurulumu	7
1.3. .NET Yazılımının Özellikleri ve Avantajları	11
1.4. İnternet Uygulamalarında İşleyiş Modelleri	16
1.4.1. İstemci Tarafı (Client-Side) Kodlama	16
1.4.2. Sunucu Tarafı (Server-Side) Kodlama	17
1.4.3. Bağlantısızlık ve Olay Yönlendirmeli (Eylem Yönelimli, Event Driven) Programlama	18
1.5. ASP.NET Sayfalarının Çalıştırılması	18
1.5.1. Web Sunucusu Yönetimi	24
1.5.2. Sanal Dizin (Virtual Directory)	27
UYGULAMA FAALİYETİ	32
ÖLÇME VE DEĞERLENDİRME	33
ÖĞRENME FAALİYETİ - 2	34
2. ASP.NET SAYFA YAPISI	34
2.1. Türkçe Karakterler	34
2.2. Sayfa Yapısı	36
2.3. ASP.NET Sayfa Elemanları	40
2.3.1. Sayfa Bildirimleri Bölümü	41
2.3.2. Kod bildirim bloğu	41
2.3.3. HTML Sayfa Bildirimleri ve HTML Etiketleri	42
2.3.4. Kod Çalıştırma Bloğu	42
2.3.5. Web Formu ve Web Kontrolleri	43
2.4. Kodu Alt Satırdan Devam Ettirme	44
2.5. Program Kodlarına Açıklamalar Ekleme	49
UYGULAMA FAALİYETİ	51
ÖLÇME VE DEĞERLENDİRME	52
MODÜL DEĞERLENDİRME	53
CEVAP ANAHTARI	55
ÖNERİLEN KAYNAKLAR	57
KAYNAKÇA	58

AÇIKLAMALAR

KOD	482dk0090
ALAN	Bilişim Teknolojileri
DAL/MESLEK	Web Programcılığı
MODÜLÜN ADI	Etkileşimli Web Uygulamaları 1
MODÜLÜN TANIMI	Etkileşimli web uygulamaları geliştirmek için gerekli olan yazılımların kurulumunu anlatan öğrenim materyalidir.
SÜRE	40/32
ÖN KOŞUL	Web Tasarımı Editörü 4 modülünü bitirmiş olmak
YETERLİK	Programlama için gerekli yazılım kurulumlarını yapmak
MODÜLÜN AMACI	Genel Amaç: Gerekli ortam sağlandığında; etkileşimli web uygulamaları için gerekli olan yazılımların kurulumunu yapabileceksiniz. Amaçlar: ➤ Web programlama için gerekli öğeleri tanıyarak, kurulumunu yapabileceksiniz. ➤ Programlama diline ilişkin ait kod yapısını tanıyarak, basit bir uygulama gerçekleştirebileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Web programlama yazılımlarını çalıştırabilecek yeterlikte bilgisayar ve gerekli lisanslı yazılımlar.
ÖLÇME VE DEĞERLENDİRME	Her faaliyet sonrasında o faaliyetle ilgili değerlendirme soruları ile kendi kendinizi değerlendireceksiniz. Modül içinde ve sonunda verilen öğretici sorularla edindiğiniz bilgileri pekiştirecek, uygulama örneklerini ve testleri gerekli süre içinde tamamlayarak etkili öğrenmeyi gerçekleştireceksiniz.

GİRİŞ

Sevgili Öğrenci

İnternetin müthiş bir hızla yayıldığı günümüzde sadece firmalar değil bireyler de bir internet sitesine sahip olmak istemektedirler. Öncelikle bir internet sitemiz olsun diye oluşturulan internet siteleri zamanla geliştirilerek dinamik, etkileşimli sitelere dönüştürülmektedir.

Bu modülde etkileşimli web uygulamaları geliştirmek için gerekli programları bilgisayarınıza kurmayı ve bir web uygulamasının yapısını oluşturan öğeleri öğreneceksiniz.

Modülde karşılaşacağınız terimlerin Türkçe anlamları için modüllerin sonundaki sözlük kısmından yararlanabilirsiniz. Daha ayrıntılı bilgiler için piyasada bulunan Bilişim sözlüğünden faydalanabilirsiniz.

Unutmayın ki ne kadar araştırmacı bir tavır sergiliyorsanız o kadar çok kendinizi geliştirirsiniz.



ÖĞRENME FAALİYETİ-1

AMAÇ

Web programlama için gerekli programları tanıyarak, kurulumunu yapabilmek.

ARAŞTIRMA

ASP.NET uygulamaları oluşturmak için bilgisayara kurulması gerekli programların neler olduğunu araştırınız. Araştırmalarınızı internette tarama yoluyla, piyasadaki kitapları inceleyerek, bilgisayar alanındaki uzmanlara sorarak yapabilirsiniz. Araştırma sonuçlarınızı öğretmene teslim edecek veya sınıfta sunacak şekilde raporlaştırınız.

Etkileşimli web uygulamaları geliştirmek amacıyla ASP.NET, CGI, ColdFusion (Koldfujın) , JSP, PHP gibi çeşitli teknolojiler kullanılabilir. Web uygulamasını geliştiren programcı bunlardan herhangi birisini seçebilir. Bu teknolojiler sayesinde alışveriş siteleri, internet bankacılığı gibi gelişmiş uygulamalar yapılabilir.

Bu teknolojilerden biri olan ASP.NET, Microsoft firması tarafından üretilmiştir. ASP kısaltması, Active Server Pages kelimelerinin baş harflerinden oluşmaktadır. Active Server Pages kelimelerinin Türkçesi “Etkin Sunucu Sayfaları”dır. ASP.NET, klasik ASP’ye kıyasla daha hızlı, işlevsel (fonksiyonel) ve güvenilirdir. ASP.NET, web uygulamaları geliştirme kavramına getirdiği yeni yaklaşım, yöntemler ve programlama tekniğiyle ASP’nin yeni bir sürümü olarak değil de farklı bir teknoloji olarak ele alınabilir.

İlk olarak ASP+ adıyla geliştirilen bu yeni teknoloji, Microsoft’un sunucu taraflı internet teknolojilerini de .NET platformu altında toplama kararından sonra ASP.NET adını almıştır.

Modül boyunca ASP.NET sayfası ve ASP.NET uygulaması kavramlarıyla karşılaşacaksınız. ASP.NET sayfası dosya uzantısı aspx olan dosya iken, ASP.NET uygulaması ise sadece ASP.NET sayfasını da barındırabileceği gibi, farklı dosyaları da (örneğin, veritabanı dosyaları) barındırabilen uygulamadır. Web uygulamaları, genellikle web sunucu üzerindeki veritabanında verileri depolayıp ziyaretçilerin bu bilgiye erişmesini ve değiştirmesini sağlayan dinamik web siteleridir.

Hem ASP hem de ASP.NET, sunucuda çalışan uygulamalardır. Hem ASP hem de ASP.NET sayfalarının sunucuda işlenmesi sonucu oluşan çıktı HTML kodu şeklinde web tarayıcısında (web browser) görüntülenir. ASP.NET kodları istemci tarafından görünmez.

Sunucu taraflı programlamayla kişiye özel çözümler üretilebilirsiniz. Örneğin kullanıcının özelliklerine, kullanıcının web tarayıcısı tipine (Explorer, Firefox, Netscape,

Opera gibi), kullanıcının sitenize nereden bağlandığına ya da bir alışveriş sitesinde müşterinin daha önce aldığı ürünlere göre farklı sayfalar gelmesini sağlayabilirsiniz.

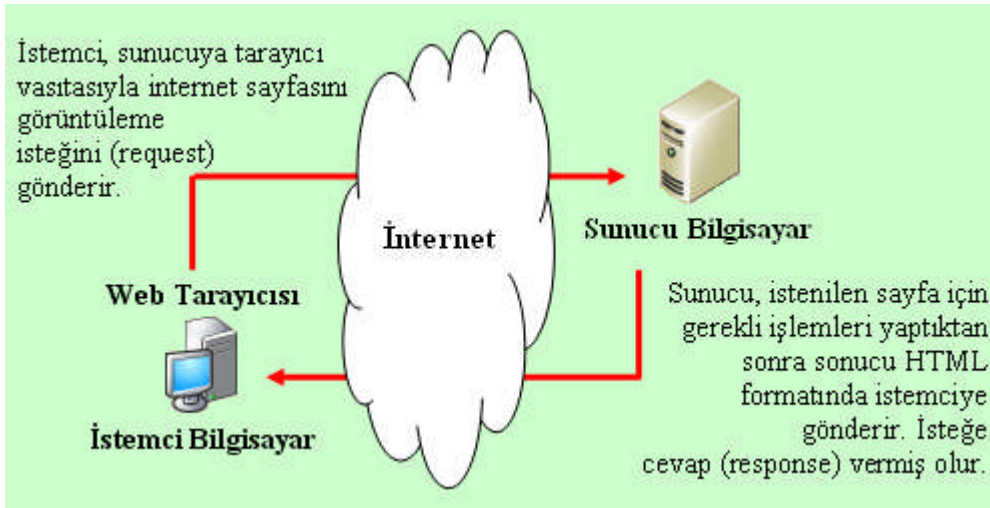
Web tarayıcısı (web browser)

Örün tarayıcısı, ürün gezgini. İnternet üzerinde bilgi kaynaklarını aramaya elveren ve bağlantılı metin ve ortamların olanaklarını kullanan istemci yazılımı (Bilişim Sözlüğü). Web kelimesi Türkçeye ürün, örütbağ olarak çevrilmiştir. Kimi kaynaklarda bu kelimelerle karşılaşabilirsiniz. Browser kelimesi de göz atıcı, gezgin, tarayıcı, görüntüleyici anlamlarına gelmektedir.

İnterneti gezerken karşımıza çıkan sayfaların bir kısmı istemci bilgisayarda çalışan, bir kısmı ise sunucuda çalışan sayfalardır. Eğer sayfayı oluşturan kodlar/komutlar/deyimler istemci bilgisayarda çalıştırılıyorsa o sayfa istemci tarafı sayfa, kodlar sunucu bilgisayarda çalıştırılıyorsa o sayfa sunucu tarafı sayfa olarak isimlendirilir.

İstemci tarafı sayfalar, sunucu tarafından üzerinde bir işlem yapılmadan istemci bilgisayara gönderilir. İstemci bilgisayardaki tarayıcı program bu sayfaları yorumlayarak görüntüler. Özetle, tüm işlemleri istemci bilgisayar yapar.

ASP.NET sayfaları gibi sunucu tarafı sayfalarda tüm kodlar sunucu bilgisayarında çalışır, işlem sonuçları HTML şeklinde istemci bilgisayara gönderilir. İstemci bilgisayardaki web tarayıcı program HTML satırlarını internet sayfaları şeklinde kullanıcıya gösterir. Örneğin, bir öğrenci ÖSYM sitesine girip, ÖSYM numarasını yazdığı anda, sunucuda kendisine ait işlemler yapılır, sonuçlar HTML şeklinde öğrencinin kullandığı bilgisayara gönderilir.



Resim 1.1: İstemci-sunucu mantığı

ASP'de dosya uzantısı **.asp**, ASP.NET'de dosya uzantısı **.aspx** olarak verilir. Aynı sunucuda ASP ve ASP.NET birlikte çalışabilir. ASP.NET'in en büyük özelliklerinden biri de ASP.NET uygulamalarını 20'den fazla programlama dili ile gerçekleştirebilmenizdir.

Örneğin, C# (C Sharp), C++ (C Plus Plus), Java, Pascal (Paskal), Perl (Pörl), Visual Basic (Görsel Basic).

Etkileşimli Web Uygulamaları modüllerinde ASP.NET'in 2.0 sürümü kullanılacaktır. ASP.NET 2.0 aşağıdaki platformlarda çalışabilir:

Microsoft Windows 2000 Professional and Server (Service Pack 3)
Microsoft Windows XP Professional.
Microsoft Windows 2003 Server Ailesi.

Yukarıdaki işletim sistemlerini rahatlıkla kullanabildiğiniz bir bilgisayar donanımı ASP.NET'i de çalıştıracaktır.

ASP.NET sayfalarını oluşturabilmek için metin editörü programına, çalıştırabilmek veya test etmek içinse .NET Framework (Çerçeve) programı ile bir web sunucu programını kurmaya ihtiyacınız vardır. Ayrıca ASP.NET sayfalarınızı görüntüleyebilmek için bir web tarayıcısı kullanacaksınız. Bu modül yazılırken aşağıdaki özelliklere sahip bir bilgisayar kullanılmıştır:

İşletim Sistemi	Windows XP Professional Service Pack 2
Web sunucu yazılımı	Internet Information Services 5.1 – IIS (internet bilgi servisi)
.NET Framework	.NET Framework Version 2.0 Redistributable Package
Web tarayıcısı	IE 6.0 (Internet Explorer 6.0)

Tablo 1.1: Modül yazılırken kullanılan bilgisayarın özellikleri

NOT: .NET Framework programının 2.0 versiyonuyla birlikte IIS'nin 5.x'den önceki sürümleri kullanılamaz.

Bir ASP.NET uygulaması geliştirirken hataları düzeltmek için 2 yol vardır. İlki, bilgisayarınızı ASP.NET uygulamaları çalıştıracak bir sunucuya çevirmek, diğeri ise internet ortamında ASP.NET sayfalarını bulunduran, sunan (hosting) bir firmanın sunucu bilgisayarına yükleyip buradan test etmektir.

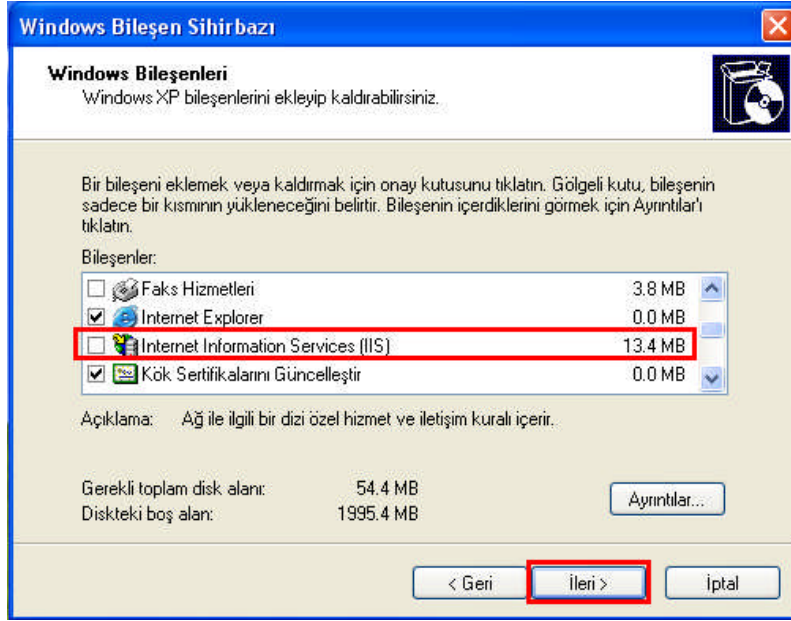
Bu iki yöntemden birincisi, zaman ve maliyet açısından daha yararlıdır. ASP.NET sayfalarını geliştirebilecek şekilde bilgisayarınızı ASP.NET sunucusuna çevirdiğinizde, internete bağlı olmadan sayfalarınıza son şeklini verebilirsiniz.

1.1. Web Sunucusu Kurulumu

Sunucuda çalışan ASP.NET sayfalarını kendi bilgisayarınızda geliştirmek, test etmek için bilgisayarınızı sunucu bilgisayar gibi kullanabilmeniz gereklidir. Bu amaçla bilgisayarınıza çeşitli web sunucu programlarını yüklemelisiniz. Etkileşimli Web Uygulamaları modüllerinde web sunucusu olarak Internet Information Services (IIS) programı kullanılacaktır. Bilgisayarınıza IIS gibi bir web sunucusu yüklediğinizde bilgisayarınızı sunucu bilgisayar gibi kullanabilirsiniz. ASP.NET sayfalarının

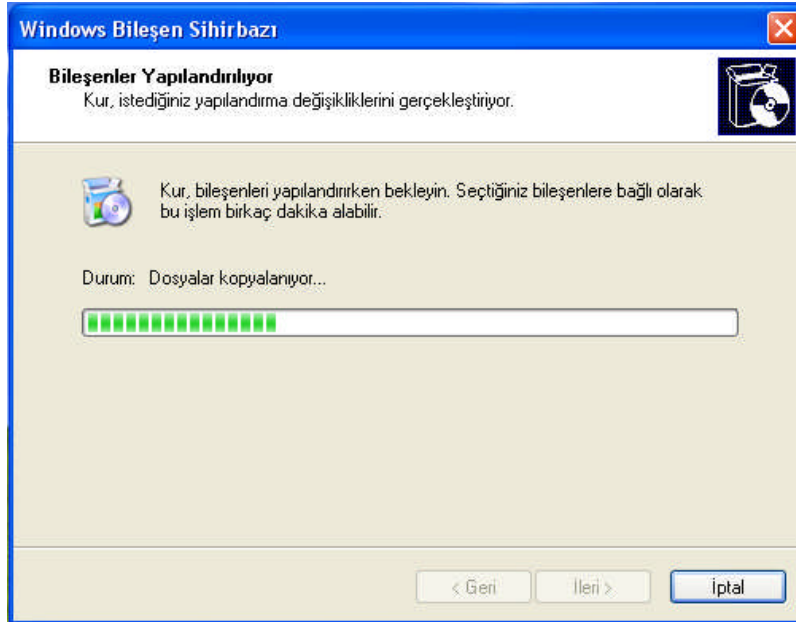
çalıştırılmasından istemci bilgisayara gönderilmesine kadar ki tüm işlemleri web sunucusu yerine getirir.

Şimdi IIS kurulumu gerçekleştirelim. **Denetim Masası**'ndan **Program Ekle/Kaldır** iletişim kutusuna gelerek, **Windows Bileşenlerini Ekle/Kaldır** düğmesine tıklarız. Karşınıza aşağıdaki pencere çıkacaktır.



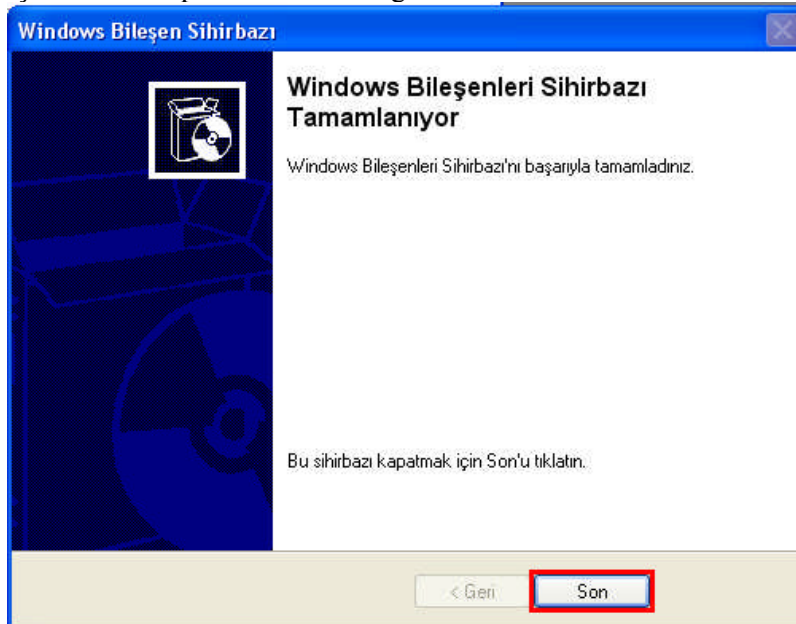
Resim 1.2: Windows Bileşenleri penceresi

Bu pencereden **Internet Information Services (IIS)**'yi seçip, Windows CD'sini takarak (İsterseniz Ayrıntılar düğmesiyle IIS için isteğe bağlı ek bileşenleri ekleyebilir veya kaldırabilirsiniz) **İleri** tuşuna basınız. IIS kurulumunu gösteren aşağıdaki pencere gelecektir.



Resim 1.3: Bileşenler yapılandırılıyor penceresi

Karşınıza çıkan sonraki pencerede **Son** düğmesine tıklatınız.



Resim 1.4: Windows bileşenleri sihirbazı tamamlanıyor penceresi

1.2. .NET Yazılımı Kurulumu

.NET Framework içinde bulundurduğu servis, program ve hizmetlerle ASP.NET sayfalarınızı test edebilmenizi ve yayımlamanızı sağlayan Microsoft'un geliştirdiği uygulama geliştirme platformudur.

İnternet üzerinde ASP.NET sayfaları barındırmak, işletim sistemini .NET uyumlu yapmak için .NET Framework yazılımının yüklenmesi gereklidir. ASP.NET sayfaları oluşturmak için gerekli olan dosyalar .Net Framework içerisinde yer almaktadır. Sisteminize Framework 1.1 kurduğunuzda ASP.NET 1.1, Framework 2.0 kurduğunuzda ASP.NET 2.0 sürümüyle çalışırsınız. .NET Framework kendini IIS'ye göre yapılandırdığından (konfigüre ettiğinden) .NET Framework kurulmadan önce IIS programı sisteminizde kurulu olmalıdır.

Framework'un "Redistributable" (yeniden dağıtılabilir) ve "SDK" isimleriyle 2 farklı dağıtımı vardır. Bir tanesini yüklemek yeterlidir.

"Redistributable" şeklindeki dağıtım, ASP.NET uygulamalarınızı çalıştırmak için gerekli bütün bileşenlere sahiptir.

"Software Development Kit – SDK (yazılım geliştirme kiti) dağıtımı ise, Redistributable dağıtımına ek olarak, yardım dosyaları, çeşitli araçlar, örnek uygulamalar içermektedir. Web uygulamaları geliştirmede tecrübe kazandığınızda büyük ihtimalle SDK dağıtımını kullanmayı tercih edeceksiniz. İçerdiği ek program ve dokümanlardan dolayı SDK dağıtımının dosya boyutu diğerine göre fazladır.

.NET Framework'ü aşağıdaki siteden indirebilirsiniz.(download)
<http://msdn.microsoft.com/netframework/downloads/updates/default.aspx>

Bu siteye girdiğinizde ".NET Framework Version 2.0 Redistributable Package" başlığı altındaki "Download x86 version" bağlantısını tıklatın. Karşınıza gelen pencerede "Download" düğmesine tıklatın. Tıklattığınızda karşınıza gelen iletişim kutusunda "Kaydet" düğmesine tıklatarak dosyayı (dotnetfx) bilgisayarınızda uygun bir yere kaydedin.

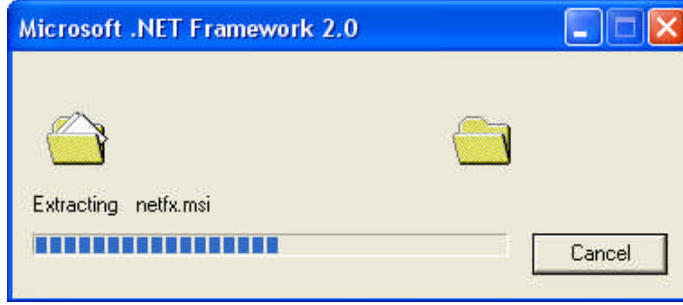
.NET Framework Version 2.0 Redistributable paketinin ihtiyaç duyduğu sistem gereksinimi yukarıda bahsedilen Microsoft' un sitesinde aşağıdaki şekilde belirtilmiştir.

Sistem Gereksinimleri
Desteklenen İşletim Sistemleri : Windows 2000 Service Pack 3; Windows 98; Windows 98 Second Edition (SE); Windows ME; Windows Server 2003; Windows XP Service Pack 2
Gerekli yazılım : Windows Installer 3.0. (Windows 98/ME için Windows Installer 2.0.) Windows Installer 3.1 veya daha üst sürümler tavsiye edilir. Internet Explorer 5.01 veya daha üst sürümleri
Gereken disk alanı: 280 MB (x86), 610 MB (x64)

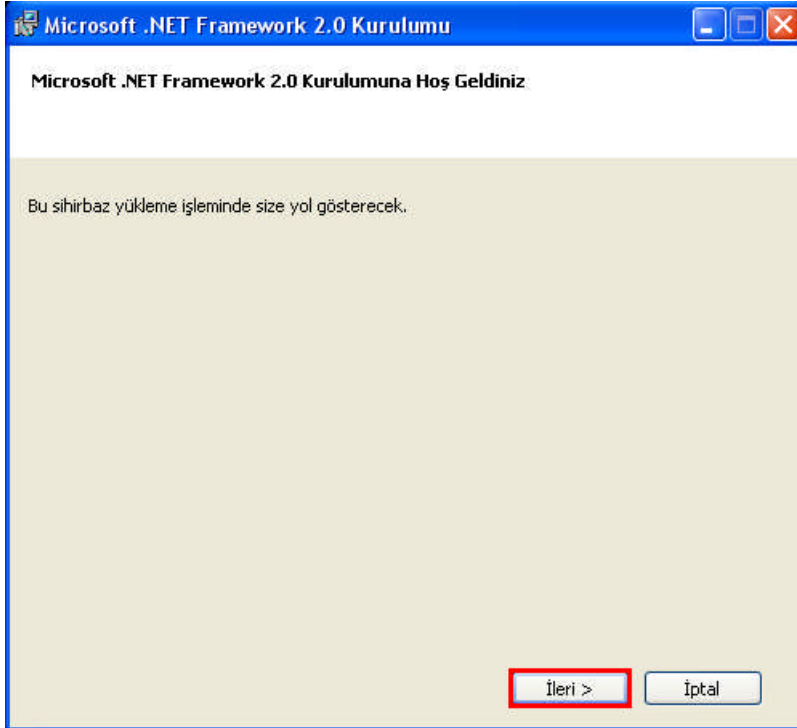
Tablo 1.2: Sistem Gereksinimleri

Etkileşimli Web Uygulamaları modüllerinde Redistributable 2.0 dağıtımını kullanmıştır. İsterseniz Redistributable 2.0 dağıtımını kurduktan sonra SDK dağıtımını kurabilirsiniz.

.NET Framework'ü kurmaya başlayalım. Yukarıdaki adresten indirmiş olduğunuz Redistributable 2.0 dağıtımını içeren **dotnetfx** dosyasını çalıştırınız. Program kurulum için gereken dosyaları açacaktır.

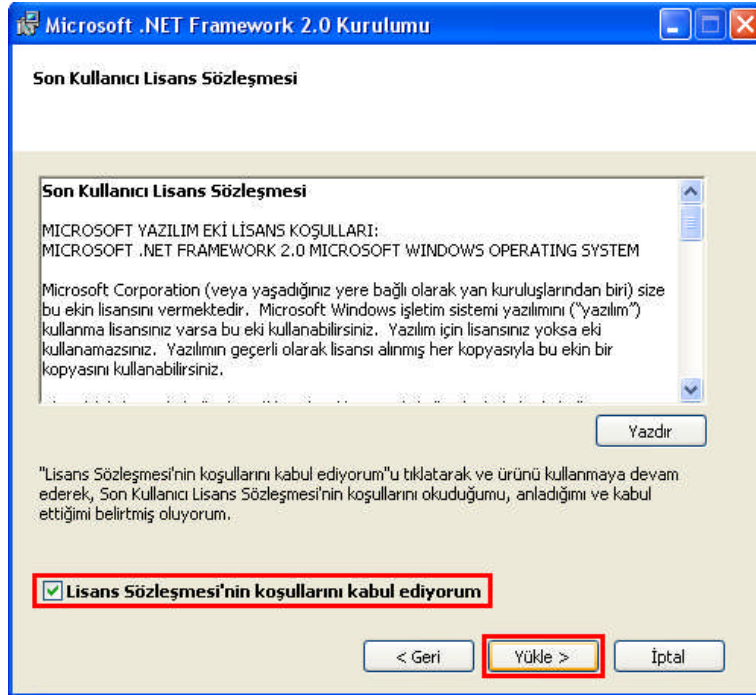


Resim 1.5: .NET Framework yükleniyor



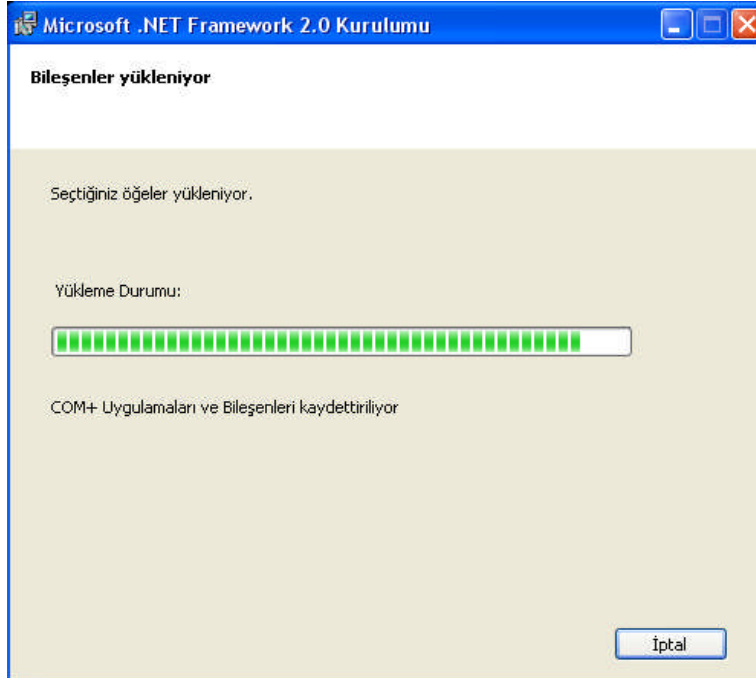
Resim 1.6 : Microsoft .NET Framework 2.0 Kurulumuna Hoş Geldiniz penceresi

Ardından kurulum için kullanacağınız sihirbaz karşınıza gelecektir. Bu pencerede **İleri** düğmesine tıklayınız. Karşınıza **Son Kullanıcı Lisans Sözleşmesini** gösteren pencere gelecektir.



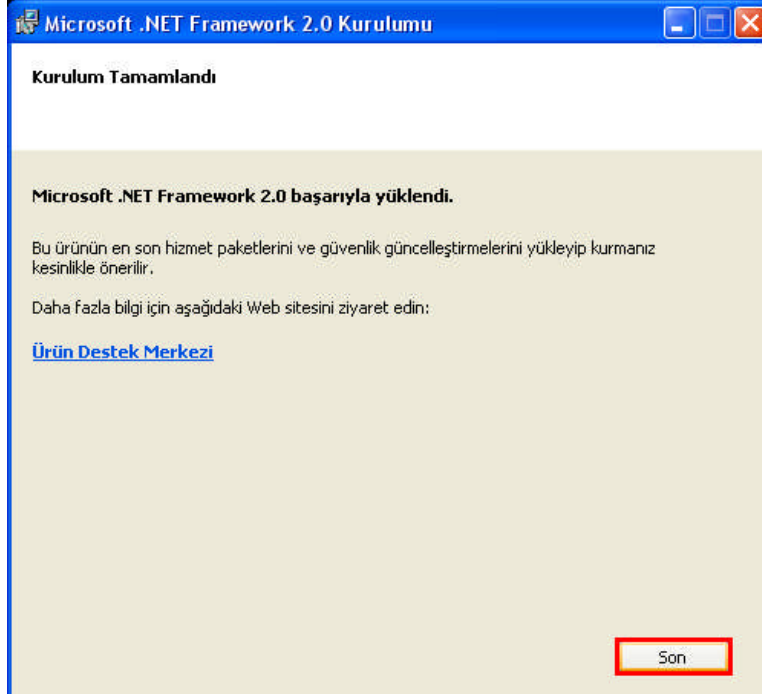
Resim 1.7: Son kullanıcı lisans sözleşmesi penceresi

Lisans sözleşmesinin koşullarını kabul ediyorum onay kutusunu işaretledikten sonra **Yükle** düğmesine basınız.



Resim 1.8: Bileşenler yükleniyor penceresi

.NET Framework'ün bileşenleri yüklenmeye başlayacaktır.



Resim 1.9: Kurulum tamamlandı penceresi

Kurulum Tamamlandı pencesinde **Son** düğmesine basarak kurulumu bitiriniz.

NOT: Bu ürünün en son hizmet paketleri ve güvenlik güncelleştirmeleri çıktıkça kurarsanız sisteminizi güncel tutmuş olursunuz.

1.3. .NET Yazılımının Özellikleri ve Avantajları

Eskiden beri programlar, çalıştırılacakları bilgisayarın işletim sistemine göre derlenerek makine dilini çevrilmekte ve bu şekilde işletim sistemiyle doğrudan bağlantı kurarak çalışmaktadır. Programın hangi işletim sisteminde çalışacak şekilde yazıldığına kullanıcı tarafından bilinmesi gerekir. Aşağıda Türkçe bir sıkıştırma programı olan Ezip programının hangi işletim sisteminde çalışacak şekilde hazırlandığı belirtilmiştir. Bu program Windows işletim sistemi dışındaki diğer işletim sistemlerinde çalışmaz.

Ezip

Sürüm	: 1.0
Boyut	: 1.39 Mb
Kullanım İzni	: Ücretsiz (Freeware)
Dil	: Türkçe
İşletim Sistemi	: Windows 95/98/NT/2000

Resim 1.10 : Ezip programı

Program bir işletim sistemine göre yazıldıktan sonra farklı işletim sistemlerinde çalışmaz. Sonuç olarak, bir programın farklı işletim sistemlerinde de çalışması için istenilen işletim sistemi için ayrı olarak yazılması gerekir. Aşağıda “Opera Changelogs” isimli internet tarayıcısının hangi işletim sistemleri (platformlar) için hazırlanmış olduğu görülmektedir.

Opera Changelogs: All platforms (Opera Changelogs : Tüm Platformlar) Choose your platform (Platformu Seçiniz)
Windows
Mac
Linux
FreeBSD
Solaris
OS2
QNX
Opera for Mobile (Mobil bilgisayarlar için)
Opera Mini™
Opera PocketPC
Opera for Windows Mobile
Opera for Symbian
Opera for Series60

Tablo 1.3: Farklı platformlar için Opera yazılımı

Firefox isimli web tarayıcısının farklı işletim sistemleri için hazırlanmış sürümlerinin linkleri aşağıdaki resimde görülmektedir. Dikkat ederseniz, farklı işletim sistemleri için yazılmış programın boyutlarında da farklılık vardır.



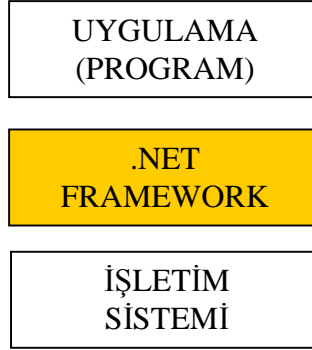
Resim 1.11 : Firefox programının farklı işletim sistemleri için hazırlanmış sürümleri

Programların farklı platformlarda çalışmasını sağlamak amacıyla Java programlama dilinin üreticisi Sun Microsystems firması, Java Sanal Makinesi programını geliştirmiştir. Böylelikle Java diliyle geliştirilmiş bir program, Java Sanal Makinesinin kurulu olduğu birçok platformda çalışabilmektedir. Fakat farklı platformda çalıştırılacak programın, Java programlama diliyle yazılma zorunluluğu, diğer programlama dillerinin üreticileri ve kullanıcıları için istenilmeyen bir durumdur.

Microsoft firması, herhangi bir programlama diliyle yazılmış bir programın her platformda çalışmasını sağlamak amacıyla .NET Framework programını geliştirmiştir. Şu an

.NET Framework, 20'den fazla dilin kullanılmasına destek vererek farklı dillerle program yazma alternatifini sunmaktadır. Fakat .NET ortamı sadece Windows üzerine kurulabilmektedir. Linux işletim sistemi üzerine kurulabilmesi için geliştirme çalışmaları devam etmektedir. Linux üzerinde ASP.NET uygulamaları oluşturmak amacıyla MONO adlı program geliştirilmektedir. Program geliştirme işlemleri tamamlandıktan sonra, herhangi bir dille yazılmış bir program Windows veya Linux platformlarında çalışabilecektir. Gelecekte varılmak istenen hedef, herhangi bir dille yazılmış bir programın herhangi bir işletim sistemi üzerinde sorunsuz çalışmasını sağlamaktır. Böylelikle yazılım geliştiriciler, yazdıkları programın işletim sistemiyle uyum sorunu yaşamayıp yaşamayacağı düşünmeden, bununla vakit kaybetmeden herhangi bir programlama diliyle programlarını yazabileceklerdir.

Sonuç olarak .NET Framework, yazılan uygulamaların derlenip işletim sistemine uygun olarak çalıştırılmasını sağlayan bir arabirimdir. Aşağıdaki şekilde de görüldüğü gibi .NET framework işletim sistemi ile uygulama arasında bir arabirim durumundadır.

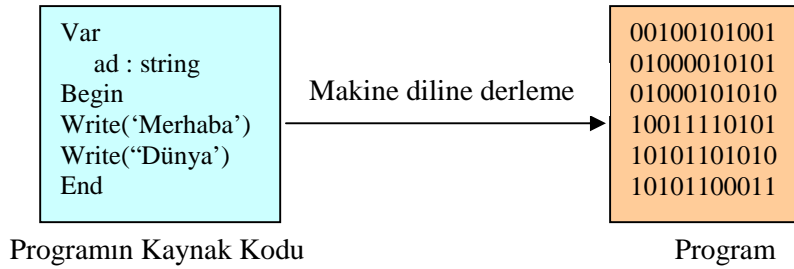


Şekil 1.1: .NET Framework'ün arabirim olma özelliği

NET Framework'ün yüklü olduğu bir bilgisayara C#, C++, JAVA, Perl, Visual Basic gibi farklı dilde yazılmış uygulamaları yükleyebilir ve sorunsuzca kullanabilirsiniz. Şimdilik sadece Microsoft tabanlı işletim sistemlerinde çalışan .NET Framework'ün diğer işletim sistemleri için sürümleri hazırlanmaktadır.

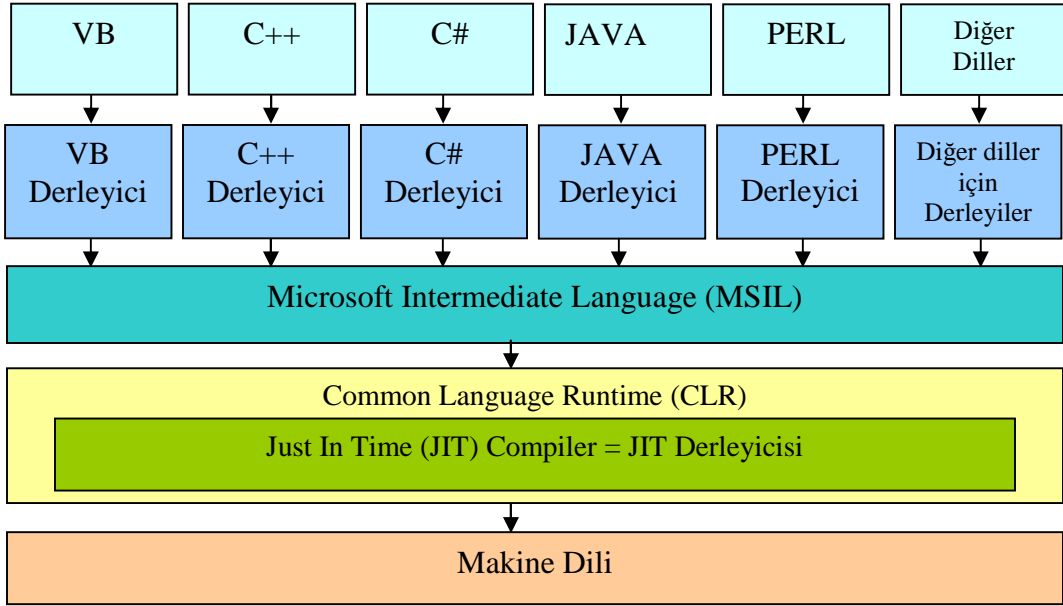
Net Framework çalışma mantığı

NET Framework çalışma mantığını anlatmadan önce derleme kavramını kısaca gözden geçirelim. Bir program herhangi bir programlama diliyle yazıldıktan sonra bilgisayarın anlayacağı dil olan makine diline (0 ve 1'lerden oluşan dil) çevrilir. Bu işleme derleme (compile) denir. Programın çalışması derleme işleminden sonra gerçekleşir. Kısacası, program derlenip bilgisayarın anlayacağı hale gelince çalıştırılır.



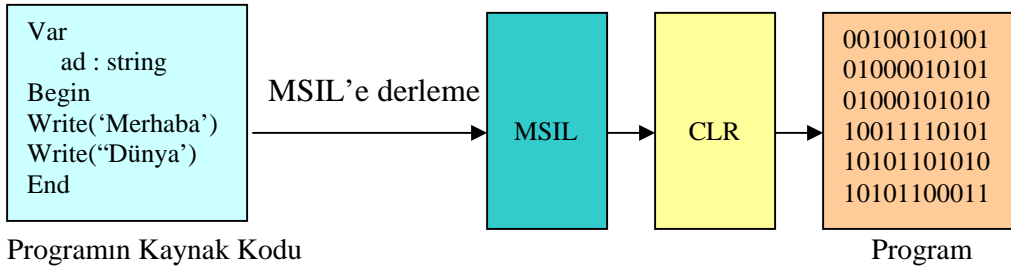
Şekil 1.2: Geleneksel yöntem

Fakat her derlenmiş program farklı yapılarıdaki bilgisayarlarda çalışmaz. Örneğin, Macintosh bilgisayarlarda derlenmiş bir program PC (Personel Computer=Kişisel Bilgisayar) yapısındaki bilgisayarlarda çalışmaz. Eğer o programı PC’de de çalıştırmak istiyorsak, programı yeniden bu bilgisayara göre derlemeniz gerekir.



Şekil 1.3 : .NET Framework’ün çalışma mantığı

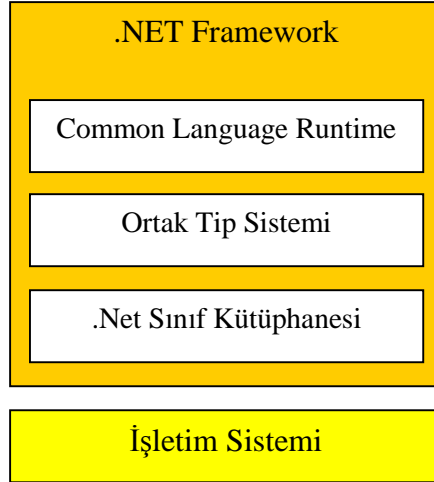
Yukarıdaki şekilden hareketle .NET Framework’ün çalışma mantığını inceleyelim. .NET Framework’te farklı dillerde yazılmış programlar kendilerine özgü, uygun derleyiciler kullanılarak MSIL adı verilen orta düzey bir dile (ara dile) çevrilir. MSIL dilinin temelini Assembly (montaj) oluşturur. MSIL diline dönüştürülen program çalıştırıldığında, CLR, Just In Time Compiler kullanarak program kodlarını makine diline çevirir. Bu işlem bir defaya mahsus gerçekleştirilir. Bu derlemenin sonucunda programınız işletim sistemlerinden bağımsız hale gelir, platform bağımsız olarak çalıştırılabilir. Özetle, .NET Framework’le platform ve programlama dili bağımsız programlar geliştirmek hedeflenmektedir.



Şekil 1.4: .NET Framework'ün kullandığı yöntem

.NET Framework 3 temel bileşenden oluşur.

1. Common Language Runtime
2. Ortak Tip Sistemi (Common Type System)
3. .Net Sınıf Kütüphanesi (.Net Class Library)



Şekil 1.5: .NET Framework'ün bileşenleri

Common Language Runtime'ın görevi .Net uygulamalarını yüklemek ve çalıştırmaktır. Ortak tip sistemi, tüm dillerdeki değişken ve referans tipleriyle uyumlu olacak şekilde oluşturulmuş tipleri içeren sistemdir. .NET Sınıf Kütüphanesi, uygulama geliştirirken ihtiyaç duyulabilecek, nesneye dayalı sınıfların yer aldığı kütüphanedir.

Peki nesne (object) ve sınıf (class) kavramları ne demektir?

Programcılar, program yazmaya başladıklarında programla ilgili her öğeyi (Örneğin, pencereler, menüler, düğmeler vs.) kendileri yazmazlar. Eğer her programcı bunları kendisi yazmaya kalkışır, zaman ve emek gereksiz yere harcanmış olur. Çünkü her programda ortak olarak kullanılacak bir öğe programcılar tarafından en baştan, tekrar tekrar yazılmış olur. İşte bu durumda önceden hazırlanmış programlar kullanılır ki bunlara nesne adı verilir. Programcı pencere, menü, düğme, iletişim kutusu vb. gibi nesnelerin bir kopyasını kendi programına dahil eder, bu nesneler üzerinde kendi geliştirdiği programa özel değişiklikler yapar, nesnelere kendi programına uyumlu hale getirir. Örneğin, pencere nesnesinin

boyutlarını, görünümünü değiştirir veya iletişim kutusundaki mesajın içeriğini değiştirir. Tüm programlar nesnelere kullanılmaktadır. Örneğin, Windows işletim sistemi bağımsız ve tekrar kullanılabilen nesnelere oluşur.

Programcı için, kendi programına dahil edeceği pencere veya menü nesnesinin hangi programlama dilinde yazıldığı önemli değildir. Programcuyu ilgilendiren bu nesneyi kendi programına nasıl tanıyacağı ve çalıştıracağını bilmektedir. Bu durum, programcıları ürettikleri programları paylaşabilme ve geliştirebilme yolunu açmıştır. Nesnelere, özelliklerini ve yapacakları işlevi ait oldukları sınıftan alırlar. Yani her nesne bir sınıfa aittir ve o sınıftan türetilirler.

Özetle programcılar geliştirdikleri programda pencere, düğme gibi program parçalarını oluşturmak için program içine kodlar yazmazlar. Bu kodları sınıflar içerisine yazırlar, böylelikle sınıflar oluşturmuş olurlar. Daha sonrada bu sınıflardan ürettikleri nesnelere vasıtasıyla bu program parçalarını kullanırlar. Bir işlem için her defasında kod yazmak yerine o işlemi yerine getiren nesneyi programlarına dahil ederler. Bu tarz programlamaya nesneye yönelik (Nesne tabanlı) programlama adı verilir.

.NET Framework'te karşılaşılacağımız diğere bir kavram ad alanı (namespace) kavramıdır. Farklı kaynaklarda namespace kavramı için "ad uzayı", "ad ortamı" ifadeleri kullanılmaktadır. .NET Framework birçok ad alanının toplamından oluşur. Ad alanı ise benzer özelliklerine göre gruplandırılmış sınıflardan oluşur. Yani, aynı isim altındaki sınıflar, o isimle bir ad alanını oluştururlar. ASP.NET, tüm uygulamalar için gerekli gördüğü ad alanlarını, uygulamalara otomatik olarak ekler. Uygulama geliştiricisi, hazırladığı uygulama için ayrıca bir ad alanı kullanmak isterse, uygulamasına bu ad alanını eklemelidir.

1.4. İnternet Uygulamalarında İşleyiş Modelleri

İnternet üzerinde tüm uygulamalar istemci-sunucu modeline göre çalışır. Yani, istemci bilgisayar internet üzerinden bir istekte/talepte bulunur, sunucu bilgisayar bu isteğe cevap verir. Tarayıcınızda bir internet adresi yazmanız bir istektir. Bu internet adresini barındıran sunucu bu siteyle ilgili bilgileri tarayıcınıza göndererek talebinize cevap vermiş yani bu isteği karşılayan hizmeti size sunmuş olur.

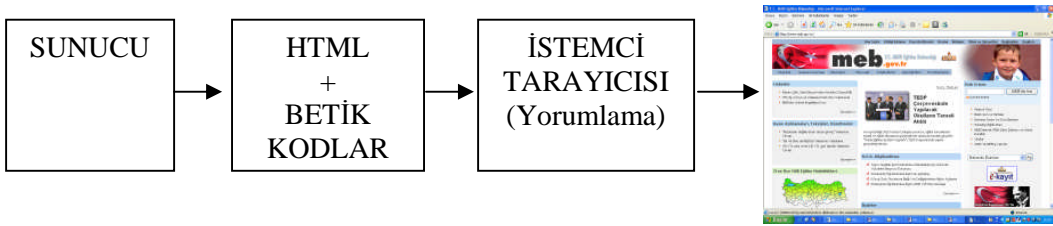
En basit internet sayfası sadece HTML etiketlerinden (tag) oluşur. Zamanla internet sayfalarını daha işlevsel yapmak amacıyla HTML etiketleri dışında çeşitli kodlar kullanılmaya ve internet sayfalarının çalışma mantığında değişiklikler yapılmaya başlanmıştır.

1.4.1. İstemci Tarafı (Client-Side) Kodlama

Web sayfalarının istemci üzerinde çalıştığı sistemdir. Sunucu bilgisayarlar HTML dosyalarını bulundurlurlar, barındırırlar (hosting işlemi). Bir kullanıcı internet tarayıcısından bu sayfanın internet adresini girdiğinde (sunucuya talep geldiğinde), sunucu bu sayfanın bir kopyasını istemciye gönderir. Dikkat edilirse, sunucu sayfa üzerinde hiçbir işlem yapmadan sayfanın bir kopyasını istemciye gönderir. İstemci bilgisayardaki tarayıcı, gelen HTML kodlarını yorumlayarak gösterir.

HTML ile yapılan basit sayfaları geliştirmek, daha fazla amacı gerçekleştirmek niyetiyle betik (script) diller kullanılmaya başlanmıştır. Bu sayede net sayfaları daha işlevsel, etkileşimli olmuşlardır. Yani kullanıcı sadece internet sayfalarını gören kişi olmaktan çıkıp, sayfayla etkileşim kuran kişi duruma gelmiştir. Örneğin, düz HTML sayfasını inceleyen bir kullanıcı, sayfadaki bilgileri okur, resimlere bakar, animasyonları seyrederek. Fakat betik dillerin eklendiği bir HTML sayfasında, kullanıcının seçimlerine, isteğine göre değişen menüler kullanılabilir, resim ve animasyonlar seyredilebilir. Böylelikle kullanıcı sayfayla etkileşim, iletişim halindedir.

İstemci taraflı kodlamada (kullanıcı taraflı kodlama da denir), betik dillerle güçlendirilmiş net sayfaları eskiden olduğu gibi istemci bilgisayarda çalışır. Sunucu hala kendisinden talep edilen sayfada hiçbir işlem yapmadan istemciye gönderir. İstemci taraflı HTML kodlarıyla beraber betik kodları da yorumlar, kullanıcıya gösterir. Dikkat edilirse yapılan işlemler tamamıyla istemci bilgisayarda gerçekleşir ve sunucu bilgisayarın istemci bilgisayardaki işlemlerle ilişkisi yoktur. Sunucu sadece kendisinden bir sayfa talep edildiğinde o sayfanın bir kopyasını istemciye göndererek bu talebe cevap verir.



Şekil 1.6 : Sayfanın istemci tarayıcısında yorumlanıp gösterilmesi

1.4.2. Sunucu Taraflı (Server-Side) Kodlama

Web sayfalarının sunucu üzerinde çalıştığı sistemdir. Bu sayfalarda kullanılan betik kodlar, özel belirteçlerle sunucuda çalışan kodlar olduklarını belirtirler. Net sayfaları web sunucuda çalıştırdıktan, işlendikten sonra, HTML biçiminde istemciye gönderilir. Bu durumda sunucu pasif durumdan aktif duruma geçmiştir. Kullanıcıdan gelen bilgilerin sunucuda işlenmesi imkanı web sitesi sahibinin işlerini kolaylaştırmakta, site üzerindeki denetimini artırmaktadır.

Kodlar istemcide çalışmadığından istemci bilgisayarlarda performans artışı olur. Tabii ki, hızla çalışan sayfalar görebilmeniz için sayfayı sunan sunucu bilgisayarın yüksek performanslı olması gereklidir.

Sunucu taraflı programlamanın bir diğer avantajı güvenlidir. İnternet tarayıcınızda istemci taraflı bir sayfanın kaynağını görüntüleyebilirken, sunucu taraflı bir sayfanın kaynağını görüntüleyemezsiniz. Kaynağı görüntüleyemediğinizden uygulamanın açıklarını bulmanız da zorlaşır. Sunucu taraflı uygulamaların daha güvenli olması elektronik ticaret (e-ticaret) kavramını geliştirmiştir.

1.4.3. Baęlantısızlık ve Olay Yönlendirmeli (Eylem Yönelimli, Event Driven) Programlama

İnternet uygulamalarının işleyiş modeli olan istemci-sunucu (istek/cevap) modelinde istemci ile sunucu arasında sürekli bir baęlantı yoktur. Birisiyle telefonda görüştüğünüzde konuşmanızı bitirene kadar aranızda bir baęlantı vardır. Ancak internette istemci ile sunucu arasında doğrudan (direkt) bir baęlantı yoktur. Sunucu, istemcinin isteklerini, istemciyi belirleyen IP adresini de ekleyerek paket halinde internet ortamına gönderir. Bu istek internet üzerinde belli merkez noktalardan geçerek istemciye ulaşır.

Sunucuyla istemcinin birbirleriyle sürekli baęlantı halinde olmamasından dolayı, istemci herhangi bir istekte bulunmadıkça, sunucu istemci bilgisayarda meydana gelen deęişiklikleri bilemez, takip edemez. Fakat web sitesi geliřtiricileri istemcideki geliřmelerden haberdar olmak isterler.

ASP.NET olay yönlendirme mantığıyla bu sorunu çözmüřtür. Nesneye yönelik programlamada kullanılan bu kavram kullanıcının her hareketinin bir “olay” olarak deęerlendirilmesi mantığına dayanır. Örneęin, kullanıcının bir düğmeye basması, bir onay kutusunu işaretlemesi, bir seçeneęi tıklaması, farenin hareket etmesi, klavyeden bir tuřa basması bir olaydır. Programcı bir olay gerçekteřtięinde yapılmasını istedięi işleri kodlayarak programına ekler. Böylelikle program kullanıcının hareketini ifade eden olayın gerçekteřmesi sonucunda kullanıcının isteęini yerine getirir.

ASP.NET sayfaları içerisine hem sunucu hem de istemci tarafta çalışan kodlar eklenir. İstemci tarafındaki kodlar olay yönlendirme mantığıyla istemci bilgisayarda gerçekteřen işlemleri takip ederek, bu işlemleri sunucuya bildirir. Bu şekilde sunucu, istemcideki her hareketi izler. İnternet uygulamaları geliřtirenler de istek/cevap modelinin sıkıntılarını hissetmeden, uygulamalarını geliřtirebilmektedirler. Sonuç olarak nesneye yönelik programlama teknięini kullanan ASP.NET, olay yönlendirme kavramıyla istemcideki deęişiklikleri takip edebilmektedir.

Ayrıca olay yönlendirmeli programlama sayesinde sayfadaki bir deęişiklikte tüm sayfa üzerinde işlem yapmak yerine hangi olay gerçekteřmişse sadece o olaylar ilgili işlem yapılmaktadır. Bu durum aynı anda yüzlerce siteyi barındıran sunucuları, tüm sayfayı tekrar tekrar çalıştırma işleminden kurtarmaktadır.

1.5. ASP.NET Sayfalarının Çalıştırılması

ASP.NET sayfalarınızı “Not Defteri” gibi basit bir metin düzenleyicisi (editör) kullanarak oluşturabilirsiniz.. Ancak Html, ASP.NET kodlarını tanıyıp farklı renklerle gösteren bir metin düzenleyici kullanmanız işinizi kolaylařtıracaktır. Ařaęıda ASP.NET sayfasını oluşturan ifadelerin çeřitli programlardaki renklendirilmesi gösterilmektedir.

```

<%@ page language="VB" codepage="28599"%>

<script runat="server">
sub page_load(obj as object, e as eventargs)
metin.text ="ASP.NET Dünyasına Hoşgeldiniz"
end sub
</script>

<html><head>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-9">
</head><body>
<asp:label id="metin" runat="server" />
</body>
</html>

```

Resim 1.12: Dreamweaver MX2004 programında kodların görünümü

```

<%@ page language="VB" codepage="28599"%>

<script runat="server">
sub page_load(obj as object, e as eventargs)
metin.text ="ASP.NET Dünyasına Hoşgeldiniz"
end sub
</script>

<html><head>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-9">
</head><body>
<asp:label id="metin" runat="server" />
</body>
</html>

```

Resim 1.13 : Homesite+ programında kodların görünümü

```

1  <%@ page language="VB" codepage="28599"%>
2  <script runat="server">
3  sub page_load(obj as object, e as eventargs)
4  metin.text ="ASP.NET Dünyasına Hoşgeldiniz"
5  end sub
6  </script>
7  <html>
8  <head>
9  <meta http-equiv="content-type" content="text/html; charset=iso-8859-9">
10 </head>
11 <body>
12 <asp:label id="metin" runat="server" />
13 </body>
14 </html>

```

Resim 1.14 : Visual Studio programında kodların görünümü

İsterseniz web uygulamaları tasarlamayı, görsel işlemler yapmayı, kodlamayı, geliştirmeyi kolaylaştıran Dreamweaver, Frontpage, Homesite+, Visual Studio.NET, Web Matrix gibi programları kullanabilirsiniz. Sık kullanılan bu programların dışında da birçok program mevcuttur.

Visual Studio.NET, .NET uygulamaları geliřtirmek için çok uygun bir ortam sunar fakat ASP.NET sayfalarını çalışmak için zorunlu olarak Visual Studio .NET kurulması gerekmez. ASP.NET sayfaları oluşturmak için kullanılan programların demo veya paylaşımli yazılım (shareware) hallerini kullanıp program hakkında bilgi edinebilirsiniz. Fiyat/performans ve kullanım rahatlığı açısından hangisini uygun buluyorsanız o programı seçersiniz. Modüllerde ASP.NET sayfaları Not Defteri kullanılarak oluşturulacaktır.

Ařağıdaki kodları Not Defteri programını açarak yazınız. Programı oluřturan kodların rahat okunmasını saėlamak amacıyla çeřitli yerlerde boş satırlar bırakılmıřtır. Bu satırların program için özel bir anlamı yoktur. Siz de rahat kod yazmanızı ve okumanızı saėlayacak şekilde çeřitli düzenlemeler yapabilirsiniz. Örneėin, kimi satırları içeriden bařlayarak farklı bölümler oluşturabilirsiniz. Kodların saė tarafındaki numaralar, sayfanın açıklaması yapılırken bahsedilen satırın kolay bulunması için konulmuřtur. Dolayısıyla dosyayı oluřtururken yazmayınız.

```
1 <%@ page language="VB" %>
2 <script runat="server">
3     sub page_load(obj as object, e as eventargs)
4         metin.text ="ASP.NET Dünyasına Hořgeldiniz"
5     end sub
6 </script>
7 <html><body>
8 <asp:label id="metin" runat="server" />
9 </body></html>
```

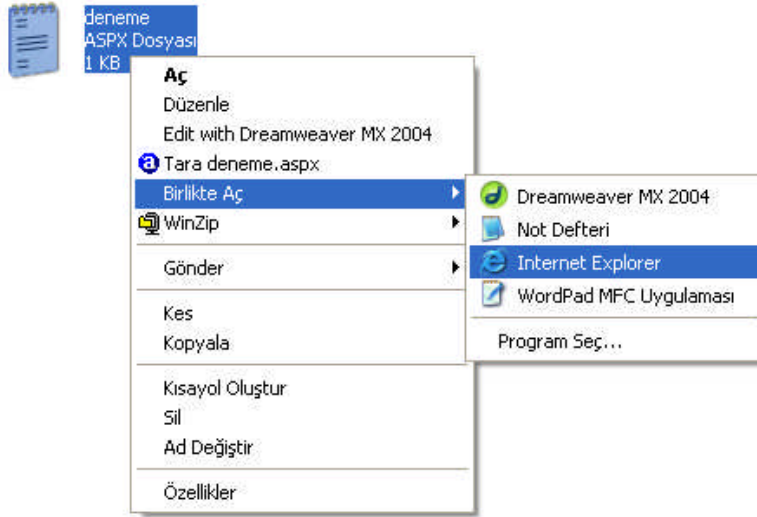
Dosya menüsünden Kaydet komutunu çalıştırıp Farklı Kaydet penceresine ulaşınız. **Konum** olarak D sürücüsünü seçiniz. Dosya adı kısmına **deneme.aspx** yazıp, kodlama kısmından **UTF-8**'i seçerek dosyayı kaydediniz. Örnek ASP.NET sayfası ekranda **ASP.NET Dünyasına Hořgeldiniz** yazısını görüntüleyen bir sayfadır. Sayfayı oluřturan kodların yapısı, görevleri, işleyiş mantığı daha sonra anlatılacaktır.

DİKKAT : Modüller boyunca oluřturduėunuz dosyaları sabit diskinizde uygun bir yerde depolayınız. İleride bu dosyalarla çalışmak gerektiėinde tekrar yazmakla vakit kaybetmezsiniz. Ayrıca evde bilgisayarınız varsa, hem evde çalışmak hem de dosyalarının bir yedeėini bulundurmak açısından disket, CD, flař disk, taşınabilir disk vb. ortamlar kullanarak dosyalarınızın yedeklerini ev bilgisayarınızda tutabilirsiniz. Deepfreeze benzeri programlar kullanıyorsanız dosyalarınızı kaybolmayacak bir sürücüye kaydediniz.

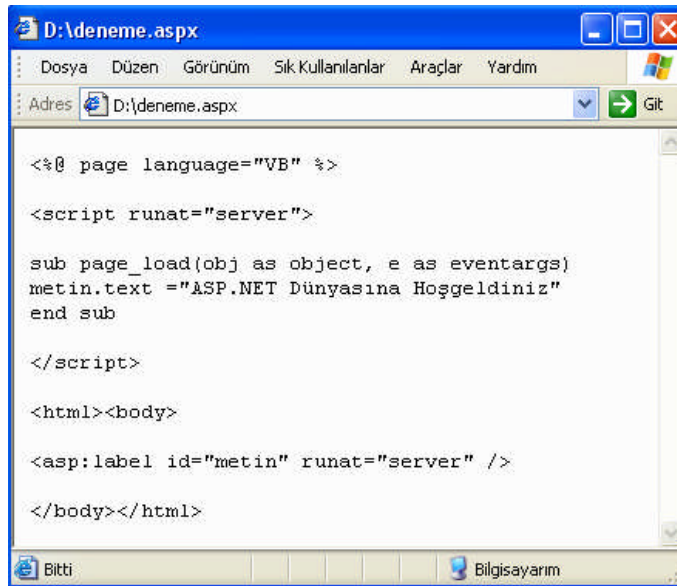
Bir sayfadan yararlanarak başka sayfalar oluřturmamak amacıyla ASP.NET sayfasını Not Defterinde açıp, Farklı Kaydet komutu ile farklı bir isim vererek yeni bir dosya oluřturabilirsiniz. Oluřturduėunuz bu dosyada istediėiniz satırları deėiřtirip hızlıca yeni ASP.NET sayfası geliřtirebilirsiniz.

Şimdi gelelim ASP.NET sayfalarının çalıştırılması mantığına. Acaba geliştirdiğiniz ASP.NET sayfalarını, bir HTML sayfalarını çalıştırdığınız gibi web tarayıcının adres çubuğundan dosyanın adını yazarak veya dosyanın üzerine çift tıklayarak çalıştırabilir miyiz? Deneyelim görelim.

Yukarıdaki dosyayı, üzerine çift tıklayarak veya web tarayıcının adres çubuğundan çağırarak çalıştırdığınızda istenilen görüntü elde edilemeyecektir. Aspx uzantılı dosyalarınızı başka bir programla ilişkilendirdiyse o program çalışacaktır. Eğer dosyanızı aşağıda gösterildiği gibi Internet Explorer gibi bir tarayıcıyla açmak isterseniz aşağıdaki görüntü karşınıza çıkacaktır. **İstenilen mesaj değil yazılan tüm komutlar ekranda gözükecektir.** Kimi ASP.NET dosyaları çalıştırıldığında ekranda hiçbir şey gözükmeyebilir.



Resim 1.17 : Deneme.aspx dosyasının Internet Explorer'la açılması

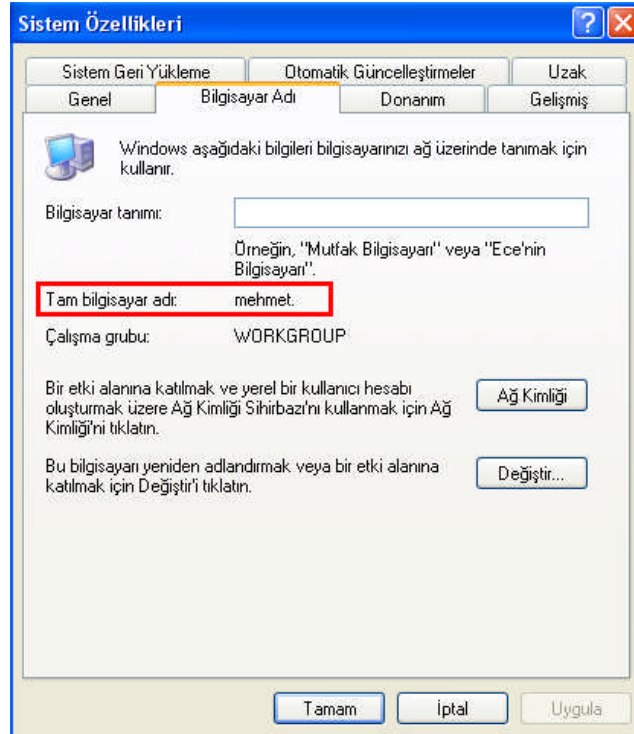


Resim 1.18 : Deneme.aspx dosyasının Internet Explorer'daki ekran görüntüsü

Görüldüğü gibi “aspx” uzantılı ASP.NET dosyaları HTML dosyalarının çalıştırıldığı gibi çalıştırılmamaktadır. Çünkü aspx dosyaları, istemci bilgisayardaki tarayıcı program tarafından değil, sunucu bilgisayar tarafından çalıştırılmalıdır. Sunucu, dosyaları çalıştırmalı ve istemci bilgisayara göndermelidir. Daha önce anlatılan bilgisayara IIS programını kurmanın amacı burada kendini göstermektedir. IIS ile bilgisayar sunucu bilgisayar gibi çalıştırılmaktadır.

IIS, sunucu taraflı sayfalarının, uygulamaların çalıştırılacağı varsayılan (default) bir klasör belirler. Bu varsayılan klasör **C:\inetpub\wwwroot** klasörüdür. Bu klasöre kaydettiğiniz dosyalar sunucuda çalıştırılacak dosyalar olarak değerlendirilecektir. Buradaki dosyaları tarayıcınızın adres çubuğundan girerek çağırmaya çalışmayınız, daha öncede belirtildiği gibi ASP.NET uygulamaları bu şekilde çalıştırılmazlar. Dosya çalışsa dahi tam olarak istenilen sonuç elde edilemez.

ASP.NET sayfaları, sunucu adıyla birlikte sayfanın adresi belirtilerek çalıştırılır. IIS kurulduktan sonra sunucu bilgisayar olarak çalıştıracağınız bilgisayarınızın sunucu ismi, Windows'u kurarken girdiğiniz bilgisayar ismiyle aynıdır. Bilgisayarınızın dolayısıyla sunucunuzun ismini şu şekilde öğrenebilirsiniz. Masaüstündeki **Bilgisayarım** simgesine **sağ tıklatınca** gelen kısayol menüsünden **Özellikler** komutunu tıklatın. Karşınıza gelen pencerede **Bilgisayar adı** sekmesine tıklatın. Gelen pencerede bilgisayarınızın ismini göreceksiniz.



Resim 1.19 : Sistem özellikleri

Sunucu adı olarak bilgisayar adı yerine **localhost** ifadesini de kullanabilirsiniz. Hem bilgisayar adı hem de localhost ifadesi wwwroot klasörünü referans alır.

ASP.NET sayfalarını çalıştırma yöntemi aşağıdaki şekildedir. (Tarayıcınızın adres çubuğuna yazılacak.)

```
http://bilgisayarınızın adı/ASP.NET sayfası adı
```

veya

```
http://localhost/ASP.NET sayfası adı  
(http://localhost/sayfaadi.aspx)
```

Örnek:

```
http://mehmet/deneme.aspx  
http://localhost/deneme.aspx
```

NOT : Tarayıcınızda **http://** ifadesini kullanmadan da yukarıdaki ifadeleri yazarsanız aynı sonucu elde edersiniz.

NOT : **localhost** ifadesi bilgisayarınızın yerel IP numarası olan 127.0.0.1 adresine karşılık gelir. Sayfalarınızı **http://localhost/sayfaadi.aspx** şeklinde veya **http://127.0.0.1/sayfaadi.aspx** şeklinde çağırmanız arasında fark yoktur.

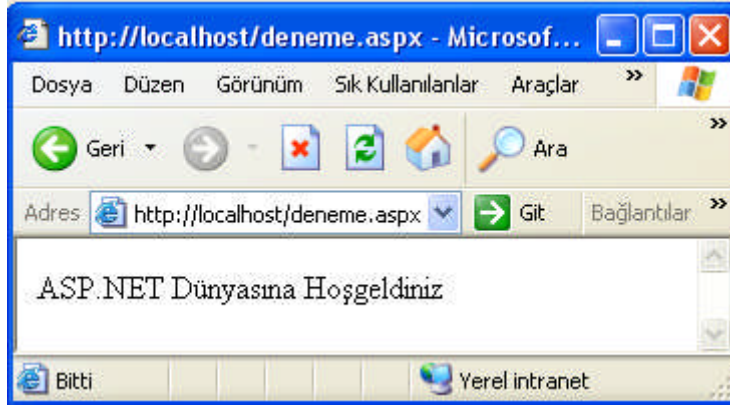
Yukarıdaki ifadelerin her ikisi de, sunucudan **C:\inetpub\wwwroot** klasörü içindeki **deneme.aspx** dosyasını çalıştırmasını istemektir. Sunucu bu dosyayı çalıştıracak, sonuçları HTML biçiminde tarayıcınıza gönderecektir, tarayıcı da sayfayı görüntüleyecektir. Unutmayın, şu an sizin bilgisayarınız hem sunucu hem istemci bilgisayar olarak çalışmaktadır.

Eğer ASP.NET sayfası **C:\inetpub\wwwroot** içindeki bir alt klasörlerde bulunuyorsa, bu klasörleri de belirtmeniz gerekir. Örneğin,

```
http://mehmet/uygulamalar/form1.aspx (form1.aspx dosyası  
uygulamalar klasörü içindedir.)
```

```
http://localhost/resimler/manzara/doga.aspx (doga.aspx sayfası  
resimler klasörünün alt klasörü olan manzara klasörü  
içindedir.)
```

Şimdi daha önce D sürücüsüne kaydettiğiniz deneme.aspx dosyasını C:\inetpub\wwwroot klasörü içerisine kopyalayınız. Tarayıcınızı açıp adres çubuğuna **http://localhost(veya bilgisayarınızın adı)/deneme.aspx** adresini yazarak Enter tuşuna basınız.

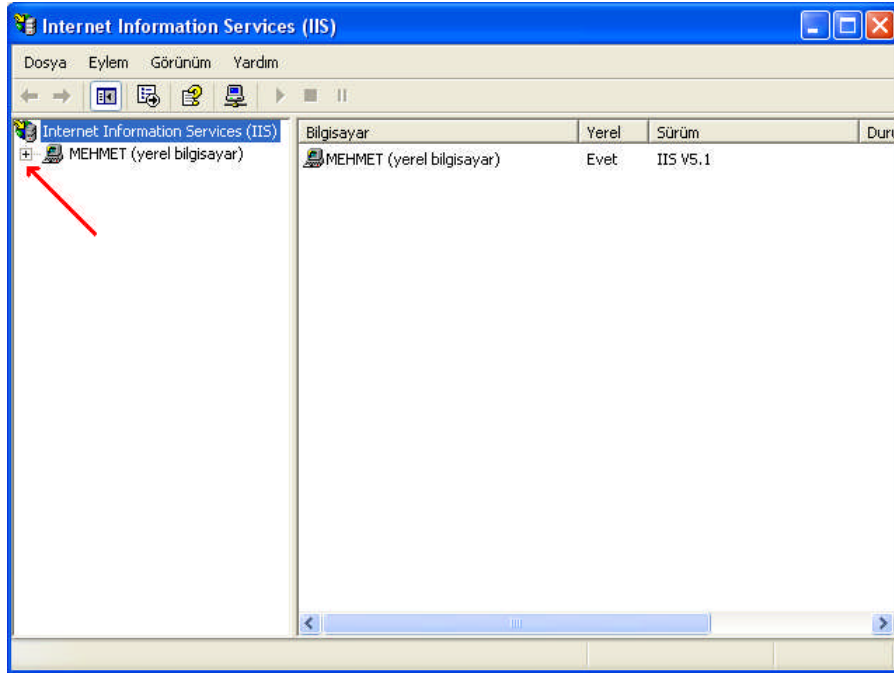


Resim 1.20 : deneme.aspx dosyasının ekran görüntüsü

İşte ilk ASP.NET sayfasını çalıştırdınız. Bu kısımda dikkat edilecek nokta, deneme.aspx dosyası için bir fiziksel yol (adres) bir de sanal yol tanımının olduğudur. Bu dosyanın diskinizdeki fiziksel adresi C sürücündeki **inetpub** adındaki klasörün alt klasörü olan **wwwroot** klasörüdür. **Fiziksel yol tanımı** ise C:\inetpub\wwwroot şeklindedir. Bu dosyayı çalıştırmak için kullanılan **sanal yol tanımı** ise http://localhost/deneme.aspx'dir. Görüldüğü üzere sanal yol internet adres yapısına uygundur. Her sanal yol bir fiziksel adrese karşılık gelir. Özetle, siz dosyayı çalıştırmak için tarayıcınızdan internet adresini yazıp, internet adresini görüntülemesi için IIS'den talepte bulunursunuz, bilgisayarınızı sunucu bilgisayar haline getiren IIS'de bu talebinizi yerine getirir ve sayfanız ekranda görüntülenir.

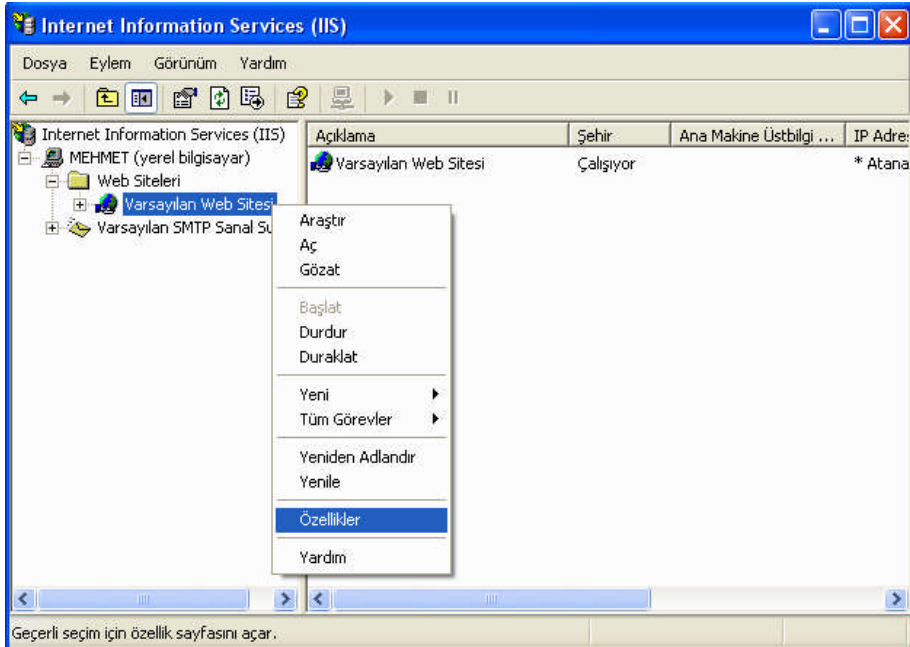
1.5.1. Web Sunucusu Yönetimi

IIS programının ASP.NET sayfalarının sunuculuğunu yapacağı klasör önceden tanımlı, varsayılan şekilde "C:\inetpub\wwwroot" klasörüdür. İsterseniz bu klasörü değiştirebilirsiniz. Örneğin, bu IIS'nin varsayılan klasörünü **D sürücüsünde Sayfalar klasörü** şeklinde değiştirelim. D sürücüsünde Sayfalar isimli bir klasör oluşturduktan sonra Denetim Masası\Yönetimsel Araçlar\Internet Information Services (IIS) programını çalıştırınız. (Başka bir yol Başlat menüsünde Çalıştır komutunu verip, gelen ekranda **inetmgr** yazıp Enter tuşuna basmaktır)



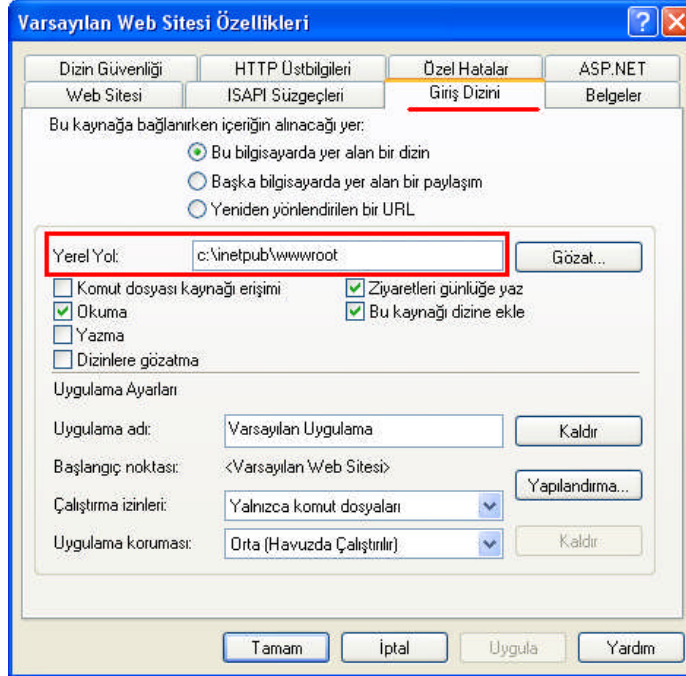
Resim 1.21 : Yerel bilgisayar simgesinin alt seçeneklerini açma

Bilgisayar adının yazılı olduğu seçeneğin solundaki “+” simgesine tıklatarak alt seçenekleri ekrana getiriniz



Resim 1.22 : Varsayılan web sitesinin özelliklerine bakma

Bu pencereden **Web Siteleri** seçeneğini çift tıklattıp, karşınıza gelen **Varsayılan Web Sitesi** seçeneği üzerine sağ tuşla tıkladığınızda gelen kısayol menüsünden **Özellikler** komutunu çalıştırınız.



Resim 1.23 : Yerel Yol

Bu pencerede **Giriş Dizini** sekmesine tıklatınız. Karşınıza yukarıdaki pencere çıkacaktır. Görüldüğü gibi yerel yol, C:\inetpub\wwwroot klasörüdür.

Bu penceredeki **Yerel Yol** ifadesi karşısındaki metin kutusunda yazılan C:\inetpub\wwwroot fiziksel adresi yerine **D:\Sayfalar** fiziksel adresini yazınız. Bu durumda **localhost** ifadesi D sürücüsündeki Sayfalar klasörünü ifade etmeye başlar. Yerel Yol tanımındaki değişiklik ASP.NET dosyalarınızı çalıştırmınızı etkilemez. Sadece artık ASP.NET sayfalarınızı bu klasöre kaydetmelisiniz. Daha önce oluşturduğunuz deneme.aspx dosyasını D:\Sayfalar klasörüne kopyalayın. Ardından tarayıcınızdan http://localhost/deneme.aspx ifadesini yazarak dosyayı çalıştırın. Dosyanın sorunsuz çalıştığını göreceksiniz. Artık, IIS için yerel yol C:\inetpub\wwwroot yerine D:\sayfalar olmuştur.

İşlerinizi kolaylaştırmak açısından yaptığımız web sitelerini farklı klasörler halinde tutmanız yararlıdır. Örneğin, bir kitabevi (firmanın adı A olsun) için yaptığımız site ile bir bilgisayar firması (firmanın adı B olsun) için yaptığımız siteyi farklı klasörlerde tutmanız organizasyon açısından işinize yarayacaktır. Böylelikle alt klasörlerde birbirine karışmayacaktır. Bu durumda bilgisayarınızda oluşturabileceğiniz klasör yapısı aşağıdaki gibi olacaktır.

A firması için ---C:\inetpub\wwwroot\A klasörü ve içinde de alt klasörler.

B firması için ---C:\inetpub\wwwroot\B klasörü ve içinde de alt klasörler.

Bu iki siteyi çalıştırırken de

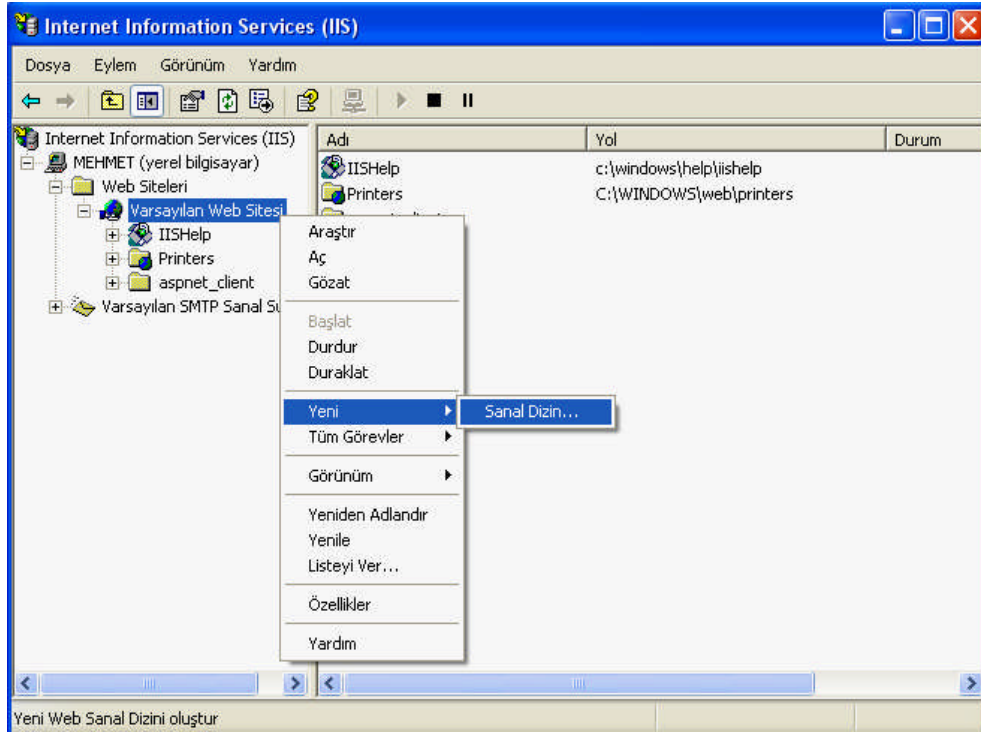
http://localhost/A/anasayfa.aspx

http://localhost/B/anasayfa.aspx satırlarını kullanırsınız.

1.5.2. Sanal Dizin (Virtual Directory)

Oluşturduğunuz web sitelerinizi web sunucusunda belirlediğiniz fiziksel klasörde/dizinde (c:\inetpub\wwwroot) tutmak zorunda değilsiniz. İsterseniz **Sanal Dizin** komutuyla diskinizdeki herhangi bir klasörü, sunucu klasörü yapabilirsiniz. Örneğin, D sürücünüzde oluşturduğunuz **Uygulamalar** isimli klasörü sanal dizin olarak belirleyeceğinizi varsayalım.

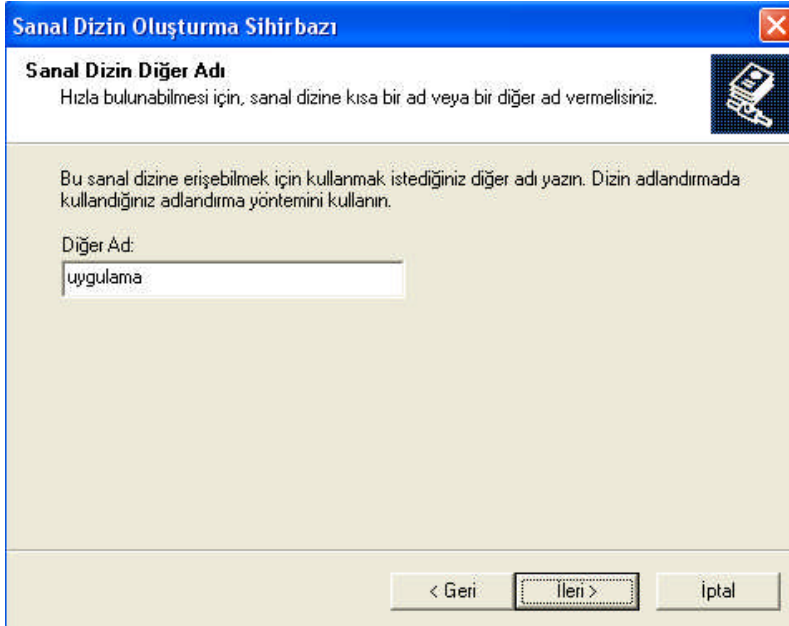
IIS programını çalıştırın. Bilgisayar adını gösteren seçeneğin solundaki “+” simgesine tıklatın. Bu pencereden **Web Siteleri** seçeneğini çift tıklatıp, karşınıza gelen **Varsayılan Web Sitesi** seçeneği üzerine sağ tuşla tıkladığımızda gelen kısayol menüsünden **Yeni** komutunu, ardından **Sanal Dizin** komutunu çalıştırın.



Resim 1.24 : Sanal Dizin



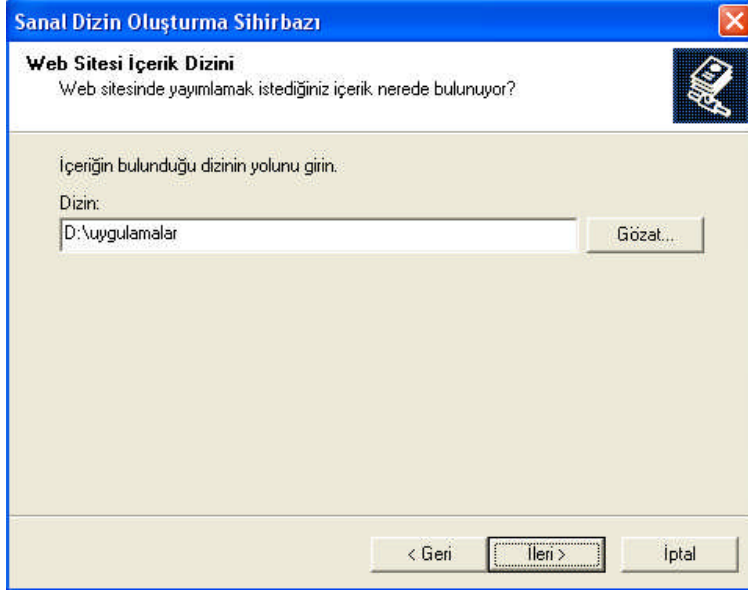
Resim 1.25 : Sanal dizin oluşturma sihirbazına hoş geldiniz penceresi
Karşınıza gelen **Sanal Dizin Oluşturma Sihirbazı** penceresinde **İleri** düğmesine tıklayın.



Resim 1.26 : Sanal dizin diğer adı penceresi

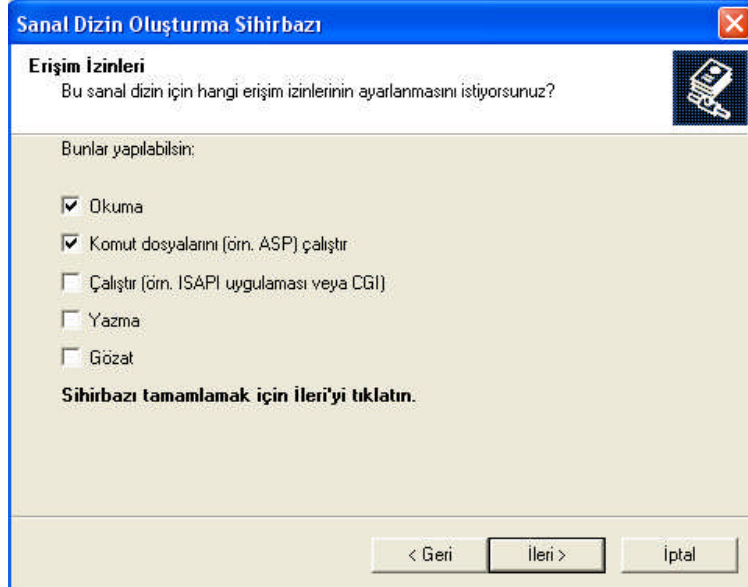
Sanal Dizin Diğer Adı başlıklı pencerede, oluşturacağınız sanal dizini temsil edecek bir ad yazılır. Bu ad bir nevi takma ad gibidir. Bu ad, yaptığınız web sitesinin içeriğini

tuttuđunuz klasörünüzü ifade edecek ve tarayıcı adres çubuğunda yazılacaktır. Diğer ad kutusuna **uygulama** ifadesini yazdıktan sonra **İleri** düğmesine basınız.



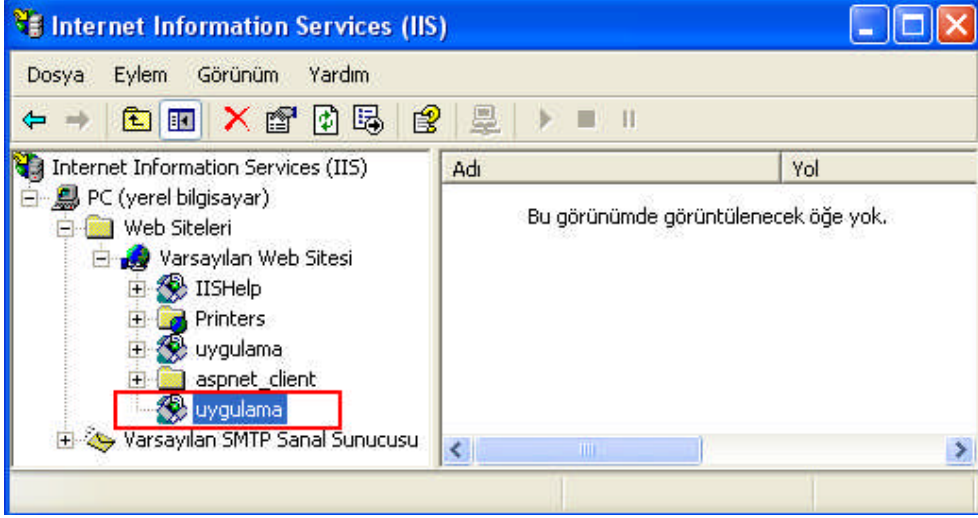
Resim 1.27 : Web sitesi içerik dizini penceresi

Oluşturacağınız web sitesinin içeriğini tutacağınız klasörün fiziksel adresi olan **D:\uygulamalar** ifadesini bu pencerede yazarak (veya Gözet düğmesiyle belirterek.) **İleri** düğmesine tıklattınız.



Resim 1.28 : Erişim İzinleri penceresi

Erişim izinleri penceresinde şu aşamada bir değişiklik yapmadan **İleri** düğmesine basınız. Karşınıza **Sanal Dizin Oluşturma Sihirbazını Başarıyla Tamamladınız** yazan pencere gelecektir. Bu pencerede **Son** düğmesine tıklatarak işlemi sonlandırınız.



Resim 1.29 : Uygulama diğer adlı sanal dizin

DİKKAT : D:\uygulamalar ifadesinin sabit diskte bir adresi, **uygulama** ifadesinin **diğer adı (alias)** ifade ettiğine dikkat ediniz.

NOT: Bazı kaynaklarda **Diğer ad** kavramı için **Takma ad** kavramı kullanılır.

Oluşturduğunuz sanal dizinin özelliklerini sonradan değiştirmek için sanal dizinin üzerinde sağ fare tuşuna tıklatınca gelen kısayol menüsünden **özellikler** komutunu kullanabilirsiniz.

Şimdi deneme.aspx dosyasını “D:\uygulamalar” klasörü içine kopyalayın. Ardından tarayıcının adres çubuğuna <http://localhost/uygulama/deneme.aspx> yazın. ASP.NET dosyanızın hatasız çalıştığını göreceksiniz. Dikkat ederseniz, c:\inetpub\wwwroot klasörü içindeki bir klasöre nasıl ulaşıyorsa, sanal dizine de o şekilde ulaşılmaktadır.

Etkileşimli web uygulamaları modüllerinde oluşturulan ASP.NET sayfaları “D:\uygulamalar” klasörüne kaydedilecektir. Siz çalışmalarınızı hangi klasörde tutmak istiyorsanız, IIS’nin tanımlı yerel yolunu değiştirerek veya sanal dizin oluşturarak farklı bir klasörü belirleyebilirsiniz.

Uygulama : Kişisel web sitenizi oluşturduğunuzu düşünelim. D sürücüsünde “Kendisitem” isimli bir klasör oluşturunuz. Bu klasörü “Sitem” **diğer adıyla sanal dizin** olarak belirleyiniz.

Uygulama : Bergama Anadolu Teknik, Teknik ve Endüstri Meslek Lisesi için bir web sitesi oluşturduğunuzu farzedelim. D sürücüsünde “Bergama_atteml” isimli bir klasör oluşturunuz. Bu klasörü “Bergama” **diğer adıyla sanal dizin** olarak belirleyiniz.

UYGULAMA FAALİYETİ

.NET yazılımının kurulumu, IIS kurulumu

Bilgisayarınızda yüklü olan .NET Framework yazılımını, ardından IIS'yi bilgisayarınızdan kaldırınız. Kaldırma işlemi tamamlandıktan sonra ilk önce IIS'yi, sonra .NET Framework yazılımını tekrar kurunuz.

Sanal Dizin (Virtual Directory) Oluşturma

a)D sürücüsünde oluşturacağınız “Projeler” isimli klasörü “Proje” **diğer adıyla** sanal dizin olarak ayarlayınız.

İşlem Basamakları	Öneriler
IIS programını çalıştırınız.	
Sanal Dizin komutunu veriniz.	Varsayılan Web Sitesi seçeneğini seçtikten sonra kısayol menüsünden Yeni komutunu veya Eylem menüsünden Yeni komutunu kullanabilirsiniz.
Sanal dizini temsil edecek diğer adı (proje) yazınız.	Diğer adın, yaptığımız web sitesinin içeriğini temsil edecek kısa ve özlü bir ad olmasına dikkat ediniz.
Web sitesinin içeriğini tutacağımız klasörün fiziksel adresini (D:\Projeler) belirtiniz.	Fiziksel adresi isterseniz klavyeden isterseniz Gözet düğmesini kullanarak belirtebilirsiniz.
Erişim izinlerini ayarlayınız.	Şimdilik varsayılan ayarları kullanmanız yeterlidir. Gerekğinde erişim izinlerini değiştirebilirsiniz.

b)İleride kuracağımız şirketiniz için bir web sitesi oluşturacaksınız diyelim. Bu web sitesinin dosyalarını bulduracağımız klasörü oluşturup, bu klasörü sanal dizin olarak ayarlayınız. (Şirket sizin olduğuna göre istediğiniz ismi verebilirsiniz. ☺)

ÖLÇME VE DEĞERLENDİRME

A- ÖLÇME SORULARI

Doğru/Yanlış Soruları

Aşağıdaki ifadeleri okuyunuz ve doğru olanlar ifadelerin önündeki boşluğa “D”, yanlış olan ifadelerin sonundaki boşluğa “Y” harfi koyunuz.

- 1) ASP.NET sayfalarının uzantısı aspx'dir. ()
- 2) ASP, Alternatif Sunucu Protokolleri kelimelerinin kısaltmasıdır. ()
- 3) ASP sayfası ile ASP uygulaması kavramları birbirinden farklı kavramlardır. ()
- 4) Sunucu taraflı programlamayla kullanıcıya özel internet sayfaları geliştirebilirsiniz.
()
- 5) İnternet sayfaları sadece istemci taraflı sayfalardan oluşmaktadır. ()
- 6) İnternet sayfalarını bilgisayarlarında tutan, barındıran firmalara hosting firmaları adı verilir. ()
- 7) Bir bilgisayar web sunucu programı kurularak sunucu bilgisayar haline dönüştürülür.
()
- 8) Nesne tabanlı programlama programcıların işlerini zorlaştırmıştır. ()
- 9) ASP.NET sayfaları Not Defteri programı dışında hiçbir programla oluşturulamaz.
()
- 10) IIS programı sanal dizin oluşturmaya izin vermez. ()

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrar tekrarlayınız

Tüm sorulara doğru cevap verdiyseniz diğer faaliyete geçiniz.

ÖĞRENME FAALİYETİ-2

AMAÇ

Programlama diline ilişkin ait kod yapısını tanıyarak, basit bir uygulama gerçekleştirebileceksiniz.

ARAŞTIRMA

İnternette sörf yaparken karşılaştığınız kimi sitelerde Türkçe karakterlerden ğ,ş,ü,ç,ö,ı gibi karakterleri göremiyoruz. Bunun sebebi ne olabilir? İnternette araştırma yaparak, bilgisayar uzmanlarına sorarak veya deneyimli arkadaşlarınıza danışarak vs. elde ettiğiniz sonuçlarınızı öğretmene teslim edecek veya sınıfta sunacak şekilde hazırlayınız.

2.1. Türkçe Karakterler

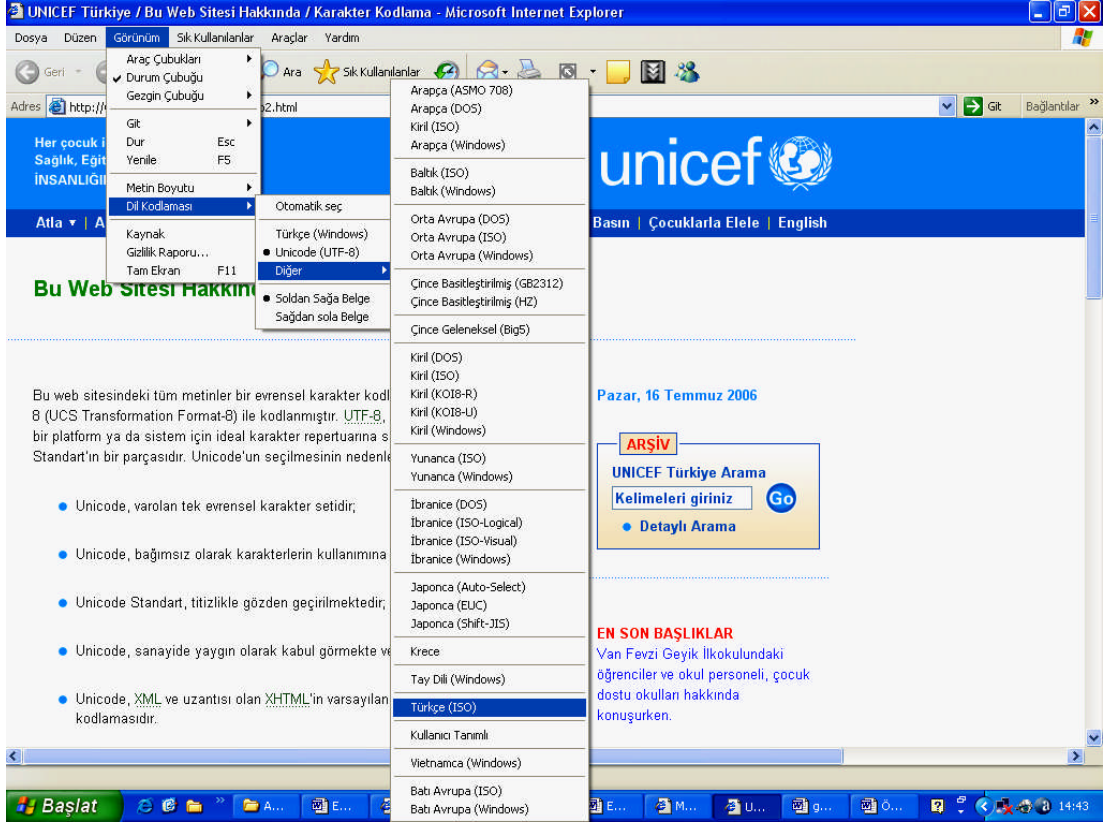
ASP.NET uygulamaları geliştirirken Türkçe karakterlerin kullanımı nasıl olacak? Bu konuyu karakterlerin bilgisayarda nasıl gösterildiği ve kodlama standartları konularını da ele alacak şekilde açıklayalım.

Bilgisayarda tüm karakterler (harfler, rakamlar, noktalama işaretleri, semboller) kodlanarak işlenir. Kodlama işlemi her karakterin bir numarayla (sayıyla) ifade edilmesi, yani o karaktere bir numaranın atanmasıdır. Karakterlerin sayısal karşılıklarıyla ilgili çeşitli standartlar vardır. Örneğin, sadece İngilizce karakterleri kapsayan ANSI standardı.

Fakat tüm dünya ülkelerinin kendi dillerinde programlar kullanma ihtiyacından dolayı, farklı ülke dillerindeki karakterleri de içerecek kodlama standartlara oluşturulmuştur. Yazılım üreticileri de bu kodlama standartlarına göre programlarını yazmaktadırlar. Farklı üreticiler kendi kodlamalarını oluşturabildikleri gibi (örneğin, Windows-1254 kodlaması.) uluslar arası organizasyonlar da çeşitli kodlamalar (örneğin, Türkçe'nin dahil olduğu karakter kodlama standardı, ISO-8859-9) oluşturmuşlardır. ISO-8859-9, belli başlı tüm taraftarlarda desteklenmektedir.

Her dilde farklı harfler kullanılır. Örneğin, Türkçe'de bulunan ş,ç,ö,ğ,ü,ı harfleri İngilizce'de yokken, İngilizce'de bulunan x, w, q harfleri, Türkçe'de yoktur. Bilgisayar dünyasında farklı dilleri kapsayan farklı kodlamalar oluşturulmuştur. Fakat her dilin farklı bir kodlamaya sahip olması, aynı sayfada birden fazla dil içeren sitelerde ve siteden siteye geçişlerde kod çakışmalarına neden olmuştur. İki farklı dil kodlamasında (encoding) aynı numaranın, farklı karakterleri tanımladığı durumlarda "ı" yerine "y" işaretinin çıkması, "ğ" yerine "?" işaretinin çıkması gibi kod çakışmaları meydana gelmiştir.

Yazılımcılar tarayıcılara **dil kodlaması** seçeneği koyarak farklı dillerin internet ortamında görüntülenebilmesine çalışmışlardır. Bu seçeneğin altında çeşitli dil kodlama standartları listelenmektedir.



Resim 2.1 : Internet Explorer’da Görünüm menüsündeki Dil kodlaması komutu

İnternette farklı dillere ait karakterleri görüntüleyememe problemine çözüm bulmak amacıyla birçok üreticinin desteklediği, **Unicode** adı verilen yeni bir kodlama tekniği geliştirilmiştir. Unicode, **Un**iversal **C**ode (Evrensel Kod) kelimesinin kısaltılmış halidir. “Yeryüzündeki her farklı karakter için farklı ve sabit bir numara, sayı atamak” yaklaşımıyla geliştirilen Unicode çatısı altında tüm dil kodlamaları toplanmaya başlanmıştır. Bu kodlama tekniğinde dünya üzerinde herhangi bir karakter için bir sayı değeri atandıysa, bu sayı değeri hiçbir şekilde farklı bir karaktere verilmeyecektir. Böylelikle tarayıcı, kendisine gelen bir sayısal değer net bir şekilde hangi karakteri ifade ettiğini bildiğinden karakterleri gösterememe problemi yaşamayacaktır.

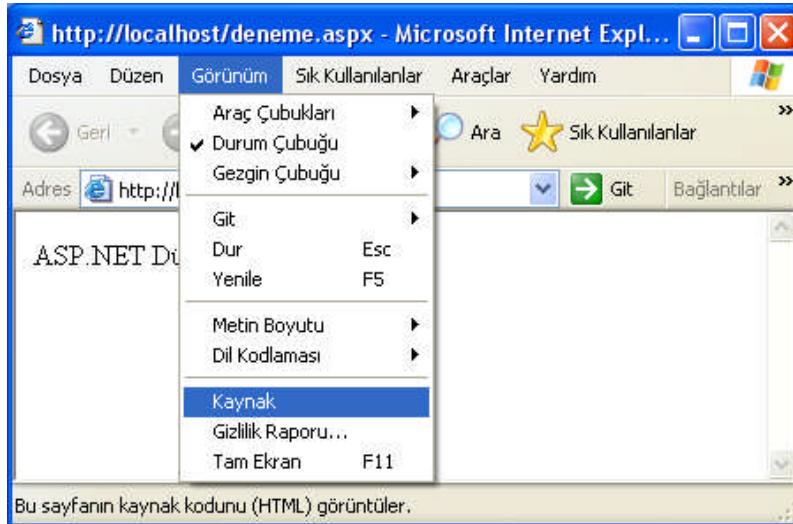
Tüm dillerdeki karakterleri kapsayacak kodlamalar geliştiren bir organizasyon olan Unicode organizasyonu Apple, HP, IBM, Microsoft, Sun, Oracle gibi yüzlerce yazılım şirketleri tarafından desteklenmektedir. Günümüzde halen kimi programlar Unicode (UTF-8)’i desteklerken, kimileri desteklememektedir. UTF-8 ile kodlanmış bir metin UTF-8’i desteklemeyen bir editör (örneğin, kodlama ayarı ISO olan bir editör) ile açıldığında kimi Türkçe karakterler görünmeyebilir. Tüm dünyada büyük bir hızla Unicode’u destekleyen

program sayısı artmaktadır. Tüm programlar Unicode (UTF-8)'i desteklediğinde Türkçe ve diğer dillere ait karakterlerin görüntülenmesinde bir sıkıntı yaşanmayacaktır.

ASP.NET'in tanımlı karakter kodlama tekniği Unicode (UTF-8) olduğundan ASP.NET, oluşturduğunuz sayfalardaki karakterleri farklı bir dil kodlaması belirtmediyseniz UTF-8 kodlamasına göre değerlendirir. Etkileşimli web uygulamaları modüllerindeki sayfalarda farklı bir karakter kodlaması belirtilmeyecek, böylelikle Unicode (UTF-8) kodlaması kullanılacaktır. İsterseniz UTF-8'in dışında Türkçe karakter kodlamasını içeren ISO-8859-9'u ve Windows-1254'ü kullanabilirsiniz. İnternet Explorer'daki dil kodlaması menüsünde ISO-8859-9'u Türkçe(ISO) olarak, Windows-1254'ü Türkçe(Windows) olarak görürsünüz.

2.2. Sayfa Yapısı

İnternet sayfaları hangi programlama diliyle yazılırlarsa yazılışın görüntülenmelerini sağlayan program tarayıcılarıdır. Tarayıcıların yorumlayabildikleri dil ise HTML (ve betik kodlardır, örneğin javascript, vbscript, jscript). Bu nedenle bir web uygulaması tarayıcıda görüntülediği sürece hangi programlama diliyle geliştirilirse geliştirin uygulamanın çıktıları HTML ve betik kodlar olacaktır. ASP.NET uygulamaları da çalıştırıldıktan sonra HTML ve betik kodlar şeklinde tarayıcıya gönderilirler. Tarayıcı da bu kodları yorumlayarak tarayıcı penceresinde görüntüler. Bir sayfanın bahsedilen HTML (ve betik kodlar) kodlarını sayfanızın **kaynak koduna** (source code) bakarak görebilirsiniz. Daha önce oluşturduğunuz deneme.aspx dosyasını çalıştırıp, aşağıdaki resimde gösterildiği gibi kaynak koduna bakınız.



Resim 2.2 : deneme.aspx dosyasının kaynak koduna bakma



Resim 2.3 : Örnek deneme.aspx sayfasının kaynak kodu

Başlangıç düzeyinde ASP.NET sayfa yapısını anlamak için aşağıdaki ASP.NET sayfasını inceleyelim. Bu sayfayı IIS'nin tanımlı yerel yolu olan C:\inetpub\wwwroot klasörüne **sayfaYapisi.aspx** ismiyle kaydedin. (Dosya adında Türkçe karakterlerin kullanımı çeşitli uyum sorunlarına neden olabilir düşüncesiyle dosyalara ad verilirken Türkçe karakterler kullanılmamıştır.)

sayfaYapisi.aspx

```
1 <%@ page language="vb" %>
2 <script runat="server">
3     sub tiklandi(sender as object, e as eventargs)
4         mesaj.text = "Merhaba " & metin_kutusu.text & " ASP.NET
5             sayfa yapısını öğrenmeye hazır mısınız?"
6     end sub
7 </script>
8 <html><body>
9     ASP.NET Sayfa Yapısı<p>
10    <%Response.write("Kutuya adınızı yazıp, Tıkla ve gör düğmesine
11        tıklayınız:<p>") %>
12    <form runat="server">
13        <asp:textbox id="metin_kutusu" runat="server"/>
14        <asp:button id="buton" text="Tıkla ve gör" onclick="tiklandi"
15            runat="server"/>
16    <p />
17    <asp:label id="mesaj" runat="server" />
18 </form>
19 </body></html>
```

NOT: Yukarıdaki sayfaYapisi.aspx örneğinde alt satıra geçen ifadeler Not Defteri programında aynı satıra yazılacaktır.

Bir ASP.NET sayfası temel olarak iki bölüme ayrılabilir.

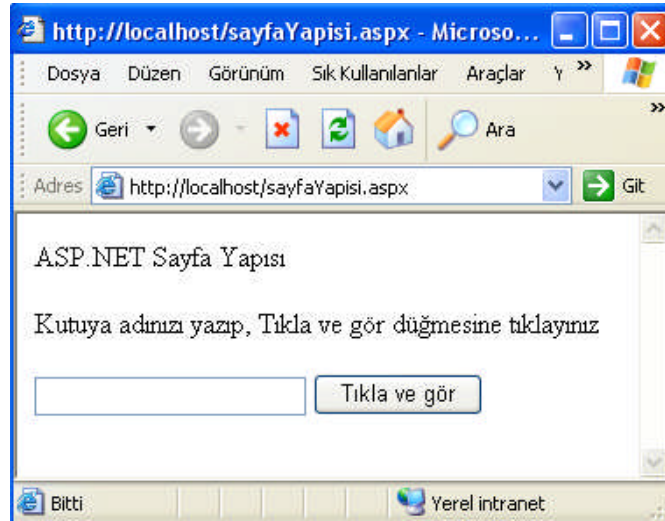
- 1)<script> </script> etiketleri arasında yazılan bölüm.
- 2)<html> </html> etiketleri arasında yazılan bölüm.

<script></script> etiketleri arasındaki bölüme ASP.NET sayfalarınızın dinamik, etkileşimli olmasını sağlayan, sayfanın işleyişini gerçekleştiren kodlar yazılır.

<html></html> etiketleri arasındaki bölüm ise, sayfanın sabit içeriğinin bulunduğu, sayfanın görsel kısmını oluşturan elemanların yerleştirildiği bölümdür. Bahsedilen her iki bölümde birbirleriyle ilişki içerisinde çalışırlar.

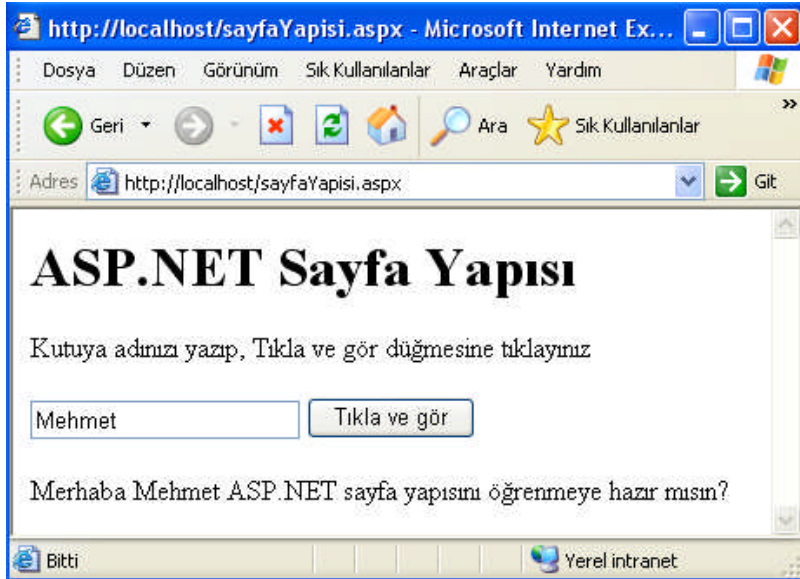
Gelişmiş web geliştirme programlarında uygulamanın görsel kısmı ayrı bir dosyada, kod kısmı ayrı bir dosyada tutulur. Yani yukarıda anlatıldığı gibi bölümler aynı dosya içinde değildir. Bu yöntem web uygulamasının görsel yüzünü tasarımcının, program mantığını programcının hazırladığı uygulamalarda çok yararlıdır.

Oluşturduğumuz sayfaYapisi.aspx dosyasını çalıştırarak nasıl çalıştığını görelim. Tarayıcınızın adres çubuğuna **http://localhost/sayfaYapisi.aspx** satırını yazarak, enter tuşuna basınız.



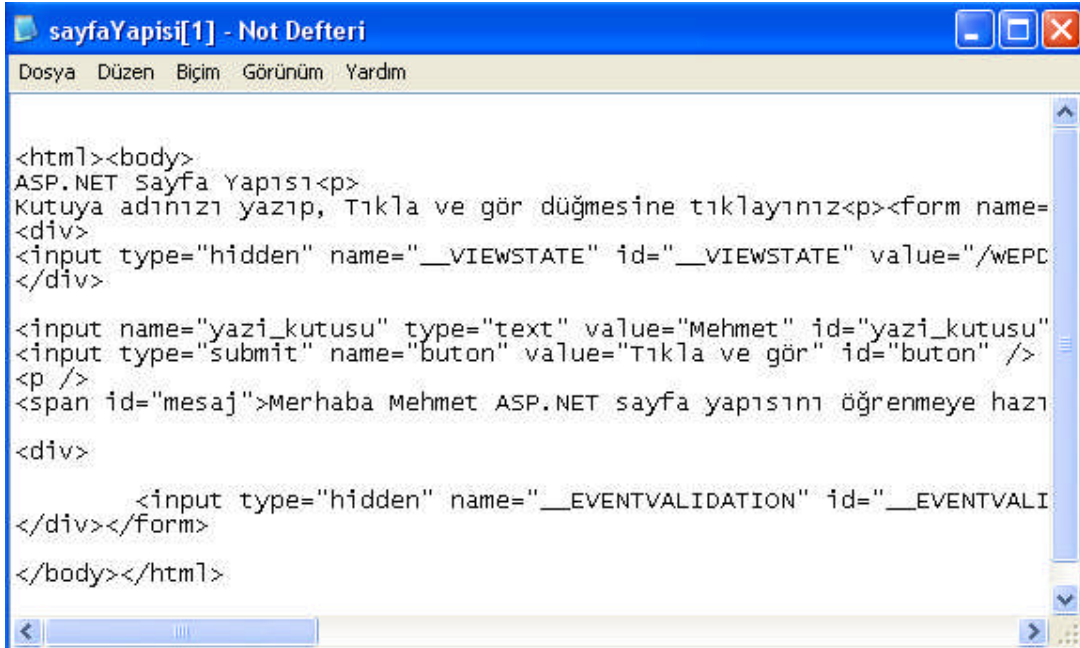
Resim 2.4 : sayfaYapisi.aspx sayfasının tarayıcıdaki görüntüsü

Sayfadaki metin kutusuna (textbox) adınızı yazıp, **Tıkla ve gör** düğmesine bastıktan sonra karşınıza aşağıdaki sayfa görüntüsü çıkacaktır.



Resim 2.5 : Düğmeye basıldıktan sonra sayfaYapisi.aspx sayfasının tarayıcıdaki görüntüsü

Çalıştırdığımız sayfanın bir de kaynak koduna bakalım. (**Görünüm** menüsünden **Kaynak** komutunu çalıştırınız.) ASP.NET sayfasını oluşturmak için yazılan ifadelerle tarayıcıya gönderilen HTML ifadeler arasındaki farkı inceleyiniz.



Resim 2.6 : sayfaYapisi.aspx sayfasının kaynak kodu

Dikkat ederseniz, ASP.NET sayfasındaki ifadelerle kaynak koddaki ifadeler birbirinden farklıdır. Çünkü ASP.NET sayfası, sunucu bilgisayarda çalıştırıldıktan sonra

çıktısı HTML ifadeleri şeklinde istemci bilgisayara gönderilmiştir. İstemcideki tarayıcı bu ifadeleri yorumlayarak ASP.NET sayfasını görüntülemiştir. Görünüm menüsünden Kaynak komutu verdiğinizde işte bu HTML ifadeleri görmekteyiz.

2.3. ASP.NET Sayfa Elemanları

Bir ASP.NET sayfası 5 sayfa elemanından oluşur.

- 1.Sayfa bildirimleri bölümü
- 2.Kod bildirim bloğu
- 3.HTML sayfa bildirimleri ve HTML etiketleri bölümü
- 4.Kod sunucu bloğu
- 5.Web formu ve web kontrolleri bölümü

<pre><%@ page language="vb" %></pre>	SAYFA BİLDİRİMLERİ BÖLÜMÜ
<pre><script runat="server"> sub tiklandi(sender as object, e as eventargs) mesaj.text = "Merhaba " & metin_kutusu.text & " ASP.NET sayfa yapısını öğrenmeye hazır mısınız?" end sub </script></pre>	KOD BİLDİRİM BLOĞU
<pre><html><body> <h1 align="left"> ASP.NET Sayfa Yapısı</h1><p></pre>	HTML SAYFA BİLDİRİMLERİ ve HTML ETİKETLERİ
<pre><% Response.write("Kutuya adınızı yazıp, Tıkla ve gör düğmesine tıklayınız:<p>") %></pre>	KOD ÇALIŞTIRMA BLOĞU
<pre><form runat="server"> <asp:textbox id="metin_kutusu" runat="server"/> <asp:button id="buton" text="Tıkla ve gör" onclick="tiklandi" runat="server"/> <p /> <asp:label id="mesaj" runat="server" /> </form></pre>	WEB FORMU VE WEB KONTROLLERİ
<pre></body></html></pre>	

Tablo 2.1 : ASP.NET sayfa elemanları

2.3.1. Sayfa Bildirimleri Bölümü

Sayfanın başında genelde sayfa bildirimleri (yönergeleri, direktifleri) denilen bir bildirim bölümü bulunur. Bildirimlerin başında @ karakteri kullanılır. Bu bölümde sayfa hakkında birtakım bilgiler verilir.

`<%@ page language="vb" %>` ifadesiyle sayfada kullanılacak programlama dilinin VB (Visual Basic) olacağı belirtilmiştir.

Sayfada kullanılan dil kodlamasını belirtmek için **codepage** ifadesi kullanılır. Eğer Windows-1254 kodlaması kullanılmak isteniyorsa;

`<%@ page language="vb" codepage="1254" %>` bildirimi,

ISO-8859-9 kodlaması kullanılmak isteniyorsa;

`<%@ page language="vb" codepage="28599" %>` bildirimi kullanılır.

Bunun gibi birçok bildirim, sayfanın en başında `<%@ %>` etiketleri arasında yapılır. Örneğin `<%@import.....%>`, `<%@register.....%>` gibi.

Uygulama : Sayfada kullanılacak programlama dilinin C# (Si şarp) olduğunu belirten bildirim yazınız.

2.3.2. Kod bildirim bloğu

`<script runat="server"></script>` ifadeleri arasında yazılan bölüme kod bildirim bloğu (Code declaration block) denir. Kod bildirim bloğunda sayfanın program kodları bulunur. Bu kısımda kullanacağımız programlama dili, sayfa bildirim alanında bildirilir. `<script runat="server">` ifadesindeki `runat="server"` kısmı bu bloktaki komutların sunucuda çalışacağını belirtir. `Runat="server"` ifadesini “sunucuda çalış” olarak çevirebiliriz. Bu bölüm bir olayın gerçekleşmesi (örneğin, sayfanın yüklenmesi, gönder düğmesine basılması) durumunda çalışacak kodları bulundurur. Örnek sayfada düğmeye basılması durumunda çalışması istenen kodlar buraya yazılmıştır.

```
<script runat="server">
    sub tiklandi(sender as object, e as eventargs)
        mesaj.text = "Merhaba " & metin_kutusu.text & " ASP.NET sayfa
        yapısını öğrenmeye hazır mısın?"
    end sub
</script>
```

sub tiklandi(sender as object, e as eventargs)

.....

end sub şeklindeki yapı VB.NET programlama diline özgü bir alt programdır. **Sub** ifadesi alt programı belirtmek için kullanılır. Alt program, tüm programın bir parçasını oluşturan, belirli bir işlemi gerçekleştirmek için oluşturulmuş kodlardır. (Sender as object, e

as eventargs) ifadesi sık kullanılan standart bir ifadedir. Alt programlar, Etkileşimli Web Uygulamaları 2 modülünde inlenecektir.

Yukarıdaki alt program içerisine örnek sayfada metin kutusuna ifade girilip düğmeye tıkladığında yapılacak işlemi belirten kodlar yazılmıştır. Metin kutusuna yazılan ifadenin önüne “Merhaba” kelimesi, sonuna “ASP.NET sayfa yapısını öğrenmeye hazır mısınız?” kelimeleri eklenerek etiketin (label) içeriği oluşturulmuş ve bu içerik sayfada görüntülenmiştir. Noktasal gösterim kullanılarak (mesaj.text) **mesaj** isimli **label** kontrolünün **text** özelliğine (property) değer atanarak işlem gerçekleştirilmiştir. Eğer label kontrolünün genişliği ayarlanmak istenseydi **mesaj.width** (örneğin mesaj.width=”5”) şeklinde yine noktasal gösterim kullanılacaktı. Eğer label kontrolünün yüksekliği ayarlanmak istenseydi nasıl bir ifade yazılırdı?

NOT: Kod bildirim bloğunun sayfanın başında bulunma zorunluluğu yoktur. İstenirse sayfanın sonuna (<html>...</html> bloğundan sonra) yazılabilir.

Uygulama : Yukarıdaki örnek ASP.NET sayfasında “script” ifadeleri arasındaki kodları sayfasının en altına alarak ASP.NET sayfasını çalıştırınız, sayfa görüntüsünde bir değişiklik oldu mu?Belirtiniz.

2.3.3. HTML Sayfa Bildirimleri ve HTML Etiketleri

HTML sayfa bildirimlerinin ve HTML etiketlerinin kullanıldığı bölümdür. Örnek sayfada <h1>, <p> etiketleriyle **ASP.NET Sayfa Yapısı** yazısı biçimlendirilmiştir. Aslında HTML etiketlerinin kullanımı sadece bu bölümle sınırlı değildir, etiketler ASP.NET sayfasının diğer bölümlerinde de kullanılabilir.

2.3.4. Kod Çalıştırma Bloğu

ASP.NET sayfasına eklenebilecek mesaj ve yönergelerin bildiriminde kullanılır. <% %> etiketleri arasında kalan bölümdür. Bu etiketler ASP bildirimidir. Bu alandaki ifadeler derlenmeyip yorumlandıklarından performansları düşüktür. Bu nedenle bu bloğun kullanılması tercih edilmez.

Burada kullanılan **Response.write("Kutuya adınızı yazıp, Tıkla ve gör düğmesine tıklayınız:<p>")** satırıyla ekrana bir metin yazdırılmaktadır. **Response**, sunucu bilgisayarın istemci bilgisayara veri aktarmasını sağlayan ASP.NET nesnesidir. Bu nesnenin **Write** metodu kullanılarak istemci tarayıcısında istenilen yazı görüntülenir.

Response.write("MEGEP") şeklinde bir komut, sunucunun istemci tarayıcısında MEGEP ifadesini görüntülemesini sağlar. Write metodunda HTML etiketleri de kullanılabilir. Örneğin, ekrana istenilen mesajı yazdıktan sonra alt paragrafa geçmek için şu komut kullanılır : **Response.write("MEGEP <p>")**

2.3.5. Web Formu ve Web Kontrolleri

Sayfada web formunun oluşturulduğu bloktur. Web formu, ASP.NET ile birlikte gelen yazım biçimi ve işleyişi farklı yeni bir form yapısıdır. Web formu, <form runat="server"> ifadesiyle başlar </form> ifadesiyle biter. Kullanıcının çeşitli veriler girmesini sağlayan formun oluşturulduğu bu alanda HTML etiketleri de kullanılabilir. Web formu, istemci bilgisayarda değil tamamen sunucu bilgisayarda çalışır ve çalıştırma sonucu üretilen HTML etiketleri istemci bilgisayara gönderilir. Runat="server" ifadesi web form ve kontrollerinin sunucuda çalıştığını göstermektedir. Bu ifade yazılmadığında sayfa hata verir. Form elemanlarının sunucuda çalıştırılması, form kullanımının kontrolünün tamamen web sitesi geliştiricisinde olmasını sağlar.

Uygulama : Örnek sayfa olan sayfaYapisi.aspx sayfasındaki <form runat="server"> satırındaki runat="server" ifadesini silip, kaydederek tarayıcıdan çağırınız. Sayfa hata verdi mi?

Uygulama : Aynı sayfada <asp:label id="mesaj" runat="server"/> satırındaki runat="server" ifadesini silip, kaydederek tarayıcıdan çağırınız. Bu sefer sonuç ne oldu?

Formdaki veriler, istemci tarafında çalışan betik kodlar vasıtasıyla işlenebileceği gibi sunucu tarafında çalışan kodlar vasıtasıyla da işlenebilir. Formlardaki değişiklikler olaya bağlanmıştır. Kullanıcı yaptığı değişiklikle bir olayı gerçekleştirdiğinde, bu değişiklik sunucuya iletilir ve sunucudaki program komutları çalıştırılır. Eğer olay gerçekleşmezse, sunucu sayfayı tekrar çalıştırmaz.

Web formları ve web form elemanları "Etkileşimli web uygulamaları 3" modülünde incelenecektir. Şimdilik örnek sayfadaki web form ve elemanlarıyla ilgili kısa bilgiler verilecektir. Örnek sayfada kullanılan form ve form elemanlarını oluşturma ifadeleri aşağıda belirtilmiştir.

```
<form runat="server">
    <asp:textbox id="metin_kutusu" runat="server"/>
    <asp:button id="buton" text="Tıkla ve gör" onclick="tiklandi"
        runat="server" />
    <asp:label id="mesaj" runat="server" />
</form runat="server">
```

Form oluşturmak amacıyla **form** ifadesi kullanılmış ve bu formun sunucuda çalışacağını belirtmek amacıyla da **runat="server"** ifadesi eklenmiştir. Oluşturulacak form elemanları (kontrolleri) web form elemanları olduklarından satıra **<asp:** ifadesiyle başlanmıştır. Bu ifadenin ardından forma hangi web form elemanı eklenecekse onun İngilizce ismi (textbox, button, label vs.) yazılmıştır. Daha sonra form elemanına özel isim **id** bildirimiyile atanmıştır. Textbox form elemanı için id değerine **metin_kutusu**, button (düğme) form elemanı için ise **buton** ifadesi atanmıştır.

Buton kontrolünün üzerindeki yer alacak yazıyı belirlemek için **text** özelliği kullanılmıştır. Ardından, **onclick="tiklandi"** ifadesiyle düğme tıklandığında (click), **tiklandi** alt programının çalıştırılacağı belirtilmektedir.

NOT : Click (Klik) kelimesi Türkçe’de **tıklamak** anlamına gelir.

NOT : `<asp:button id="buton" text="Tıkla ve gör" onclick="tiklandi" runat="server"/>` satırı `<asp:button id="buton" text="Tıkla ve gör" onclick="tiklandi" runat="server"></asp:button>` satırının daha kısa yazımıdır.

2.4. Kodu Alt Satırdan Devam Ettirme

ASP.NET sayfaları oluştururken bazı satırlar uzun olabilir. Böyle bir durumda alt satırdan devam etme ihtiyacı hissedilir. Fakat alt satıra geçerken kimi yazım kurallarına uyulmalıdır. Kod yazarken doğrudan alt satıra geçip devam etmek hataya neden olur. Ayrıca kimi zaman fazladan bırakılan bir boşluk hataya neden olabilir.

HTML etiketleriyle bir işlemi yerine getirirken ifadenin her zaman aynı satırda olması şart değildir. İfadenin alt satırdan devam etmesinde esneklik söz konusudur. Örneğin ;

```
<u> Sevgi </u>
```

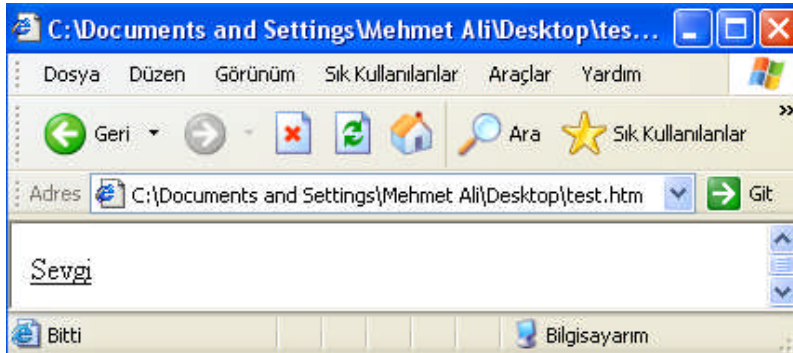
şeklindeki bir yazım ile

```
<u>  
Sevgi </u>
```

veya

```
<u>  
Sevgi  
</u>
```

şeklindeki bir yazım tarayıcı için aynı anlama gelir. Her üç yazımın ekran görüntüsü de aşağıdaki gösterildiği gibi altı çizili halde yazılmış “Sevgi” kelimesi şeklinde olur.

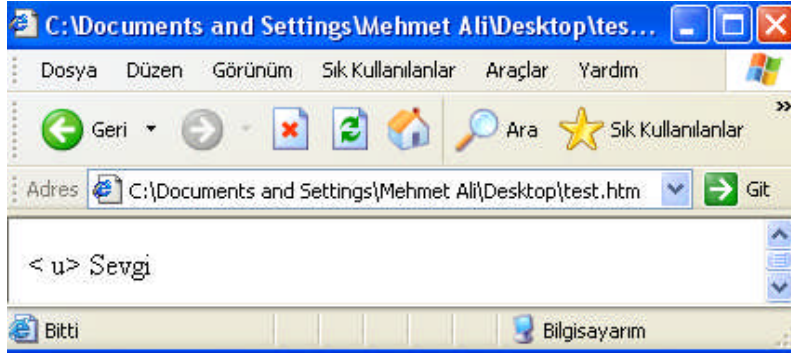


Resim 2.7 : test.htm sayfasının tarayıcıdaki görüntüsü 1

Fakat HTML etiketleri bölündüğünde, tarayıcı hata vermese de ekranda istenilenden farklı bir görüntüyle karşılaşma ihtimali vardır. Örneğin;

```
<
u>
Sevgi
</u>
```

şeklindeki yazım, tarayıcının hata vermesine neden olmamış, ancak aşağıda görüldüğü gibi altı çizili halde yazılmış “sevgi” kelimesi görüntüsünü de vermemiştir.



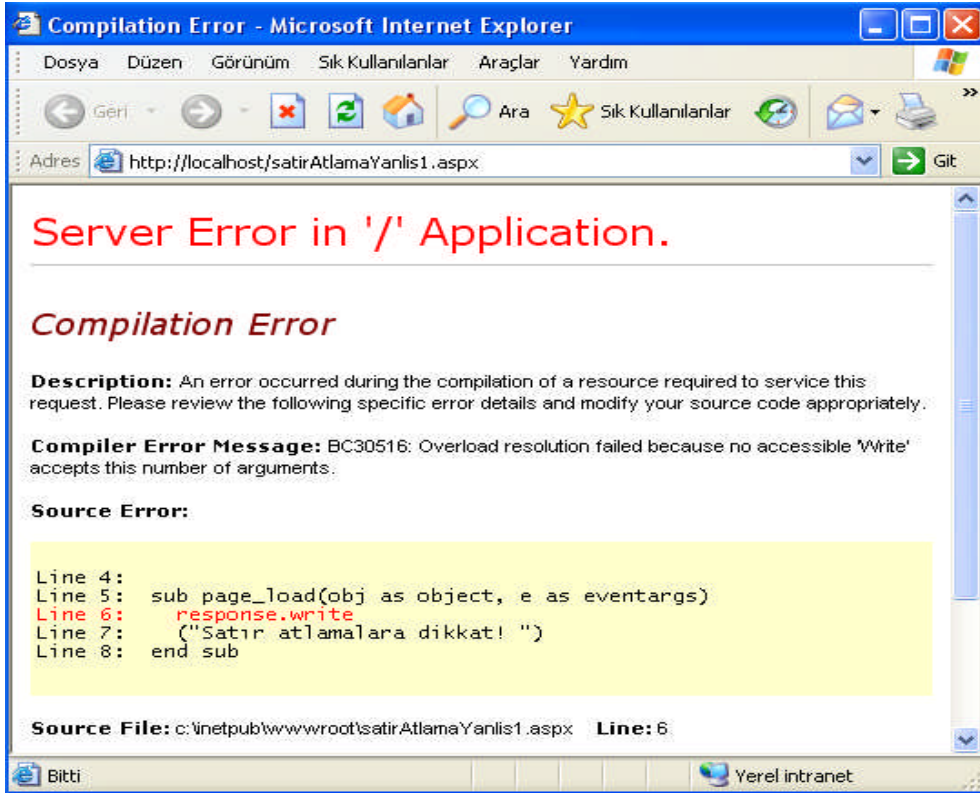
Resim 2.8 : test.htm sayfasının tarayıcıdaki görüntüsü 2

HTML kodlarının istemci bilgisayarın tarayıcısında yorumlanmasından dolayı, HTML ifadelerinin yazım kurallarında esneklik payı vardır. Ancak sunucu bilgisayarda yorumlanan veya derlenen kodlarda yazım kurallarına daha fazla dikkat edilmelidir. Örneğin;

response.write(“Satır atlamalara dikkat! ”) kodunun parçalanarak yazıldığı aşağıdaki örnekteki yazım şekli hataya neden olacaktır.

satirAtlamaYanlis1..aspx

```
1 <%@ page language="VB" %>
2 <script runat="server">
3 sub page_load(obj as object, e as eventargs)
4     response.write      (DİKKAT! DOĞRUDAN ALT SATIRA GEÇİLMİŞ)
5     ("Satır atlamalara dikkat! ")
6 end sub
7 </script>
8 <html><body></body></html>
```



Resim 2.9 : satirAtlamaYanlis1.aspx sayfasının tarayıcıdaki görüntüsü

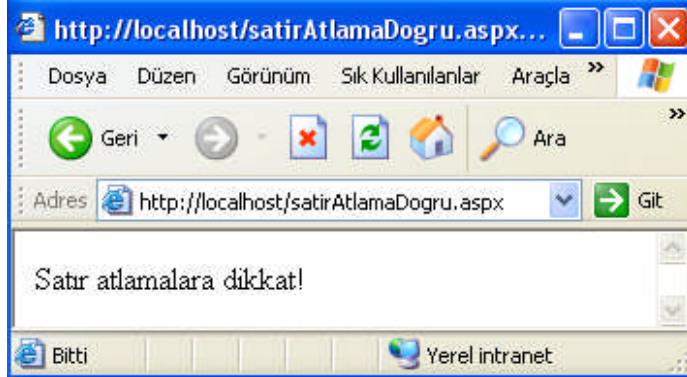
Eğer herhangi bir nedenle alt satıra geçilecekse, bulunulan satırda, ara (boşluk) çubuğuyla bir boşluk bırakıldıktan sonra alt çizgi (_) karakterine basılır ve alt satıra geçilir. Örneğin;

```
response.write _  
("Satır atlamalara dikkat! ")
```

şeklindeki bir yazım doğru olup, hataya neden olmaz. Aşağıda görülen satirAtlamaDogru.aspx sayfasını ve çıktısını inceleyiniz.

satirAtlamaDogru.aspx

```
1 <%@ page language="VB" %>  
2 <script runat="server">  
3     sub page_load(obj as object, e as eventargs)  
4         response.write _  
5             ("Satır atlamalara dikkat! ")  
6     end sub  
7 </script>  
8 <html>body</body></html>
```

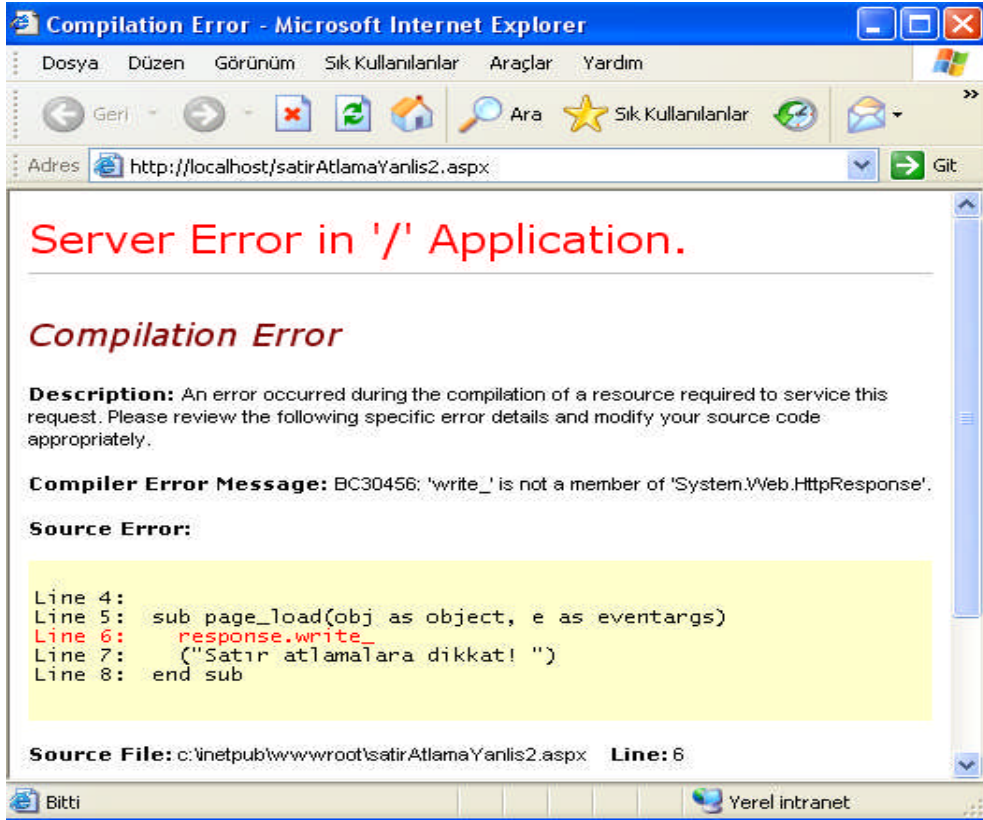


Resim 2.10 : satirAtlamaDogru.aspx sayfasının tarayıcıdaki görüntüsü

Alt satıra geçme işleminde alt çizgiden önce boşluk bırakılmamasının hataya neden olduğu aşağıdaki örneği inceleyelim.

SatirAtlamaYanlis2.aspx

```
1 <%@ page language="VB" %>
2 <script runat="server">
3 sub page_load(obj as object, e as eventargs)
4     response.write_ (DİKKAT: ALT ÇİZGİDEN ÖNCE BOŞLUK BIRAKILMAMIŞ)
5     ("Satır atlamalara dikkat! ")
6 end sub
7 </script>
8 <html><body></body></html>
```



Resim 2.11 : satirAtlamaYanlis2.aspx sayfasının tarayıcıdaki görüntüsü

VB.NET'te kullanılan her komut tek satıra yazılır. Bununla birlikte, alt satırdan devam etme gereği duyulduğunda, VB.NET'te de bulunan satır "boşluk + alt çizgi" karakterleri ile bitirilir ve alt satıra geçilir.

Daha önce oluşturduğunuz sayfaYapisi.aspx dosyasını açarak, aşağıda kırmızı renkle yazılan alanda olduğu gibi alt satıra geçme işlemi yapınız. Dosyayı kaydedip, çalıştırdığımızda sayfa hatasız görüntülenecektir.

```
<%@ page language="vb" %>
<script runat="server">
sub tiklandi(sender as object, e as eventargs)
mesaj.text = "Merhaba " & yazi_kutusu.text & _
" ASP.NET sayfa yapısını öğrenmeye hazır mısın?"
end sub
.....
```

Alt satıra geçme işlemi, bir komutu oluşturan ifadelerin bağlantı noktalarında yapılsa hataya neden olmaz. Aşağıdaki örneklerde gösterilen alt satıra geçme işlemleri uygulandığında dosya çalışma hatası vermez.

```
mesaj. _  
text = "Merhaba " & yazi_kutusu.text & + " ASP.NET sayfa yapısını öğrenmeye hazır mısın?"
```

```
mesaj.text _  
= "Merhaba " & yazi_kutusu.text & + " ASP.NET sayfa yapısını öğrenmeye hazır mısın?"
```

```
mesaj.text = _  
"Merhaba " & yazi_kutusu.text & + " ASP.NET sayfa yapısını öğrenmeye hazır mısın?"
```

```
mesaj.text = "Merhaba " _  
& yazi_kutusu.text & + " ASP.NET sayfa yapısını öğrenmeye hazır mısın?"
```

Kod içindeki bir metni parçalayarak alt satırdan devam edersek metnin herhangi bir yeri bir bağlantı noktası olmadığından hatayla karşılaşırız. Aşağıda gösterildiği gibi metni böldüğümüzde sayfa hata verecektir.

```
mesaj.text = "Merhaba " & yazi_kutusu.text & + " ASP.NET sayfa _  
yapısını öğrenmeye hazır mısın?"
```

Eğer metni bağlantı noktası da oluşturarak bölersek herhangi bir hatayla karşılaşmayız.

```
mesaj.text = "Merhaba " & yazi_kutusu.text & + " ASP.NET sayfa" & _  
"yapısını öğrenmeye hazır mısın?"
```

2.5. Program Kodlarına Açıklamalar Ekleme

ASP.NET uygulamaları geliştirirken, yaptığınız işlemleri açıklamak amacıyla notlar koyabilirsiniz. Bu notlar sayesinde farklı zamanlarda kodlarınıza baktığınızda yaptığınız işlemleri rahatlıkla hatırlayabilirsiniz. Böylelikle zamandan ve emekten kazancınız olur. Bu açıklayıcı notların diğer bir yararı, yazdığınız uygulamaları başkalarıyla da paylaştığınızda onların da bu notlardan yararlanmasıdır.

VB.NET'te açıklayıcı notlar ‘ **(tek tırnak)** işareti ile başlar.

Örnek açıklama satırı:

‘ Kullanıcıdan kullanıcı adı ve parolasının alındığı bölüm

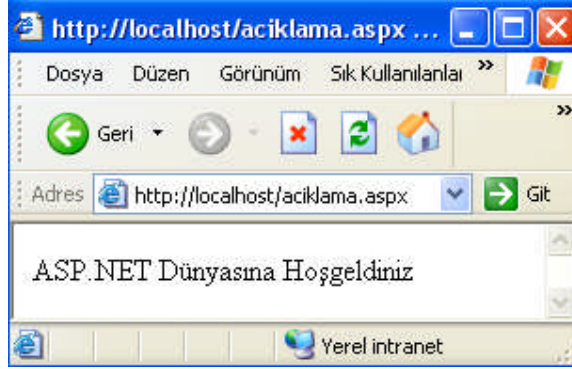
Başka bir örnek açıklama satırı

‘ Sayfadaki görsel düzenlemelerin yapıldığı bölüm

Bu açıklayıcı notlar adından da anlaşılacağı gibi sadece işlemleri açıklamak içindir. Uygulamanın çalışmasını etkilemediğini görmek için aşağıdaki uygulamayı inceleyelim.

aciklama.aspx

```
1 <%@ page language="VB" %>
2 <script runat="server">
3     sub page_load(obj as object, e as eventargs)
4         ' Mesajın yazıldığı kısım
5         metin.text ="ASP.NET Dünyasına Hoşgeldiniz"
6     end sub
7 </script>
8 <html><body>
9 <asp:label id="metin" runat="server" />
10 </body></html>
```

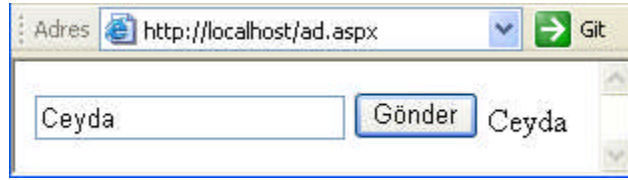


Resim 2.12 : aciklama.aspx sayfasının tarayıcıdaki görüntüsü

Oluşturduğumuz aciklama.aspx dosyası içindeki açıklama satırının, ekran çıktısına herhangi bir etkisi olmamıştır.

UYGULAMA FAALİYETİ

Metin kutusuna yazılan metni düğmeye tıklandığında etiket kontrolünü kullanarak tarayıcıda görüntüleyen ASP.NET sayfasını oluşturunuz. Sayfayı hazırlarken size kolay gelen farklı işlem basamaklarını kullanabilirsiniz. Sayfayı **ad.aspx** adıyla kaydediniz. Sayfanızın tarayıcıdaki görüntüsü aşağıdaki resme benzer olacaktır.



İşlem Basamakları	Öneriler
Sayfada kullanılacak programlama dilini ifade eden bildirim yazınız.	
Web formunu oluşturunuz.	Formun sunucu kontrolü olduğunu belirten runat="server" ifadesini eklemeyi unutmayınız. Etiket kapatmayı unutmayınız.
Metin kutusu, düğme ve etiket kontrollerini tanımlayınız.	Kontrollerin id ve text parametrelerine kısa, öz, içeriğe uygun ifadeler kullanmaya dikkat ediniz. Butona tıklandığında (click) hangi alt programın çalışacağı belirtiniz.
Alt programı oluşturunuz.	Program kodları <script> blokları arasında yazılmalıdır. Alt programı sonlandırmayı unutmayınız.
Metin kutusundaki metni etiket kontrolüne atayan komutu yazınız.	Label1.Text şeklinde noktasal gösterim kullanılmalıdır.

ÖLÇME VE DEĞERLENDİRME

A. OBJEKTİF TESTLER (ÖLÇME SORULARI)

Doğru/Yanlış Soruları

Aşağıdaki ifadeleri okuyunuz ve doğru olanlar ifadelerin önündeki boşluğa “D”, yanlış olan ifadelerin sonundaki boşluğa “Y” harfi koyunuz.

- 1) Çalıştırılan bir ASP.NET sayfasının çıktıları HTML olarak tarayıcıya gönderilir. ()
- 2) ASP.NET'in tanımlı karakter kodlama tekniği Unicode (UTF-8)'dir. ()
- 3) ASP.NET sayfasına `<script>` `</script>` ifadeleri arasında kalan bölüm program kodlarının yazıldığı bölümdür. ()
- 4) ASP.NET sayfaları sadece VB (Visual Basic) programlama diliyle oluşturulurlar. ()
- 5) ASP.NET dosyalarını üzerinde çift tıklatarak çalıştırabilirsiniz. ()
- 6) ASP.NET uygulamasının görsel yüzünü oluşturan ifadeler ile program kodları farklı dosyalarda tutulabilir. ()
- 7) Tarayıcıda görüntülenen bir ASP.NET sayfasının kaynak kodundaki ifadelere bakarak içeriğini inceleyebiliriz. ()
- 8) ASP.NET sayfalarından HTML etiketleri kullanılamaz. ()
- 9) Sayfada kullanılacak dil kodlamasını belirtmek için Page bildirimden `codePage` seçeneğini kullanırız. ()
- 10) Response nesnesi, istemciden sunucuya veri aktarmak için kullanılır. ()
- 11) `<form runat="server">` satırındaki `runat="server"` ifadesi formun istemci bilgisayarda çalışacağını belirtir. ()

MODÜL DEĞERLENDİRME

MODÜL YETERLİK ÖLÇME (PERFORMANS TESTİ)

Aşağıdaki ifadeleri okuyunuz ve doğru olanlar ifadelerin önündeki boşluğa “D”, yanlış olan ifadelerin sonundaki boşluğa “Y” harfi koyunuz.

- 1) Günümüze kadar Türkçe karakterlerin programlarda gösterilmesinde bir problem yaşanmamıştır.()
- 2) Harfler, rakamlar, noktalama işaretleri gibi tüm semboller bilgisayar dünyasında karakter olarak isimlendirilir. ()
- 3) Bir internet sayfasını oluşturan kodlar/komutlar istemci bilgisayarda çalıştırılıyorsa o sayfaya sunucu taraflı sayfa adı verilir. ()
- 4) IIS, bir web sunucusudur. ()
- 5) IIS programının varsayılan yerel yolu “c:\inetpub\wwwroot” klasörüdür. ()
- 6) Belirlediğiniz herhangi bir klasörü web sunucu klasörü yapabilirsiniz. ()
- 7) ASP.NET sayfalarının uzantısı “asp”dir. ()
- 8) İnternet uygulamaları istemci-sunucu modeline göre çalışırlar. ()
- 9) ASP.NET sayfaları geliştirmek için sadece C# programlama dili kullanılır. ()
- 10) İstemci taraflı kodlama, etkileşimli web uygulamaları geliştirmeyi olanaklı hale getirmiştir. ()
- 11) Sınıflar, ad uzaylarından oluşur. ()

PERFORMANS TESTİ (YETERLİK ÖLÇME)

Modül ile kazandığınız yeterliği aşağıdaki ölçütlere göre değerlendiriniz.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
Web sunucusunu kurdunuz mu?		
.Net Framework'ü kurdunuz mu?		
Basit bir ASP.NET sayfasını oluşturduğunuz mu?		
Sanal izin oluşturduğunuz mu?		
Sayfanızı Türkçe karakterleri gösterebilecek şekilde ayarladınız mı?		
Sayfanıza açıklamalar eklediniz mi?		

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'İN CEVAP ANAHTARI

Sorular	Cevaplar
1	D
2	Y
3	D
4	D
5	Y
6	D
7	D
8	Y
9	Y
10	Y

ÖĞRENME FAALİYETİ 2'İN CEVAP ANAHTARI

Sorular	Cevaplar
1	D
2	D
3	D
4	Y
5	Y
6	D
7	Y
8	Y
9	D
10	Y
11	Y

MODÜL DEĞERLENDİRME CEVAP ANAHTARI

Sorular	Cevaplar
1	Y
2	D
3	Y
4	D
5	D
6	D
7	Y
8	D
9	Y
10	Y
11	Y

SÖZLÜK

Aşağıdaki İngilizce terimlerin okunuşları mümkün olduğunda kelimelerle ifade edilmeye çalışılmıştır. Okunuşları en doğru telaffuz etmek için kelimelerin okunuşlarını da veren bir İngilizce sözlük programı kullanabilirsiniz. Örneğin, redhouse veya internette www.seslisozluk.com adresi.

Terim	Okunuşu	Anlamı, Açıklaması
Active	Aktiv	Aktif, Etkin
Alias	Eliyıs	Takma ad, başka ad, diğer ad
Box	Baks	Kutu
C#	Si şarp	Bir programlama dili
Charset	Çarset	Karakter seti, karakter kümesi
Click	Klik	Tıklamak
Client	Kılayınt	İstemci
Code	Kod	Kod, program komutu
Default	Difolt	Varsayılan, ön tanımlı
Directory	Direkdırı	Dizin, klasör
Dot	Dot	Nokta
Encoding	İnkodin	Dil kodlaması
Event	İvent	Olay
Host	Houst	İnternete yayınlanacak web sitesini barındırmak
Information	İnfirmeşın	Enformasyon
Internet	İntirnet	Genel ağ, internet
Language	Lengiviç	Dil, lisan
Local	Lokıl	Yerel
Namespace	Neymsipeys	Ad uzayı
Page	Peyc	Sayfa
Perl	Pörl	Bir programlama dili
Professional	Prifeşinıl	Profesyonel
Request	Rikvest	İstek, talep
Response	Rispans	Cevap
Root	Rut	Kök
Run	Ran	Çalıştırmak
Script	Sıkript	Betik Dil
Server	Sörvır	Sunucu bilgisayar, sunucu
Source	Sors	Kaynak
Shareware	Seyırweyır	Paylaşımlı
Text	Tekst	Metin
Unicode	Yunikod	Evrensel Kod
Virtual	Vörçuvıl	Sanal
Visual Basic	Vijyul Beyzik	Bir programlama dili
Web	Veb	örün

ÖNERİLEN KAYNAKLAR

- DEMİRKOL Zafer, ASP.NET, Pusula Yayıncılık, İstanbul, 2005
- DUTHIE G. Andrew, Adım Adım Microsoft ASP.NET, Arkadaş Yayınevi, Ankara, 2005
- SANKUR Bülent, Bilişim Sözlüğü 2005 programı, Yazılım: Hakan GÜLERYÜZ, Pusula Yayıncılık
- PALA Zeydin, ASP.NET İle Adım Adım Web Uygulamaları, Türkmen Kitabevi, İstanbul, 2006
- office.microsoft.com/tr-tr/help/HA011331361055.aspx
- www.macromediaturk.com
- www.fazlamesai.net/index.php?a=article&sid=1481#6626
- www.startvbdotnet.com/aspsite/asp/
- www.startvbdotnet.com/dotnet/framework.aspx
- www.startvbdotnet.com/dotnet/languages.aspx

KAYNAKÇA

- DEMİRKOL Zafer, **ASP.NET**, Pusula Yayıncılık, İstanbul, 2005
- DUTHIE G. Andrew, **Adım Adım Microsoft ASP.NET**, Arkadaş Yayınevi, Ankara, 2005
- SANKUR Bülent, **Bilişim Sözlüğü 2005 programı**, Yazılım: Hakan GÜLERYÜZ, Pusula Yayıncılık
- PALA Zeydin, **ASP.NET ile Adım Adım Web Uygulamaları**, Türkmen Kitabevi, İstanbul, 2006
- office.microsoft.com/tr-tr/help/HA011331361055.aspx
- www.macromediaturk.com
- www.fazlamesai.net/index.php?a=article&sid=1481#6626
- www.startvbdotnet.com/aspsite/asp/
- www.startvbdotnet.com/dotnet/framework.aspx
- www.startvbdotnet.com/dotnet/languages.aspx