

AJAX UYGULAMALARININ SUNUCU PERFORMANSINA ETKİSİ

M. Salih KAYA

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR EĞİTİMİ ANABİLİM DALI**

**GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ**

**HAZİRAN 2008
ANKARA**

Mehmet Salih KAYA tarafından hazırlanan AJAX UYGULAMALARININ
SUNUCU PERFORMANSINA ETKİSİ adlı bu tezin Yüksek Lisans tezi olarak
uygun olduğunu onaylım.


Yrd. Doç. Dr. Halil İbrahim BÜLBÜL
Tez Yöneticisi

Bu çalışma, jürimiz tarafından Bilgisayar Eğitimi Anabilim Dalında Yüksek
Lisans tezi olarak kabul edilmiştir.

Başkan : Prof. Dr. Doğan ÇALIKOĞLU



Üye : Doç. Dr. O. Ayhan ERDEM



Üye : Yrd. Doç. Dr. Halil İbrahim BÜLBÜL



Bu tez, Gazi Üniversitesi Bilişim Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.



M. Salih KAYA

AJAX UYGULAMALARININ SUNUCU PERFORMANSINA ETKİSİ**(Yüksek Lisans Tezi)****Mehmet Salih KAYA****GAZİ ÜNİVERSİTESİ****BİLİŞİM ENSTİTÜSÜ****Haziran 2008****ÖZET**

Bu çalışmada internet kullanıcılarının artan taleplerine karşılık vermek için geliştirilmeye devam eden, web 2.0 döneminin önemli bir basamağı Ajax tekniği üzerinde durulmuştur. Bu tekniğin neden kullanılması gerektiği ve kullanım yollarından bahsedilmiştir. Ajax tekniğinin İnternet kullanıcılarına sunduğu görselliğin yanında, sunucu performansına etkisi ve sağladığı kazanımlar da yapılan testlerle ortaya konulmuştur. Klasik ve Ajax tekniği ile hazırlanmış benzer özellikte iki ayrı web uygulaması; performans test yazılımlar araçlarıyla test edilmiş, kullanılan bant genişliği, işlem süresi açılarından sonuçlar karşılaştırılmıştır. Testler sonucunda, Ajax tekniği ile hazırlanmış web sayfalarının, veri trafiğini yaklaşık %51 oranında düşürerek, sunucu performansını artırdığı görülmüştür.

Bilim Kodu : 702.3.006**Anahtar Kelimeler : Ajax, asenkron çalışma, senkron çalışma,
Xmlhttprequest nesnesi****Sayfa Adedi : 73****Tez yöneticisi : Yrd. Doç. Dr. Halil İbrahim BÜLBÜL**

THE EFFECT OF AJAX APPLICATIONS ON SERVER PERFORMANCE
(M.Sc. Thesis)

Mehmet Salih KAYA

GAZİ UNIVERSITY
INFORMATICS INSTITUTE

June 2008

ABSTRACT

This research emphasized Ajax technique that is part of Web 2.0 has been continue to developed for responding to increasing number of requests of Internet users. Moreover, this research state that why Ajax must be used and ways of using Ajax technique. The done tests say that Ajax technique increase performance of server and provides so many advantages other than supplying visual quality to internet users. The similar two web applications of coded by using classical way and Ajax technique has been tested by performance test tools. Then results of using bandwidth and time of process of applications are compared. As a result of these tests, it can be seen that the web application coded by Ajax technique decrease web traffic as percent of 51 by the way increase performance of server.

Science Code : 702.3.006

**Keywords : Ajax, asynchronous work, synchronous work,
XmlHttpRequest object**

Page Number : 73

Adviser : Asist. Prof. Dr. Halil İbrahim BÜLBÜL

TEŐEKKÜR

Çalıőmam süresince sonsuz sabır ve hoőgörüsünü esirgemeyen, yardım ve katkılarıyla beni yönlendiren deęerli danıőman hocam Sayın Yrd. Doç. Dr. Halil İbrahim BÜLBÜL'e, yine tavsiyeleriyle çalıőmama yön veren Sayın Doç. Dr. O. Ayhan ERDEM'e, aileme ve performans testlerine katılarak çalıőmanın tamamlanmasına katkıda bulunan Ted Ankara Koleji Vakfı Özel Lisesi öęrencilerine teőekkür ederim.

Sayfa

ÖZET.....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER.....	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ.....	x
RESİMLERİN LİSTESİ.....	xi
SİMGELER VE KISALTMALAR.....	xii
1. GİRİŞ.....	1
2. KONU İLE İLGİLİ YAPILAN ÇALIŞMALAR.....	4
3. MATERYAL VE METOT.....	6
4. YENİ WEB UYGULAMALARI.....	8
4.1. Web 2.0.....	8
4.2. Ajax Tekniği.....	13
4.2.1. Ajax'ın arkasındaki teknolojiler.....	17
4.2.2. Ajax mimarisi.....	22
4.2.3. Ajax tekniğinin avantajları ve dezavantajları.....	28
4.2.4. Ajax'ın en iyi kullanım alanları.....	31
4.2.5. Ajax kullanan internet siteleri.....	33
4.3. XmlHttpRequest Nesnesi.....	35
4.3.1. Uygulamalar.....	38
4.4. Ajax Kütüphaneleri.....	48

	Sayfa
5. PERFORMANS TESTLERİ VE DEĞERLENDİRMESİ.....	50
5.1. Web Sitesi Performans Testi	50
5.2. E-Posta Servislerinin Performans Testleri.....	53
5.3. Veritabanı Uygulamalarında Ajax Performans Testi.....	56
6. SONUÇ VE ÖNERİLER	62
KAYNAKLAR	65
EKLER.....	67
ÖZGEÇMİŞ	73

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 4.1. Web 1.0 uygulamalarının web 2.0 dönemindeki gelişimi.....	11
Çizelge 4.2. Klasik teknik ile Ajax tekniğinin karşılaştırılması	26
Çizelge 5.1. İlk test sonuçları	52
Çizelge 5.2. Test sonuçları ortalaması	53
Çizelge 5.3. Live mail ilk test sonucu	55
Çizelge 5.4. Live mail test sonucu	55
Çizelge 5.5. Yahoo mail test sonuçları	55
Çizelge 5.6. Veritabanı uygulamasının dosya boyutu	57
Çizelge 5.7. Beş deneğe ait performans testi sonuçları	58
Çizelge 5.8. Yirmibin kayıtlı veritabanı uygulamasında performans testi sonucu	59
Çizelge 5.9. Otuzbin kayıtlı veritabanı uygulamasında performans testi sonucu	59
Çizelge 5.10. Kırkbin kayıtlı veritabanı uygulamasında performans testi sonucu	59

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 4.1. Dom tablosu	20
Şekil 4.2. XmlHttpRequest nesnesi	21
Şekil 4.3. Klasik web uygulama modeli ile Ajax web uygulama modelinin çalışma mantığı	24
Şekil 4.4. Klasik ve Ajax web uygulamalarındaki veri iletişim modelleri	25
Şekil 4.5. Eşzamanlı çalışma yöntemi.....	27
Şekil 4.6. Eşzamansız çalışma yöntemi.....	28
Şekil 4.7. Basit bir geri bildirim formu uygulaması.....	39
Şekil 4.8. Otomatik doldurma için form tasarımı.....	44
Şekil 5.1. Kayıt sayısına göre sunucudan gelen veri miktarlarının ortalamaları	60
Şekil 5.2. Test sonucunda elde edilen ortalama işlem süreleri	61

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 4.1. Eyeos.com sitesinin masaüstü uygulaması	29
Resim 4.2. Google arama motorunda Ajax tekniği	31
Resim 4.3. Form verisinin kontrolü	32
Resim 4.4. Mikrolink	32
Resim 4.5. Aktif arama	33
Resim 4.6. Google suggest uygulaması	34
Resim 5.1. Test sitesi	51
Resim 5.2. Klasik tekniğin performans test sonucu	52
Resim 5.3. Ajax tekniğinin performans test sonucu	52
Resim 5.4. Wapt performans test yazılımının ara yüzü	54
Resim 5.5. Veritabanı uygulaması ara yüzü	56
Resim 5.6. Fiddler Web Debugger ara yüzü	58

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
AJAX	Asynchronous Javascript And Xml
ASP	Active Server Page
CSS	Cascading Style Sheet
DOM	Document Object Model
HTML	HyperText Markup Language
PHP	Hypertext Pre-Processor
RSS	Rate Site Summary
XHR	XmlHttpRequest
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language

1. GİRİŞ

İnternet günümüzün başlıca bilgi, haberleşme kaynağı olmuştur ve olmaya da devam etmektedir. Internetworldstast Kasım 2007 verilerine göre dünya nüfusunun %19,1'lik kısmı yaklaşık 1 262 032 697 insan internet kullanmaktadır. 2000 yılından itibaren tutulan istatistiklere göre internet kullanım oranı %249,6 oranında artmıştır.

Bu ilginin gün geçtikçe daha da artması kuşkusuzdur. İnternet hayatımızda o kadar çok yer etmiştir ki içinde bulunduğumuz yüzyılın adı "bilgi çağı" olarak adlandırılmaktadır.

İnternet kullanımının artması beraberinde sorunları ve bitmez tükenmez insan ihtiyacını da beraberinde getirmiştir. Kullanıcılar artık web sitesi sahiplerinin kendilerine sunmuş olduğu olanaklarla yetinmek istememekte, daha fazla görsellik, daha fazla kullanışlılık ve web sitesi içeriğinde daha fazla söz hakkı almak istemektedirler.

Klasik tasarım ve tekniklerle oluşturulmuş web ortamlarında;

- Yoğun içerikli sitelerde sayfalara erişim süresi uzundur.
- Sayfa içerisindeki ufak bir bölüm üzerindeki değişiklik için tüm sayfanın tekrar yüklenmesi gerekmekte bu da otomatik olarak sayfaya erişim süresini artırmaktadır.
- Küçük bir bölüm için tüm sayfanın tekrar yüklenmesi veri trafiğini gereksiz yere artırmakta, sistemi yormaktadır.
- Bant genişliğinin gereksiz yere dolması kullanıcılarda kota sorununu ve mali yükü de beraberinde getirmektedir.
- Masaüstü uygulamalarında kullanıcıların sıkça kullanıp aşına oldukları, sürükle-bırak işlemleri, silme, kaydetme gibi dosya işlemleri yapılamamaktadır.
- Kullanıcılar web sitesinin var olan tasarımını ile yetinmek zorunda kalırlar, sayfanın içindeki düzeni değiştirme hakları yoktur.

- Kullanıcılar gerçek zamanda sadece bir işlem yapabilir, bu işlemin yanıtını almadan ikinci bir istekte bulunamazlar.

Yukarıda bahsedilen istenmeyen problemlere çözüm olarak web 2.0 olarak adlandırılan yeni bir oluşum başlatılmıştır.

Son birkaç yıl içinde web ile ilişkili aramalar yapıldığında “web 2.0” terimi ile karşılaşılması kuvvetle muhtemeldir. Kimileri bunun yeni bir teknoloji olduğunu söyleyecekse de aslında birçok şirketin yeni web stratejileri etrafında toplandıkları bir kavramdır. Başka bir ifadeyle web 2.0, world wide web’i aslından ayırma ve bir sonraki adımdaki insan ve bilgisayar etkileşimindeki gelişme düşüncesidir [1].

Web 2.0 Ekim 2004’de Tim O’Reilly ve MediaLive International tarafından organize edilen ve yeni web anlayışının tartışıldığı konferansın adıdır ve o yıldan itibaren her yıl artan katılımcı sayısı ile devam etmektedir.

Web 2.0 döneminin en önemli buluşu da Ajax(Asynchronous JavaScript and XML) tekniğidir. Web 2.0 Ajax değildir, Ajax’ta web 2.0 değildir. Bu dönem içinde kullanılmaya başlanmış uygulamalar arasında oldukça ilgi gören bir tekniktir.

Ajax tekniği kullanılan bir sayfada;

- Klasik web anlayışına göre sayfalara erişim daha hızlıdır.
- Sayfa içerisinde güncellenmesi gereken ufak bir bölüm için tüm sayfanın tekrar yenilenmesine gerek kalmaz, ilgili bölüm diğerlerinden bağımsız şekilde güncellenebilmektedir.
- Sayfanın bütünü yerine parçalarının güncellenebilmesi veri trafiğini düşürmekte, para ve zaman tasarrufu sağlamaktadır.
- Masaüstü uygulamalarında kullanılan sürükle-bırak işlemleri, dosya taşıma, silme, kaydetme gibi işlemler yapılabilmektedir.

- Verilen tepkiye sayfanın tekrar güncellenmeden karşılık vermesi sayfa tasarımını daha kullanışlı ve daha cazip hale getirmektedir.
- Kullanıcılar aynı zaman dilimi içinde sunucudan birden fazla istekte bulunabilmekte, cevapları ayrı ayrı aynı sayfa içinde görebilmektedirler.

Bu çalışma da, Ajax tekniğine hızla artan ilginin nedenleri, Ajax tekniğinin nasıl kullanıldığı, en önemli özelliği olan web sayfalarının sunucu ile eşzamansız veri iletişimi yaparak bant genişliğinden elde ettiği tasarrufun nasıl ve ne ölçüde olduğu üzerinde durulmuştur.

Bu tez altı bölümden oluşmuştur. İkinci bölümde bu alanda yapılmış çalışmalara yer verilmiştir. Üçüncü bölümde tez çalışmasında kullanılan materyal ve metotlar belirtilmiştir. Dördüncü bölümde Ajax tekniğinin tanımı, kullanım alanları, avantaj ve dezavantajları ve XmlHttpRequest(Xhr) nesnesi üzerinde durulmuştur. Beşinci bölüm Ajax tekniğinin sunucu performansı üzerindeki etkisini görebilmek için performans testlerine yer verilmiştir. Son bölüm ise sonuç ve önerileri içermektedir.

2. KONU İLE İLGİLİ YAPILAN ÇALIŞMALAR

Web sayfalarının görsel tasarımının ve kullanılabilirliğinin daha fazla önem kazandığı bu günlerde Ajax tekniği de merak edilen konuların başında gelmektedir. Bu bölümde üniversitelerin kütüphaneleri, internet siteleri, yayınlanmış makaleler ve kitaplar taranmak suretiyle elde edilen çalışmalardan bazıları özetlenmiştir.

Garrett, J., J. 18 Şubat 2005 tarihinde yayımladığı makalesinde Ajax tekniğini ilk defa tüm dünyaya duyurmuştur. Bu makalesinde Ajax tekniğinin tanımını yapmış, nasıl çalıştığını ve diğer tekniklerden farklarını ortaya koymuştur. Ajax'ın bir teknoloji olmadığını, bir çok teknolojinin beraber kullanılmasıyla oluşturulmuş bir yaklaşım olduğunu ve daha yolun çok başında olduklarından bahsetmiştir.

Paulson, L. D., Ekim 2005'te IEEE Computer Society tarafından yayımlanan Computer dergisinde yer alan makalesinde Ajax tekniğinin kullandığı teknolojileri ve nasıl çalıştığının yanında, büyük web şirketlerin bu tekniğe önem verdiklerini belirtmiştir. Macromedia Flash ile kıyaslandığını söyleyen Paulson, Ajax tekniğinin ses, video ve vektörel grafiği desteklemediği için ilerleyen yıllarda web geliştiricilerin Ajax tekniğini bırakabileceklerini iddia etmiştir [2].

Ajax'ı oluşturan tüm teknolojilerin aslında var olduğu halde bugüne kadar keşfedilmemesi ve bu tekniğin Google şirketi tarafından birçok uygulama da aktif olarak kullanılması nedeniyle konu hakkında bir çok yabancı kitap yazılmıştır.

Bu konuda yazılmış ilk kitaplardan biri de Holzner, S.'nin Şubat 2006'da yayımladığı "Ajax For Dummies" adlı kitaptır. Bu çalışmada, Ajax uygulamaların nasıl yapılması gerektiğini başlangıç düzeyinden ele almış, Ajax kütüphanelerinden, Ajax'ı oluşturan teknolojileri sunucu tarafı bir programlama dili kullanarak açıklamıştır [3].

Konu ile ilgili diğer kitaplarda da benzer başlıkların ele alındığı görülmüştür. Bu konuda yayımlanmış ilk Türkçe kaynak ise Özveri, G.'nin Haziran 2007'de orijinal adı "Professional Ajax" olan çeviri kitabıdır. Çalışmanın yazarları Zakas, N.C., McPeak. J. ve Fawcett, J. diğer kitaplarda da olduğu gibi Ajax mantığı ve nasıl kullanıldığı üzerinde durmuş, örnek uygulamalarla çalışmayı bitirmişlerdir [4].

5-7 Ekim 2007 tarihlerinde Meksika'da düzenlenen "2007 4th International Conference on Electrical and Electronics Engineering (ICEEE 2007)" isimli konferansın sonuç bildirgelerinden biri de Ajax tekniği ile ilgilidir. Zepeda, J., S. ve Chapa, S. V. tarafından hazırlanan makalede, Ajax tekniğinin tanımı, çalışma mantığı, avantaj ve dezavantajları üzerinde durulmuş, Ajax'ın Javascript temelli olduğu için güvenlik sorunları çıkabileceğinden bahsetmişlerdir [5].

Yapılan literatür ve kaynak taramaları sonucunda, Ajax'ın sunucu performansına etkisi konusu sayfa yenilenmesine ihtiyaç duymadığı için bant genişliğini rahatlattığı ifadeleriyle geçiştirilmiş olduğu görülmüştür.

Bu nedenle, Ajax'ın sunucu performansına etkisini konu alan bu çalışmada, web uygulamaları üzerinde bazı performans testleri gerçekleştirilmiş ve sonuçlar çalışmanın sonunda belirtilmiştir.

Bu çalışmanın son dönemleri olan Mart 2008'de, Tennessee Üniversitesinden Smullen III, C., W. ve Smullen S., A., "An Experimental Study of AJAX Application Performance" isimli bir makale yayımlamışlardır. Bu makalede, Ajax tekniği ile klasik teknik karşılaştırılmış, her iki tekniğinde sunucu performansındaki etkileri sayısal verilerle ifade edilmiştir. Ajax ve klasik teknikle ayrı ayrı hazırlanmış bir formun, üniversite öğrencileri tarafından doldurulmaları istenerek; her iyi teknik test edilmiştir. Bu makalenin sonucunda Ajax tekniğinin sunucu performansında yaklaşık % 54 oranında artış gözlenmiştir [6].

3. MATERYAL VE METOT

Bu çalışmada, Ajax tekniđi kullanılarak yapılandırılmış web sayfalarının, mevcut klasik yöntemle yapılmış web uygulamalarından daha performanslı olduğunun ispatlanması amaçlanmıştır.

Bu kapsamda gerçekleştirilen performans testleri üç aşamada uygulanmıştır.

- Birinci aşamada, test materyali olarak; bir web sitesinin klasik teknikli ve Ajax teknikli iki ayrı uyarlaması kullanılmıştır.
- İkinci aşamada, ücretsiz e-posta hizmeti veren Hotmail ve Yahoo servislerinin Ajax ve klasik teknik uyarlamaları aynı iş yüküne tabii tutulmuş ve test edilmiştir.
- Son aşama da ise Ajax tekniđinin veritabanı uygulamaları üzerindeki performansı test edilerek değerlendirmeye alınmıştır.

İlk aşama için geliştirilen web sitesinde aşağıdaki işlem basamakları uygulanmıştır:

- Web sitesindeki tüm bağlantıların çalıştırılması
- Fotoğraf galerisinden dört adet resmin tıklanarak orijinal boyutunun gösterilmesi
- İletişim sayfasındaki formun doldurularak sunucuya gönderilmesi

İkinci aşama da test edilen e-posta servisleri için belirlenen iş yükü aşağıdaki gibidir:

- Gelen e-postalardan birinin okunması
- Gelen e-postalardan birinin silinmesi
- İkinci bir adrese e-posta gönderilmesi

Test esnasında silinen, okunan ve gönderilen e-postalar birbirinin aynısı olup aralarında boyut bakımından hiçbir fark olmamasına dikkat gösterilmiştir.

Üçüncü test aşaması için gerçek dünyaya uygun olması bakımından ilk önce 20,000 kayıtlı veritabanı uygulaması geliştirilmiştir. Bu uygulama ile veritabanına kayıt ekleme, düzeltme ve silme işlemleri yapılabilmektedir. Test aşaması için aşağıdaki işlem basamakları belirlenmiştir:

- Bir kişiye ait kişisel bilgilerin girilmesi
- Arama sonucu bulunan bir kişinin kayıt bilgilerinin düzenlenmesi
- Arama sonucu bulunan bir kişinin tüm kayıtlarının silinmesi

Ajax tekniğinin sunucu performansına etkisini ölçmek için yapılan bu testler, veritabanındaki kayıt sayısı artırılarak devam etmiştir. Veritabanında bulunan kayıt sayısı sırasıyla 30,000'e ve 40,000'e çıkarılmış ve yukarıda açıklanan işlem basamaklarıyla test tekrarlanmıştır.

Geliştirilmiş olan web sayfalarına uygulanan performans testlerinin sonuçlarını görebilmek için Wapt ve WebPerformance Suite isimli uluslararası kabul görmüş yazılımlar ile Microsoft tabanlı Fiddler Web Debugger yazılımı kullanılmıştır.

4. YENİ WEB UYGULAMALARI

Bu bölümde içinde bulunduğumuz on yıla adını veren web 2.0 dönemi hakkında bilgi verilmiş ve bu dönemin içindeki en önemli buluş olan Ajax tekniğinin kullanım yolları, avantaj ve dezavantajları, hangi internet şirketlerinin bu tekniği kullandığı açıklanmıştır.

4.1. Web 2.0

İnternet her yeri o kadar çok etkilemiştir ki içinde bulunduğumuz çağın adı bilişim çağı olarak adlandırılmaktadır. Artık yegâne bilgi kaynağımız kütüphanelere çok güçlü bir alternatif olarak web sitelerini kullanmaktayız. Hızlı, kolay ve çok geniş olması web'i cazip kılmakta ve artık internet demek web sitesi demek anlamına gelmektedir.

İlk web sitesinin 1991 yılında yapımıyla başlayan web teknolojisinde kullanılan teknik ve yöntemler, genel olarak Web 1.0 olarak adlandırılmaktadır. Yıllar içinde yapılan web sayfalarının kullanıcı ile etkileşimli olmaması, almaktan ziyade sadece vermek üzerine kurulu olması 2000'li yıllarda yepyeni bir döneme geçilmesi gerektiğinin sinyallerini vermiştir.

Ekim 2004'de yapılan Tim O'Reilly ve MediaLive International tarafından organize edilen, web dünyasının önemli şirketlerinin de (Google, Yahoo, Msn, Amazon, Ebay...) katıldığı bir konferansta web'in geleceği tartışılmıştır. Artık web 1.0 olarak adlandırılan web düzeninin çökmekte olduğu yepyeni bir oluşum olması gerektiği savunulmaktadır [2].

Web 2.0 genel bir tanımla; bilgi paylaşımının ve bilgiyi paylaşıp sunmada insanların katılımcılığının önem kazandığı yepyeni bir dönemdir.

Ajax, Rss(Rate Site Summary), Xml(eXtensible Markup Language), CSS(Cascading Style Sheet) gibi web teknolojilerinin gelişimi artık web 2.0 gerçeğinin göz ardı edilemeyeceğini göstermektedir. Php (Hypertext Pre-

Processor), Asp(Active Server Page), Asp.net gibi sunucu tabanında çalışan kullanıcı etkileşimli dillerin de kullanılmaya başlanması da bu değişimi kuvvetlendirmiştir.

Web 1.0'da kullanıcılar sadece okuyucu rolündedir. Sadece bilgiyi tüketen konumda olan kullanıcı, artık web 2.0 ile bilgiyi oluşturan bir topluluk oluşturmaktadır.

Kullanıcılar hiçbir web dilini bilmemesine rağmen birkaç dakika içinde kendi sitelerini oluşturup, bilgilerini paylaşabilmelerini sağlayan blog sistemi bu değişimin en güzel örneklerdendir.

İnternet dünyasının en iyi ve en hızlı arama motoru sayılan google.com sitesinde "web 2.0" terimi aratıldığında 75 milyondan –(haziran 2008)- fazla sonuç elde edildiği görülmektedir.

Aynı arama motoru kullanılarak "web 3.0" terimi arandığında ise 39 milyondan fazla sonuç alındığı görülmektedir. Bu artık web 2.0 döneminin de sonuna gelindiğini ve yakın gelecekte web 3.0 dönemine geçileceğini göstermektedir.

Web 2.0 kavramını altı ana başlıkta inceleyebilir:

Xml'e geçiş : Html(HyperText Markup Language) dili bilgileri derleyip, kullanıcılara sunmada kullanılan en temel web dilidir. Ancak sayfa üzerinde yapılan yenilikleri, kullanıcılara anında iletmek istenildiğinde kullanıcının sayfayı tekrar ziyaret etmesini beklemekten başka bir yol bulunmamaktadır.

Değişen içeriğin kullanıcılara dağıtmak için html taglarının daha dinamik bir şekilde kullanımına olanak sağlayan Xml ile geliştirilmiş Rss Feed'ler bu soruna çare olmaktadır. Sayfayı sadece bir kere ziyaret eden bir kullanıcı sitenin Rss Feed'ini, listesine kaydetmesiyle site ile ilgili haberleri anında görebilmektedir.

Web sayfalarındaki gelişim: İlk web siteleri birden çok alt sayfanın tek tek tasarlanması ve içeriğinin doldurulmasıyla meydana getirilen statik anlamda sitelerdir. Bu dönem “sayfalar” dönemi olarak kabul edilmektedir. Daha sonra bu siteler Flash, CSS, Javascript gibi dinamik ya da interaktif uygulamalarla desteklenerek ufakta olsa bir gelişim yaşamıştır. Esas doyuma ise Xml ve dinamik içeriğin de kullanılmaya başlanmasıyla ulaşılmıştır.

Online bir ansiklopedi olan Wikipedia'nın içeriği ziyaretçileri tarafından doldurulmakta, açık arttırma-satış sitesi olan ebay.com'daki ürünlerin satış fiyatı da ziyaretçiler tarafından belirlenip satışı yapılır şekildedir.

İçeriğin sunumundaki değişimi: Web 2.0 anlayışıyla birlikte internet sitelerinin içeriklerini sunuş biçimleri de değişmeye başlamıştır. Gelişen tekniklerle birlikte kullanıcıların dikkati de yükselmiştir. Rss feedler, arama motorlarının önemi, video haberleri kullanan siteler daha ön plana çıkmıştır.

Kullanıcıyla etkileşim ve kullanıcıya daha çok söz hakkı verilmesi: Sitelerin statik içerikli olmasından çıkıp, site içeriğinin kullanıcılar tarafından oluşturulması, kullanıcının ihtiyaçları ve eğilimleri doğrultusunda içeriğin düzenlenmesi bu dönemin gerekliliğidir.

Etiketleme: Web 1.0 döneminde site içeriği hakkındaki bilgi meta taglarına yazılan anahtar kelimelerle sağlanmaktadır. Arama motorlarında indekslerde üst sıralarda yer almak için meta tagları içine site içeriğiyle ilişkisi sorgulanmadan yüzlerce kelime yazılmaktadır. Arama sonucunda bulunan bir sitenin, aranılan bilgiyle alakasız içerikte çıkması güvenilirliği düşürmektedir.

Ancak Web 2.0 kapsamında etiketleme yöntemi ile kendi tagını kendi kayıt eden kullanıcı dönemi başlamıştır. Flickr, Del.icio.us, msn spaces, link gibi siteler ortak bir havuz oluşturmaktadır. Bu sitelerin en ünlüsü olan del.icio.us sitesinde “Ajax” kelimesi aratıldığında içeriğinde Ajax terimi geçen tüm site isimleri listelenmektedir.

Tasarımdan daha ziyade yapının önemi: Web 1.0 döneminde tasarım her şey iken yeni dönem içinde görsellik arka plana alınmış olup, önemli olan bilgiyi kullanıcıya iletimde kullanılacak yapının sağlamlığı ve kullanışlılığı olmuştur. Web geliştiricilerinin tasarım boyutundan ziyade teknik boyutunu da düşünmeleri ve programlama kısmına adım atmaları gerekmektedir.

Çizelge 4.1. Web 1.0 uygulamalarının web 2.0 dönemindeki gelişimi [2]

Web 1.0	Web 2.0
DoubleClick	Google AdSense
Ofoto	Flickr
Akamai	BitTorrent
mp3.com	Napster
Britannica Online	Wikipedia
Kişisel web siteleri	bloglama
Domain ismi spekülasyonu	arama motoru optimizasyonu
Sayfa görünümleri	tıklama üzerinden ücretlendirme
yayınlama	katılım
içerik yönetim sistemi	wikis
Klasörler	etiketler

Çizelge 4.1.'deki ilk örnekte sitelerde yer alan kaplayan hareketli ve alakasız bannerlardan kurtulup, web reklamları konusunda yeni bir anlayış olan google adsense karşılaştırılmıştır. Eski web sitelerinin içinden çıkan can sıkıcı reklam bannerlarının yerini hızla alan; metin tabanlı, küçük, rahatsızlık vermeyen ve en önemlisi de sayfa içeriği ile ilgili çıkan google adsense reklamlarıdır.

Kişiyeye ait resimlerin saklanmasına olanak sağlayan siteler artık yerini istenilen resimlerin internette yayınlanmasına ve milyonlarca kişinin bu fotoğraflara ulaşmasına izin veren siteler haline dönüşmüştür.

Sadece mp3 indirilmesine izin veren mp3.com, kullanıcıların birbirine mp3 gönderebilmelerini sağlayan napster ve türevleri internet yazılımları karşısında gerilere düşmüştür.

Dijital ansiklopedi olarak kullanılan Britannica Online, Wikipedia olarak web 2.0'da yerini almıştır. Wikipedia bir başlığın kullanıcı tarafından girilebileceği ve başka bir kişi tarafından değiştirilebileceği bir online ansiklopedidir. Bu sistemde bilgiler tamamıyla kullanıcılar tarafından girilmektedir.

Web 1.0 döneminde akılda kalıcı alan isimleri bulmak çok önemli iken günümüzde arama motorlarında ilk sırada çıkmak çok önemli olmuştur. Bunun nedeni kullanıcılarının çoğunluğunun internete ilk girişlerini arama motorları üzerinden yapmakta olmalarıdır. Web sayfalarının ziyaret edilme istatistiklerini tutan alexa.com verilerine göre -Haziran 2008- tüm zamanlarda günlük en çok ziyaret edilen siteler sıralamasında arama motoru olan google.com ikinci sıradadır. Bir diğer kuvvetli arama motoruna sahip yahoo.com ise birinci sırada yer almaktadır.

Ne kadar çok ziyaretçi o kadar çok popülerlik niteliğindeki sayfa görüntülenme sayısı yerini görüntülenme başına para kazanmaya bırakmıştır. Dosyaları klasörlerle tasnifleme özelliği kullanımı daha kolay ve yaygın olan *etiketleme* yöntemine geçilmiştir. Bir site içerisinde aynı etikete sahip tüm veriler tek tıklamayla kolayca listelenebilmektedir.

Özetle web 2.0'daki bilgi kavramı yeniden şekillenmiş ve artık bilgi istenildiği zaman, istenildiği ortamdan, istenildiği şekilde erişebileceği bir hal almıştır.

Ajax, son nokta olan kullanıcıya daha zengin bir deneyim oluşturmakla ilgilidir. Ajax tamamen web 2.0 değildir ve web 2.0 da Ajax değildir. Ajax web 2.0'ın belirttiği gelecek kuşak kullanıcı deneyimini oluşturmanın önemli bir parçasıyken çok daha büyük bir bilmecenin sadece bir bölümüdür [4].

4.2. Ajax Tekniđi

Ajax; *Asynchronous JavaScript and XML*(Eşzamansız Javascript ve XML) sözcüklerinin baş harflerinin bir araya gelmesinden oluşmuş bir terimdir. Ajax terimi dilimizde “eyceks” olarak telaffuz edilmektedir.

Ajax; Html/xhtml, Xml, CSS, Dom(Document Object Model), Javascript ve Xmlhttprequest nesnesi ile sunucu taraflı dillerden birisinin (php, asp, jsp vb.) beraber kullanılarak masaüstü yazılımlara benzer etkileşimli web ara yüzleri ve uygulamaları hazırlamak için kullanılmakta olan bir web geliştirme yaklaşımıdır.

Ajax terminolojisi Şubat 2005'te Jesse James Garrent'ın başkanı ve kurucusu olduğu şirketin web sitesinde yayınladığı “*Ajax: A New Approach to Web Applications*” adlı makale ile duyulmuştur. Bu makalede web uygulamalarının kullanım rahatlığı bakımından masaüstü uygulamaları ile arasındaki farkın kapanabileceğinden ve Google şirketinin bu konu üzerinde yoğun şekilde çalıştığından bahsetmektedir. Google sihirli kelimesinin bu makalede yer alması Ajax'a olan ilgiyi de bir anda arttırmıştır.

Ajax uygulamaları masaüstü uygulamalarına göre bazı avantajlar sağlar. Çünkü özel sistemler, sanal makine, tarayıcı eklentileri veya kullanıcının masaüstü bilgisayarına yüklenen harici programlardan bağımsızdır. Bu özellikleri internet kullanıcılarının ilgisini fazlasıyla çekmiştir ve şimdi çok daha popülerdir. Bu uygulamalar sadece istemci taraflı bir tarayıcıya ihtiyaç duyar ve daha kolay, hızlı, esnek ve zengin kullanıcı etkileşimi sunar. Ajax web uygulamalarının masaüstü uygulamalarına doğru ilerlemesi olasılığı şimdi gerçeğe dönüşmüştür. Google Docs buna iyi bir örnektir. Google Docs ile belgeler yazabilmekte veya tablolar hazırlanabilmektedir. Ajax uygulamalarının sunduğu bu tüm avantajlar sayesinde, bunu nasıl gerçekleştirileceğini bilmek büyük bir ilgi haline gelmiştir. Ajax terimi bazı insanlar için açık değildir ve onun hakkında yanlış fikirler edinilmiştir. Bazıları yeni bir dil veya yeni teknoloji veya yeni ürün veya sadece başka yeni bir

framework olduğunu düşünürler. Fakat tüm bu terimler doğru değildir. Ajax bir teknoloji ve yeni bir programlama dili değildir. Ajax, web uygulamaları programcıları için yeni bir düşünce biçimi, tasarım, geliştirme ve yeni bir stildir. Üstelik Ajax, tüm tarayıcı ve platform uyumluğu ile var olan standart teknolojileri kullanan yeni bir tekniktir [5].

Ajax anlaşılması ustalık gerektiren bir tekniktir. Web siteleri yapmak için kullanılan araçların çoğu iki sınıfa ayrılabilir: Html, Css, ve Javascript gibi istemci teknolojileri ve Apache, Php ve Mysql gibi sunucu taraflı teknolojilerdir. Ajax, istemci ile sunucu arasında bir yerdedir [7].

Günümüzde bir uygulama iki ortamda çalıştırılmaktadır. Birincisi masaüstü, ikincisi de web ortamlarıdır. Yazılan uygulamalar da bu ortamlara göre.

Masaüstü uygulamaları, kişisel bilgisayarlarda, işletim sistemi ile beraber çalışan oldukça dinamik bir yapıdaki uygulamalardır. Tıklama, yazma, gezinme yollarına hızlı cevap verirler. Neredeyse hiç internete ihtiyaç duymazlar.

Bazı web uygulamaları vardır ki dakikasında güncellenir ve masaüstü uygulamalarında elde edemeyeceğimiz hizmeti sunmaktadırlar (haber siteleri gibi). Web uygulamalarında, masaüstü uygulamaları kadar hızlı ve dinamik bir yapı beklenemez. Çünkü web uygulamalarının yapısında; sunucunun yanıt vermesi, ekranın yenilenmesi, isteğin geri gelmesi ve yeni sayfa yaratması gibi işlemlerden dolayı bekleme söz konusudur.

Ajax, bu noktada devreye girmektedir. Ajax bir masaüstü uygulamasının işlevselliği ve etkileşimi ile her zaman güncellenen bir web uygulaması arasındaki mesafeyi kapatmaya, bir masaüstü uygulamasında bulunabilecek dinamik yapıyı web uygulamalarında bize sunmaya çalışmaktadır.

Ajax tek başına bir teknik değildir. O her biri sürekli gelişen birkaç teknolojinin birlikte oluşturduğu yepyeni bir oluşumdur.

Ajax daha canlı ve yetenekli ara yüzlerin oluşturulmasını mümkün kılmaktadır. Web uygulamalarında en çok şikayet konusu olan yavaşlık ve ara yüzün çirkin ya da kullanışsız olması sorununa iyi bir çözüm olarak durmaktadır.

Ajax yeni web uygulamaları yapılacağı zaman aşağıdaki yararları da beraberinde getirmektedir:

- Daha esnek, daha iyi web siteleri ve web uygulama yaratılmasına olanak sağlamaktadır.
- Popülaritesinden dolayı web geliştiricilerin alışagelmiş örneklerde web siteleri yapmalarını engellemektedir.
- Ajax var olan teknolojilerden yararlanmaktadır.
- Ajax web geliştiricilerin var olan yetenekleri kullanmaktadır.
- Ajax, web tarayıcıların var olan fonksiyonlarıyla sorunsuz uyum sağlamaktadır [8].

Ajax mükemmel değildir. Her zaman zengin internet uygulamaları için web geliştiricileri ve teknoloji şirketleri en iyisini bulmak için çalışmaktadırlar. Bugün Ajax vardır ve çalışmaktadır. Hem kullanıcıların hem de web geliştiricilerin istedikleri gibi her web tarayıcısında ve platformda çalışabilmektedir. Hiç kimse Ajax'ın ne zaman zengin internet uygulamalarında ilk tercih edilen teknik olacağını söyleyemez. Fakat çok fazla Ajax desteği sunulması bu tercihin yakın gelecekte olduğunu göstermektedir [9].

Ajax her ne kadar kullanıcılarına cazip gelecek güzellikler sunuyor olsa da kullanımda dikkat edilmesi gereken bazı durumlar da vardır.

Ajax tekniği ihtiyaç duyulan yerlerde kullanılmalıdır. Kullanmak olmak için kullanmak büyük bir yanılgıdan başka bir şey değildir.

Bazı kullanıcılar tarayıcıların geri, ileri, durdur gibi butonlarını sıkça kullanırlar. Ajax uygulamalarında bu butonlar işlevselliğini yitirdiği için sıkıntılar çıkabilir. Bu sorunu çözen Ajax kütüphaneleri mevcuttur. Yazılan bir Ajax uygulamasında bu sorun mutlaka giderilmelidir.

Tamamıyla Ajax tekniği kullanılmış bir çalışmada kullanıcılara sık kullanılanlara ekleme özelliği ayrıca verilmelidir. Aksi takdirde çalışmanın bir bölümünü görüp sık kullanılanlara eklemek isteyen bir kullanıcı, istediği bölümü ekleyemeyecek sadece başlangıç sayfasının sık kullanılanlara eklemiş olacaktır.

Kullanıcılar bir web sitesinde hangi tekniğin kullanıldığını bilmeyebilir. Ajax tekniğinde tüm istekler geri planda yapıldığı için bir süre beklemek gerekmektedir. Bu bekleme sürecini kullanıcıya mutlaka bir mesaj ile belirtmek zorunluluğu vardır. Aksi takdirde kullanıcı neden beklediğini bilemeyecek, birden fazla farklı tıklamalar yapabilecek ve en önemlisi siteyi terk edecektir.

Bir Ajax uygulamasından kullanıcıların iyi verim alabilmesi için yüksek hızlı bir internet bağlantısına sahip olmaları gerekmektedir. Nitekim yahoo mail, hotmail ve gmail gibi Ajax'ı iyi derecede kullanan şirketler "eğer bağlantınız yavaş ise klasik görünümü deneyiniz" gibi mesajlarla kullanıcılarını uyarmaktadır.

Ajax javascript tabanlı olduğu için her javascript her tarayıcı da aynı sonucu vermeyebilir. Mutlaka her platformda çalışmalar test edilmelidir. Gereksiz büyüklükteki javascript kodları tarayıcıları ve dolayısıyla kullanıcıları yoracaktır. Hazır kütüphanelerden sadece işe yarayan kısımları kullanmak gereksiz kod kısımlarını ayıklamak yapılacak en uygun işlemlerdir [10] .

4.2.1. Ajax'ın arkasındaki teknolojiler

Jesse James Garrent makalesinde bir Ajax tekniğini oluşturan teknolojileri aşağıdaki gibi tanımlamaktadır:

- XHTML ve CSS kullanan standart-tabanlı sunumlar;
- Dinamik görünüm ve DOM kullanan işlemler;
- Data değiş tokuşu ve XML kullanan işlemler;
- XMLHttpRequest kullanarak eşzamansız veri alımları;
- Javascript kullanan diğer her şey.

Bu teknolojilerin hepsi kullanılabilir ancak Ajax için olmazsa olmaz olan html/xhtml, dom ve javascript'tir.

Bununla birlikte, Ajax yalnızca gelip geçici bir akım değildir; web sitesi geliştirmek konusunda güçlü bir yaklaşımdır ve yeni bir dili öğrenmek kadar zor değildir.

Xml: eXtensible Markup Language'in kısaltılması olan Xml, veri tanımlayan bir biçimdir [7].

Xml hakkında birinci ve en önemli nokta onun bir dil olmadığıdır. Aslında diğer dilleri yaratmak için kullanılan metadildir. Xml, bu dillerin nasıl oluşturulacağını kurallarını belirler. Her dil birbirinden farklıdır, fakat hepsi içeriğinde tagları kullanır [11].

Xml, bir meta dildir. Markup Language (işaretleme dili) yani web dilleri yaratmak için kullanılan bir teknolojidir.

Xml'de html gibi işaretleme etiketlerini kullanan dildir. Html ve Xml arasındaki temel fark Xml işaretleme etiketlerinin bilginin içeriğini tanımlamak için kullanılmasıdır.

Xml metin tabanlı tanımlama yapmamızı sağlamaktadır. Verileri tanımlamamız için metin tabanlı bir ortam söz konusudur. Metin tabanlı olması verileri her ortamda sorunsuz olarak saklanmasını iletimini sağlar. Güvenlik duvarı uygulamalarına takılmaları gibi bir durum söz konusu değildir. Diğer bir ifade ile Xml günümüzün ve geleceğin veri depolama ve aktarım ortamıdır [12].

Xhtml(Extensible hypertext markup Language): Html internetin ortak dilidir. İçeriklerin biçimlendirilmesi sağlayan taglardan oluşmuştur. Etiketlerin doğru sırayla kapatılması sayfanın iyi şekillenmesiyle doğrudan bağlantılıdır.

`<p> Ajax bir dil değildir </p> ` örneğindeki kullanım hatalıdır. İlk önce kullanılan `<p>` etiketi en son kapatılmalıdır. İlk giren son çıkar mantığı bu örnekte uygulanmamıştır. Kapanış etiketlerinin doğru sıralaması aşağıdaki gibi olmalıdır:

`<p> Ajax bir dil değildir </p>`

Xhtml, Xml olarak şekillendirilmiş basit Html'dir.

Bu dilde bütün etiketler küçük harfle yazılmalı ve bütün etiketler kapatılmalıdır. Satır sonu ve resim gibi bağımsız öğeler için sonuna bölü işareti konur: `
`, ``.

CSS: Stil şablonları kodlar içine yerleştirilmiş seçiciler ile içeriğe biçim vermek için kullanılmaktadır. CSS dosyası ayrıca kodlanıp, sayfa içine de dışarıdan çağırılabilir. CSS'nin en güzel yanı kullanımındaki esnekliktir.

```
< style type="text/css" >
<!--
h1 {color:red; font-size:15}
-->
</style>
```

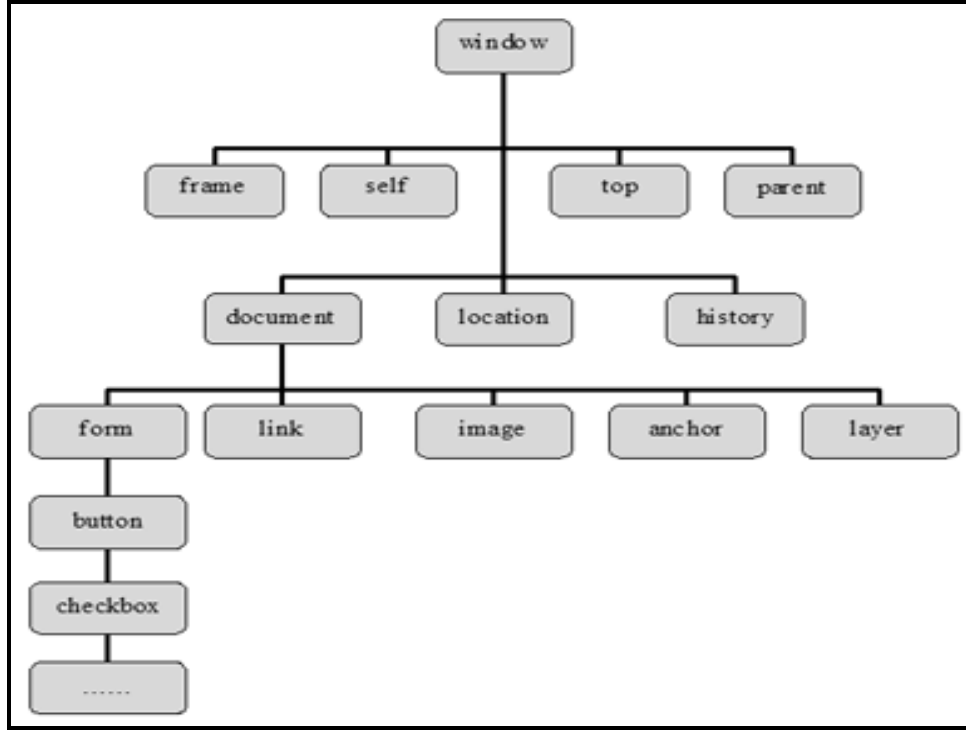
Bu kod örneği <head> etiketleri arasına yerleştirilir. Bu stil belirleme, sayfa içinde kullanılacak <h1> etiketlerinin biçimini belirlemektedir. Buna göre <h1> etiketinin kullanıldığı her yazı kırmızı renkte ve 15 punto büyüklüğünde olmalıdır.

Dom: Javascript ile web uygulamasını daha çabuk cevap verebilir hale getiren bir yapıdır. Web tarayıcısındaki Html sayfalarını Dom kullanarak sunucuya gitmeden değiştirerek çabukluk sağlanırken, sunucu ile veri alışverişi Xml olarak kullanıcı fark etmeden yapılmaktadır.

Dom, tarayıcı ekranı içerisindeki nesnelerin adreslerinin yazılı olduğu bir yol haritasına benzer bir tablodur. Her obje bir diğerinin içerisinde yer alabileceği için hiyerarşik bir yapıya sahiptir [13].

Internet tarayıcıları girilen internet sayfasını belge ve içerisinde bulunan tüm elemanları da nesne olarak kabul eder. Buna göre resim, yazı, form gibi tüm elemanlar nesnedir. Dom sayfa içindeki herhangi bir nesnenin özelliğine müdahale edebilmeyi, nesne özelliklerini değiştirebilmeyi sağlamaktadır.

Dom tablosunun en üstünde *windows* objesi yer alır. Bu web tarayıcısı anlamına gelmektedir. Web sayfaları ise hemen bir alt kademede yer alır ve *document* olarak ifade edilmektedir. Html sayfasının sahip olduğu diğer nesnelere *document* objesinin altında yer almaktadır. Üst basamaktan alt basamağa geçildikçe araya nokta konur ve anlatılmak istenen objenin yolu belirlenmiş olur. Örneğin web sayfasının zemin rengi ifadesi dom modelinde şu şekilde ifade edilir: *window.document.bgcolor*



Şekil 4.1. Dom tablosu

Sunucu ile veri alışverişi *XMLHttpRequest* nesnesi kullanılarak yapılmaktadır. Bu nesne Internet Explorer'da bir ActiveX nesnesi, Mozilla web tarayıcılarında ise doğrudan olarak javascript tarafından erişilen bir nesne olarak geliştiricilerin kullanımına hazırdır.

XMLHttpRequest nesnesi ile sunucu tarafında çalışan ve Xml üreten bir uygulama ile normal bir web uygulaması gibi konuşulmaktadır. Yani sunucudaki uygulama gelen isteklere Html değil de Xml üreterek cevap vermektedir.

CSS belgenin sunumunu belirlemek için kullanılabilir, javascript ve Dom bileşimi de belgedeki öğelerin davranışlarını özelleştirmek için kullanılabilir [7].

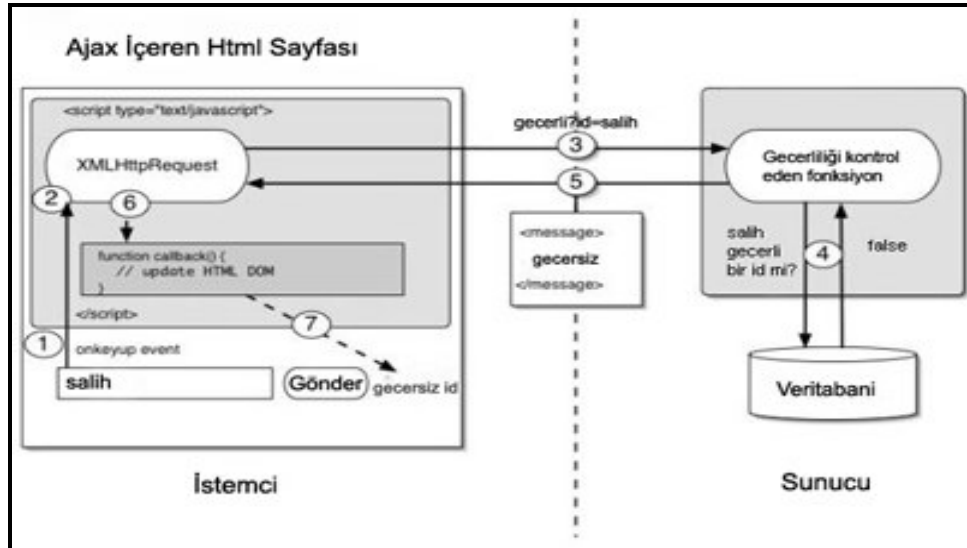
XMLHttpRequest: Xhr, istemci ve sunucu arasında iletişimi sağlayan, web sunucularından gelen Xml ve diğer metin türü verilerin transferi için

JavaScript, web tarayıcıları ve diğer script dilleri tarafından kullanılan bir API'dir¹. Xhr, dom'un bir parçası değildir.

Microsoft Internet Explorer 5.0 ile MSXML adıyla bir ActiveX kütüphanesi tanıtılmıştır. Bu kütüphane içinde en popüler olanı XMLHttpRequest nesnesidir. XMLHttpRequest nesnesi bir uygulamanın herhangi bir yerinden geliştiricilerin http isteklerini başlatabilmeleri için oluşturulmuştur.

Mozilla, Firefox tarayıcısında XMLHttpRequest nesnesini kopyalarak XMLHttpRequest adıyla aynı özellikte doğal bir Javascript nesnesi oluşturmuştur. Microsoft'ta kendi doğal XMLHttpRequest nesnesini IE 7.0 ile oluşturmuştur. Böylece bugün en çok kullanılan web tarayıcılarının hepsi bu XMLHttpRequest nesnesini desteklemektedir [4].

XMLHttpRequest nesnesi (veya benzer mekanizmalar), kullanıcı çalışırken arka planda istekleri gidip almak ve sunucu ile eşzamansız iletişim için kullanılır. Şekilde bu nesnenin Ajax tekniğindeki kullanımı gösterilmektedir [14].



Şekil 4.2. XMLHttpRequest nesnesi

¹ Application Programming Interface: Yazılım programlama ara yüzü bir yazılımın başka bir yazılımın fonksiyonlarını kullanabilmesi için yaratılmış bir tanım bütünüdür.

Xhr nesnesi ile çağırılan sonuçlar sunucudan Xml, Html veya düz metin olarak geri dönebilir.

Xhr, web sayfası ile sunucu arasına yerleştirilir. Ajax tekniğinde, bir form doldurulduğunda formdaki tüm veriler doğrudan sunucuya değil bazı Javascript kodlarına gönderilir. Javascript kodu istekleri alır ve sunucuya gönderir. Tüm bu işlemler arka planda gerçekleşir. Form ekrandan kaybolmaz ve kullanıcılar arka planda sunucu ile haberleştiğinin farkına bile varmazlar. Sunucudan gelen yanıtlar, bu yanıtlarla ne yapılacağına karar verecek olan Javascript koduna yüklenir.

Bu işlemler XmlHttpRequest nesnesinin marifetleridir. Kullanıcıya hissettirmeden, arka planda diğer uygulamalar ile bağlantı kurar ve sonuçları kullanıcıya iletir.

Ajax uygulamalarında ilk yapılacak işlem Xhr nesnesini çağırmak ve isteklerimizi almaya hazır hale getirmektir.

Bir Ajax uygulamasının en önemli parçası sayılan XmlHttpRequest nesnesi bu çalışmanın ilerleyen bölümlerinde detaylıca açıklanmıştır.

Büyük ölçekli web uygulamaları gerçekleştiren internet tabanlı şirketler, web uygulamalarında kendi teknolojilerinin yanında Ajax kullanmaya başlamışlardır. Genel olarak Microsoft, Macromedia ve Sun Microsystem Ajax'ı, küçük programlarda kullanırlar ve çok önemli yazılımlar gerçekleştirirler [2].

4.2.2. Ajax mimarisi

Klasik web uygulamalarındaki kullanıcı ile uygulama arasındaki etkileşimi şöyle gerçekleşmektedir.

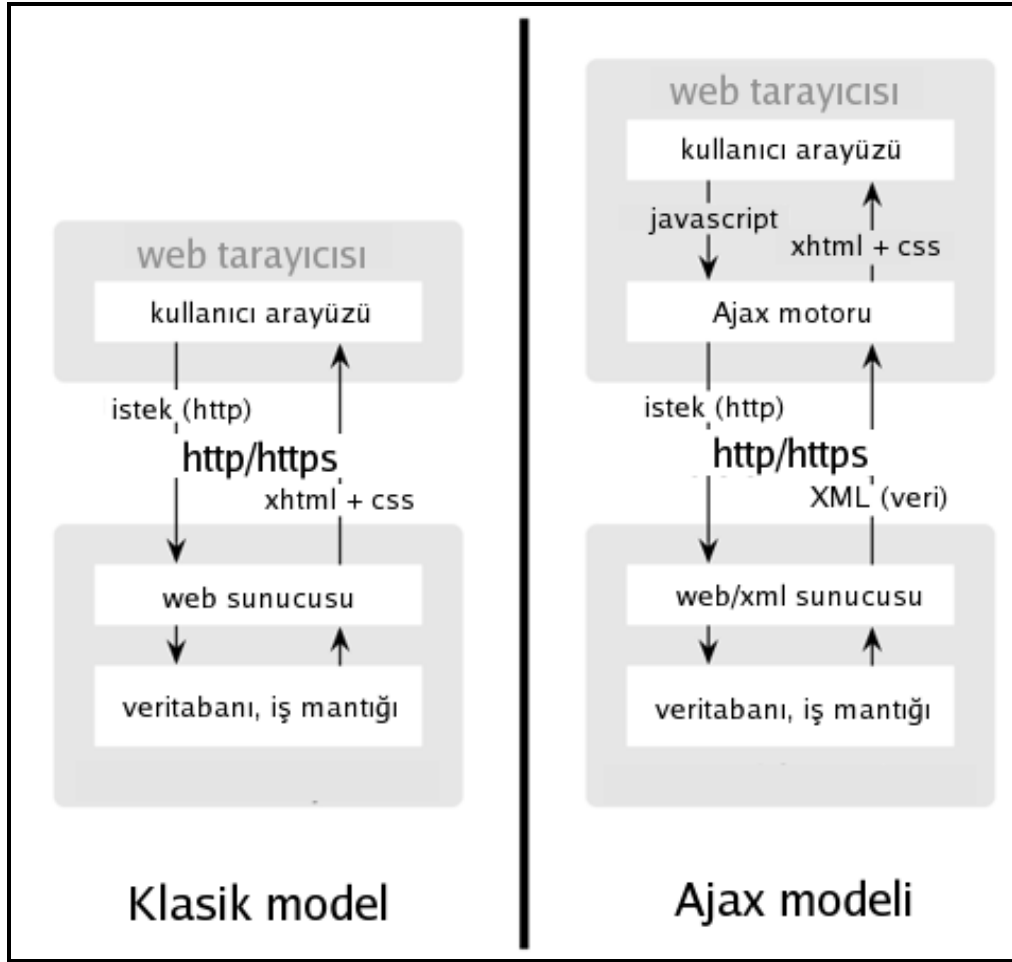
- Kullanıcı web tarayıcısı aracılığıyla bir istekte bulunur. Web adresi yazabilir, bağlantılara tıklayabilir ya da form doldurup gönderebilir.

- Web tarayıcısı girilen url için sunucudan istekte bulunur.
- Sunucu isteğe yeni bir sayfa ile cevap verir.
- Web tarayıcısı yanıt sayfasında gömülü resim, video, ses gibi objeler için sunucuya daha çok istek gönderir.
- Web tarayıcısı sunucudan gelen cevapları derleyerek sayfayı tüm içeriğiyle gösterir ve kullanıcıdan gelen yeni istekleri bekler.

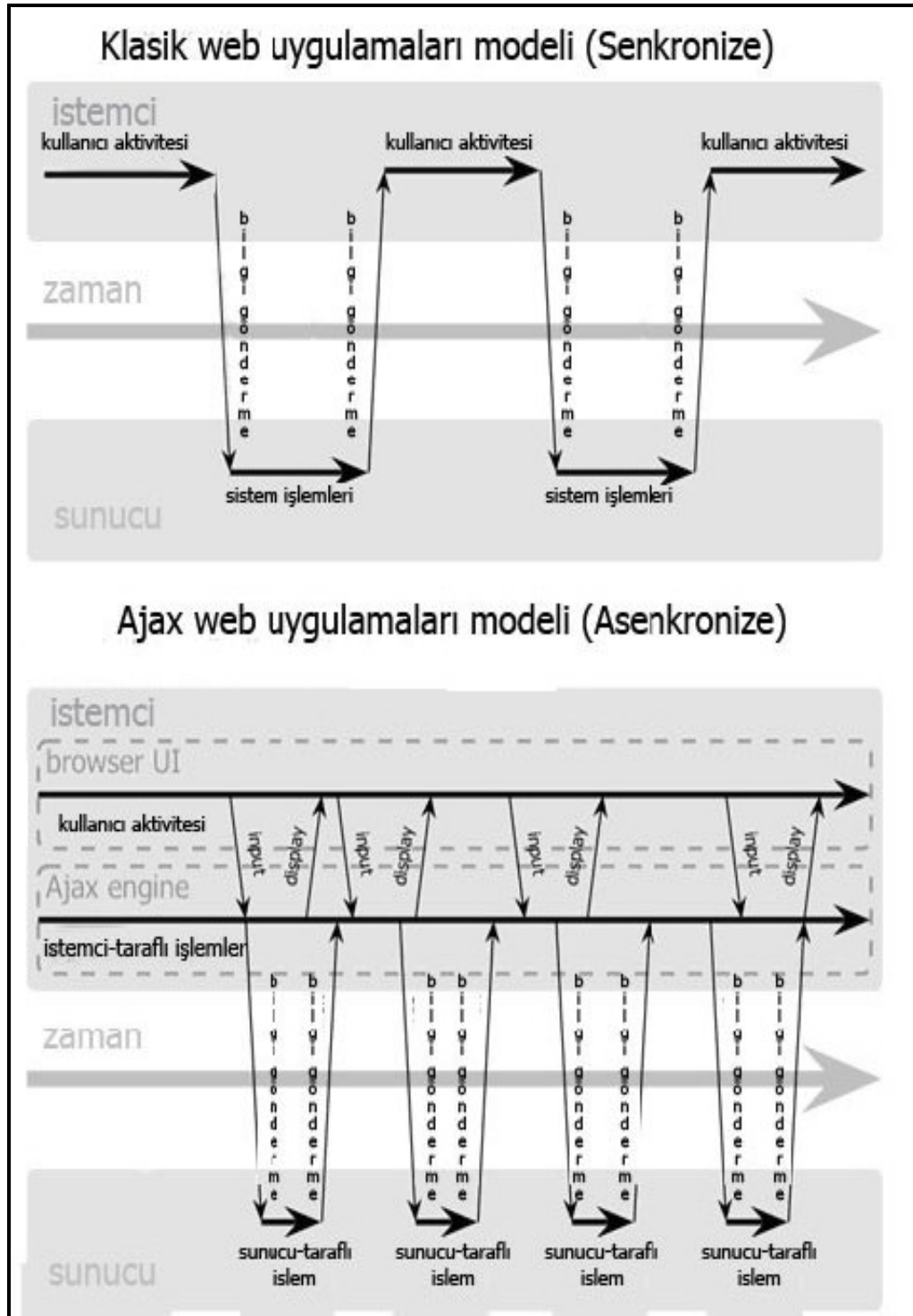
Bir Ajax uygulaması kullanıcı ve sunucu arasında Ajax motorunu arabulucu olarak kullanarak normal web uygulamalarındaki tıkla-bekle-tıkla-bekle olayını ortadan kaldırmıştır. Bu uygulamalara katman eklemeye benzeyen yöntem etkileşimli olarak daha az yanıtın işlem görmesine neden olur.

Bir Ajaxlı web sayfasının yüklenmesi sırasında oturum başlatılır. Web tarayıcı Javascript ve genellikle bir gizli çerçeve kullanılarak Ajax motorunu yükler. Bu motor kullanıcı tarafındaki sunucu ile haberleşmek ve tarayıcının isteklerine yanıt vermekle sorumludur. Ajax motoru eşzamansız olarak sunucudan bağımsız bağlantı ile kullanıcı ara yüzünü idare eder. Bu yüzden kullanıcı asla boş bir sayfa görmez veya beklemesini isteyen bir işaret görüp beklemez. Sunucunun bir şeyler yapmasını beklemektedir.

Şekil 4.3.'te klasik web uygulama modeli ile Ajax web uygulamaları modelinin çalışma biçimleri karşılaştırılmıştır. Şekil 4.4'te ise eşzamanlı ve eşzamansız çalışma metodlarının yapısı görülmektedir.



Şekil 4.3. Klasik web uygulama modeli ile Ajax web uygulama modelinin çalışma mantığı



Şekil 4.4. Klasik ve Ajax web uygulamalarındaki veri iletişim modelleri

Ajax web uygulamalarında, kullanıcının normal olarak oluşturduğu http istekleri sunucu yerine Javascript ve Ajax motorları alacaktır. Kullanıcıya giden her cevap hareketi sunucudan geri gelmeyebilir (basit veri doğrulama, hafızadaki veriyi düzenleme olayları gibi). Bu işlemleri Ajax motoru yapmaktadır.

Eğer motorun sunucudan cevap isteğine göre bir şeyler çekmesi gerekiyorsa (veri gönderme, yükleme, ara yüz kodu eklenmesi, yeni verilerin alınması gibi) motor bunları genellikle Xml kullanarak uygulamada kullanıcının beklemesini önleyecek şekilde eşzamansız iletişim kurarak halledecektir.

En basit haliyle Ajax, kullanıcının her yeni isteğinde web sayfasının tümünü yenileme zorunluluğundan kurtarır. Bunun yerine web sunucusundan yalnızca web sayfasında güncellenmesi istenilen bölüm çekilir ve ekrana o bölüm yansıtılır. Bu hem internet sitesini hızlandırmayı sağlar, hem de kullanıcıya daha kullanıcı dostu bir ara yüz ile hizmet verme imkanı tanır. Çizelge 4.2.'de her iki tekniğin özellikleri görülmektedir.

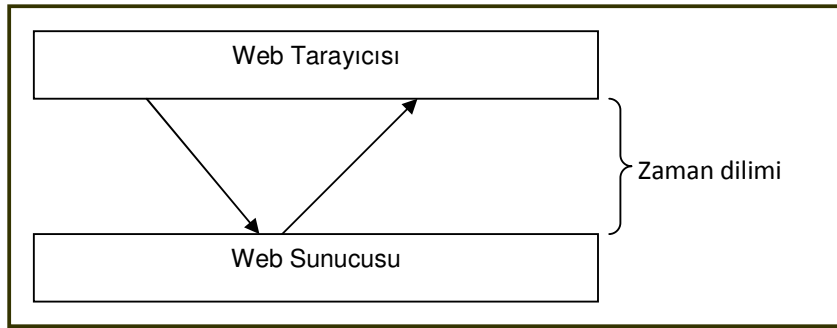
Çizelge 4.2. Klasik teknik ile Ajax tekniğinin karşılaştırılması

Klasik Teknik	Ajax Tekniği
Kullanıcı istekte bulunur, sunucu cevap verir. İletişim eşzamanlı gerçekleşir.	Kullanıcı istekte bulunur, sunucu cevaba hazırlanırken kullanıcı yeni bir istekte bulunabilir. İletişim
Formlarda GET veya POST metotlarıyla istekte bulunulur.	XMLHttpRequest nesnesi yardımıyla istekte bulunulur.
Web tarayıcısı sunucudan gelen yanıt ile tüm sayfayı yeniler.	Web tarayıcısı sunucudan gelen bir çok yanıt ile sadece sayfanın bazı bölümlerini yeniler.

4.2.2.1 Eşzamanlı ve eşzamansız iletişim

Bu bölümde eşzamansız ile eşzamanlı iletişim kavramları arasındaki farklar üzerinde durulmuştur.

Eşzamanlı iletişim: Eşzamanlı iletişim yönteminde, aynı zaman dilimi içinde sadece bir işlem yapılmasına izin verilmektedir. Web tarayıcısından uzaktaki sunucuya gönderilen istek, web sunucusunda işlenmekte ve tekrar geldiği yoldan web tarayıcısına geri dönmektedir. Bu işlem belli bir zaman almaktadır. Eşzamanlı çalışma yönteminde, geçen bu zaman diliminde başka bir istek gönderilememektedir.

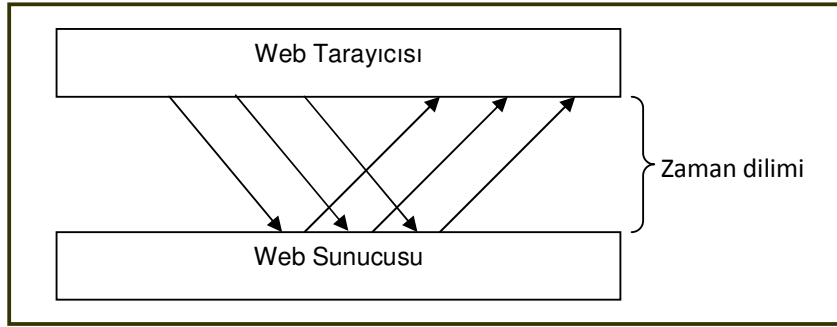


Şekil 4.5. Eşzamanlı çalışma yöntemi

Şekil 4.5.'te görüldüğü gibi, web tarayıcısından çıkan istek web sunucusuna iletilmektedir. Sunucu bu isteğe cevap vermekte ve istek sonucunu kullanıcıya geri göndermektedir.

Eşzamanlı çalışma, web sunucusu ile kullanıcı aynı bilgisayarda veya yakın olan bir ağ üzerindeyse sağlıklı çalışabilmektedir. Fakat web sunucusu ağır bir yük altındaysa veya web tarayıcısı web sunucusuyla yavaş bir bağlantı üzerinden etkileşime geçtiyse kötü bir performans vermektedir. Çünkü web tarayıcısı içerisindeki Javascript motoru, gönderdiği istek tamamlanıncaya kadar bloke edilmekte ve web tarayıcısı donmuş bir şekilde isteğin tamamlanmasını beklemektedir.

Eşzamansız çalışma: Eşzamansız çalışma yönteminde, aynı zaman diliminde birden fazla işlem yapılabilir. Web sunucusuna doldurulan bir form verileri gönderilirken, geçen zaman dilimi içinde başka işlemler de yapılabilir. Çünkü eşzamanlı çalışma yönteminde bloke edilen Javascript motoru, eşzamansız çalışma yönteminde serbesttir ve başka isteklerin gönderilmesine müsaade etmektedir. Şekil 4.6.'da görüldüğü gibi web tarayıcısından çıkan birden fazla istek aynı zaman dilimi içinde sunucuya iletilir ve cevaplar sunucudan alınır.



Şekil 4.6. Eşzamansız çalışma yöntemi

4.2.3. Ajax tekniğinin avantajları ve dezavantajları

Ajax eskiden beri var olan ancak varlığı çok sonra keşfedilen bir tekniktir. Bu tekniğin kullanımında çok fazla avantajı olduğu gibi dezavantajları da mevcuttur.

Ajax'ın avantajları: HTTP talepleri arka planda işlendiği için web sayfasının yeniden yüklenmesi gerekmemektedir.

Sayfaların yeniden yüklenmeden sadece gerekli veri alışverişi yapıldığından dolayı önemli miktarda bant genişliği tasarrufu sağlamaktadır.

Masaüstü uygulamalarındaki zengin dinamik etkileşimin, web uygulamalarında da gerçekleşmesini mümkün kılmaktadır. Resim 4.1.'de eyeos.com sitesinin web sitelerinde uyguladıkları örnek masaüstü uygulaması görülmektedir. Bu uygulamada kullanıcılar, yazı yazma, hesap

yapma, dosya yükleme, silme işlemleri, silinen dosyaları geri kurtarabilme, oyun oynama, sunum yapma gibi masaüstü uygulamalarının çoğunu kullanabilmektedirler.



Resim 4.1. Eyeos.com sitesinin masaüstü uygulaması

Ajax, birçok web tasarımcısının bildiği teknolojilerden oluştuğu için, geliştiricilerin bu yaklaşımı öğrenmesi ve bu kavrama uyum sağlamasını kolaylaştırmaktadır.

Java, Flash gibi her ortamda bir standardı olmayan, çalışabilmesi için ek programlara ihtiyaç duyan, web tarayıcılarını ve veri trafiği yoran uygulamaların aksine daha yüksek ve gerçek zamanlı etkileşim sunmakta, daha az kaynak tüketmekte ve uygulama zamanı olarak çok daha az zaman almaktadır.

Uygulama açısından sürekli sorunlu olan iframe ve popup pencerelerin bu tür uygulamalarda ortadan kalkmasına ve dolayısıyla azalmasına neden olmuştur.

Ajax, CSS, Dom, Xhtml benzeri teknolojilerin kullanımını gerektirdiği için bu teknolojilerin ve standartların gelişmesine de katkıda bulunmaktadır.

Güvenlik nedeniyle güvenilmeyen kullanıcı tabanlı JavaScript denetimleri yerine sunucu tabanlı denetimler hızlıca ve basitçe sayfa yenilenmeden ve böylece kullanıcıya zaman kaybı yaşatmadan yapılabilmektedir.

Web tarayıcısının ileri ve geri butonları Ajax uygulamalarında çalışmamaktadır. Bu durum bazen avantaj olarak değerlendirilirken, bazen de dezavantaj olarak değerlendirilebilir. Oturum kullanılan dinamik web uygulamalarında kullanıcının “geri” butonunu kullanması istenmemektedir. Bunu engellemek için ek komutlar, popup pencereler kullanılmaktadır. Ajax uygulamalarında ek komutlara ihtiyaç yoktur.

Ajax uygulamalarının içeriği arama motorları tarafından taranamaması bazı durumlarda dezavantaj bazı durumlarda avantaj olabilmektedir. İçeriği arama motorlarına karşı gizlemek için kullanılan robot.txt dosyası uygulamasına Ajaxta gerek kalmamaktadır.

Ajax dezavantajları: Yüksek miktarda bellek ve işlemci kullanımına neden olan tarayıcılar, Ajax uygulamalarında biraz daha fazla bellek ve işlemci gücü istemektedir. Sınırlı imkânları olan kullanıcılar bu durumlarda rahatsız olabilirler.

Tarayıcının ileri ve geri butonların Ajax uygulamalarında kullanılmamaktadır.

Uygulama içeriğinin dinamik ve eşzamanlı üretilmesi sebebiyle, Ajax uygulamaları arama motorları tarafından indekslenememektedir.

Klasik web uygulama sistemindeki gibi düzgün bir şekilde; sayfaların kaydedilmesi, çıktı alamama sorunları yaşanmaktadır.

Klasik web uygulamalarında kullanıcıdan veri toplamak tamamen kullanıcıya bağlı iken, Ajax web uygulamalarında kullanıcının haberi olmaksızın veri

toplama olanağı oluşmuştur. Web sayfası üzerindeki bir alanda bekleme süresi, fare imlecini en çok nerede gezdirdiğimiz gibi hareket, farkında olmadan elde edilebilir, kaydedilebilir ve kullanılabilir olmuştur.

Ajax'ın çalışabilmesi için Javascripte ihtiyaç duyar. Javascript uygulamalarını engelleyen kullanıcılar Ajax uygulamalarını da çalıştıramamaktadır.

4.2.4. Ajax'ın en iyi kullanım alanları

Avantaj ve dezavantajlarına göre Ajax tekniğinin kullanılabileceği en iyi web uygulamalarını aşağıdaki şekilde sıralanabilir:

Oylama, derecelendirme: Bazı sayfalarda kullanıcının, oy vermesi veya derecelendirme yapması gerekmektedir. Bu kısımlar sayfanın çok az bir bölümünü oluşturmaktadır. Ajax kullanılmayan sistemlerde, çok az bir alan için tüm sayfanın tekrar yüklenmesi gerekmektedir. Bu durum hem veri trafiğini artıracak gibi hem de işlem süresini uzatacaktır.

Önerilerde: Kullanıcının; aramak istediği ya da veri girişi yapmak istediği durumlarda, verinin ilk birkaç harfini yazdığı anda olası kelimelerin liste halinde kullanıcıya sunulması yöntemidir. Google Suggest uygulaması önerilerin en güzel örneğidir.

Web Images Video News Maps more »		Advanced Preferences Language
A		
amazon	11,510,000,000 results	
argos	16,500,000 results	
amazon.com	641,000,000 results	
aol	1,280,000,000 results	
ask jeeves	575,000,000 results	
aol.com	300,000,000 results	
apple	1,780,000,000 results	
ask.com	1,140,000,000 results	
ask	5,700,000,000 results	
autotrader	148,000,000 results	

Resim 4.2. Google arama motorunda Ajax tekniği

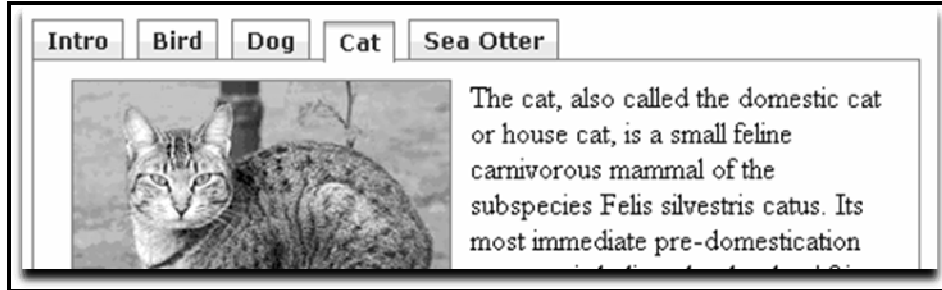
Form verilerinin kontrolü: Form girişlerinde girilen bilgilerin doğruluğunu aynı anda test etmek için Ajax tekniği kullanılabilir.



Resim 4.3. Form verisinin kontrolü

Otomatik Güncelleme: Önceden kaydı tutulmuş veriler üzerinde güncelleme yapılması gereken durumlarda Ajax tekniği kullanılabilir. Bu sayede verileri güncellemek için fazladan bir güncelleme formu oluşturmaya gerek duyulmaz.

Mikrolink Uygulamaları: Detayların tek bir tık ile değişebileceği durumlarda Ajax tekniği kullanılabilir. Mikrolink uygulamalarının en bilinen yöntemi sekmeli menülerdir. Her menüye tıklandığında içerik anında değişmektedir.



Resim 4.4. Mikrolink

Dinamik Form: Bir formda bir alanı içeriği değiştiğinde ona bağlı diğer alanların da içeriğinin değişmesi istenilen durumlarda kullanılabilir. Daha çok ülke – şehir ilişkilerinde sıklıkla karşımıza çıkmaktadır.

Aktif Arama: Klasik arama yönteminde aranılacak kelime metin kutusu içine yazılır ve ara butonuna basılır. Ajax yöntemi kullanılarak yapılan arama işlemlerinde yazdıkça ilgili arama sonuçlar ekranda belirmeye başlamaktadır.



Resim 4.5. Aktif arama

Veri Sunulması, Filtreleme: Özellikle tablolarda veri süzme işlemlerinde Ajax tekniğini kullanabiliriz. Filtrelemenin, yapılmış en güzel örneği amazon.com sitesinin elmas arama sayfasıdır.

Harita ve Büyük Resimleri Kaydırma: Ajax tekniğinin popüler olmasındaki en büyük etken olan Google Map bu uygulama için en iyi örnektir.

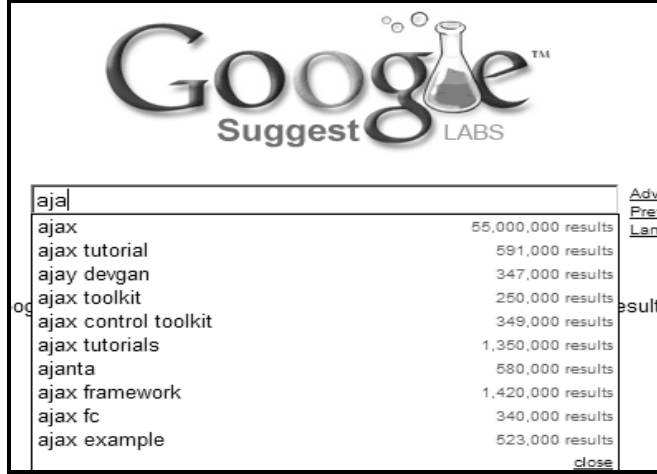
Sohbet(Chat) Uygulamaları: Sadece tanışmak, zaman geçirmek amacının dışında müşterilere kolay yoldan ulaşmak, satış öncesi fikirler vermek gibi nedenlerden dolayı da sohbet(chat) uygulamaları oldukça popülerdir. Sayfanın yeniden yüklenmemesi avantajıyla birlikte chat uygulamaları için Ajax çok pratik bir yöntemdir.

4.2.5. Ajax Kullanan İnternet Siteleri

Her geçen gün kullanım sayısı hızla artan Ajax tekniğini iyi ve doğru şekilde kullanan bazı web uygulamaları şunlardır:

Google Suggest: Klasik google arama motoruna benzer bir arayüzü vardır. Metin kutusu içine karakterleri yazmaya başlandığında ilgilenme ihtimali olan arama terimlerini bir açılır liste halinde gösterir. Bu uygulama sadece

masaüstü programlarda var iken artık web uygulamalarında da kullanılmaktadır. Ajax'ın popülaritesinin artmasındaki baş aktördür.



Resim 4.9. Google suggest uygulaması

Gmail, MsnLive, Yahoo: Ücretsiz mail servislerinin önde gelen firmaları da Ajax tekniğini kullanan ikinci bir uyarlamasını daha hazırlamışlardır. Kullanıcılar istedikleri uyarlamayı kullanabilmektedirler.

Google Maps: Ajax'ın en iyi kullanılan örneklerinden biridir. Web uygulamasında haritanın fare ile sürüklenmesi esnasında sayfanın tekrar yüklenmesini engellemek için Ajax kullanılmıştır.

A9: Dünyaca ünlü alışveriş sitesi amazon.com sitesinin arama motoru sitesidir. Özelliği farklı tipteki verileri eşzamanlı olarak aramaya olanak sağlaması ve sonuçları da farklı alanlarda gösterebilmesidir.

Netvibes: E- postaları takip etmek, arama motorlarında gezmek, video izlemek-aramak, blogları takip etmek, haber okumak gibi birçok işi web sitesini tek ekranda toplayan bir sistemdir.

Bunlarla birlikte meeboo, flickr, facebook gibi daha bir çok web sitesi sayılabilir.

4.3 XmlHttpRequest Nesnesi

XmlHttpRequest, Ajax tekniğinde önemli bir yer tutar, istemci ve sunucu arasındaki eşzamansız iletişimi sağlayan arabirim nesnesidir. Bu bölümde bu nesnenin oluşturulması kullanımı hakkında bilgi verilecektir.

Xhr nesnesi tüm tarayıcılar tarafından desteklenen bir nesnedir. Ancak tüm tarayıcılarda hatasız çalışabilmesi için bu nesnenin oluşturulma yolları farklıdır.

```
var xhr = new xmlhttprequest();
```

kod satırı çoğu tarayıcılarda yeni bir XHR nesnesi oluştururken, Microsoft IE için bu kod yeterli değildir.

```
var xhr = new ActiveXObject("Microsoft.XMLHTTP");
```

Var olan bir nesneyi tekrar oluşturmak hataya sebep olacaktır. Bu nedenle ilk yapılması gereken Xhr nesnesinin varlığının denetlenmesi gerekmektedir. Bu bölüme ait son kod örneği aşağıda verilmiştir. If denetim satırı ile nesne varlığı test edilmekte, try...catch deyimleri ile oluşabilecek hatalar yakalanmaktadır. Try...catch deyimleri kullanarak en yeni sürümü kullanma girişiminde bulunulmuş şayet hata oluşursa eski sürüm işletilmeye devam edilmiştir.

Aşağıdaki tüm tarayıcılarda sorunsuz çalışan Xhr nesnesi oluşturma kod örneğidir.

```
function xhryarat() {  
  
var xhr = false;  
  
if (window.XmlHttpRequest) {  
  
xhr = new XmlHttpRequest();
```

```
}else if (window.ActiveXObject) {  
  
    try {  
  
        xhr = new ActiveXObject("Msxml2.XMLHTTP");  
  
    } catch (e) {  
  
        try {  
  
            xhr = new ActiveXObject("Microsoft.XMLHTTP");  
  
        } catch (e) {  
  
            xhr = false;  
  
        }  
  
    }  
  
    }  
  
    }  
  
    }  
  
    return xhr;  
  
}
```

Bu fonksiyon her işlem öncesi Xhr nesnesi oluşturmak için çağırılacak ve her tarayıcı tarafından aynı nitelikte tanınacaktır.

Xhr oluşturulduktan sonra sunucu ile iletişime geçebilmek için hazırlanması gerekir. Bunun için önemli üç yöntem ve özellik vardır.

OnReadyStateChange: Bu özellik sunucunun web tarayıcısına bir güncelleme bilgisi gönderdiği anlamımıza yarayacaktır.

Xhr nesnesinin *readystatechange* özelliği sunucunun durumunu belirtir. İsteğin başlatılıp başlatılmadığını, yanıtlanıp yanıtlanmadığını ya da istek/yanıt

modelinin tamamlanıp tamamlanmadığını anlatmak için kullanılır. Sunucunun sağladığı yanıt metninin ya da verilerin okuma için güvenli olup olmadığını belirlemeye de yardımcı olur. Bu özellik beş değer alabilir:

- 0: İstek başlatılmadı.
- 1: İstek hazırlandı, ancak henüz gönderilmedi.
- 2: İstek gönderildi ve işleniyor.
- 3: Sunucu bir yanıt vermek üzeredir.
- 4: Sunucu yanıtı göndermiştir.

Readystate her değiştiğinde *onreadystatechange* olayı tetiklenmektedir. Ajax uygulamalarında *readystate = 4* olduğu durumlarda sonuçlar sayfada gösterilmektedir.

Open: Web tarayıcısı sunucudan bir istekte bulunmasını sağlar. Bu istek bir form ile gönderilmiş veriler olabileceği gibi çok büyük kapsamda verilerde içerebilir.

Genel kullanım şablonu şu şekildedir:

```
open("metod", "url", "eşzamanlı / eşzamansız", ["kullanici adi", "parola"])
```

İki türlü istekte bulunabilir. GET metodunu sunucudan sadece veri almak için tercih edilmektedir. Sunucuya veri göndermek istendiğinde ise POST methodu tercih edilmektedir.

Url parametresi çalıştırılması istenen www.eyceks.com/deneme.php gibi bir dosya yoludur.

Üçüncü parametre sunucu ile olan iletişimin eşzamanlı ya da eşzamansız olacağını belirtmek için kullanılmaktadır. Boolean tipinde olduğundan "true" ya da "false" değerini almaktadır. True değeri verildiğinde sunucu ile olan iletişimin "eşzamansız" olacağına yani sunucudan cevap gelmeden yeni bir istekte bulunulabileceğini belirtmektedir. False değeri ise eşzamanlı iletişim

gerçekleşeceğini belirtmektedir. Tarayıcı bir istekte bulunacak ve cevap gelene kadar dosyanın çalışması durdurulacaktır.

Son iki parametre kullanıcı adı ve şifre gönderilmesini sağlamaktadır. Ancak güvenlik açısından kullanılması doğru değildir.

Send: Open komutu ile başlatılan kullanıcı isteğini sunucuya göndermek için kullanılmaktadır. GET methodu ile istek yapılacağı zaman sadece tek bir komut yeterli olmaktadır.

```
nesneismi.send(null);
```

Ancak POST methodunda durum farklıdır. Gönderilen isteğin içeriği *setRequestHeader* komutuyla belirtilir.

```
nesneismi.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
```

```
nesneismi.send(var1=value1&var2=value2);
```

Sunucudan gelen cevap bilgisini içerir. Gelen bilgi html, xml ya da sadece metin olabilir.

```
nesneismi.responsetext;
```

4.3.1. Uygulamalar

Ajax mantığının kullanımına örnek olması açısından bu bölümde uygulamalara yer verilmiştir.

Web sitesi sahipleri, kullanıcılarının görüş ve isteklerini öğrenebilmek için geri bildirim formları kullanırlar. Kullanılmaması gibi bir durum söz konusu değildir. Aşağıda klasik yöntem ve Ajax teknikli geri bildirim formlarına örnek verilmiştir.

The image shows a web form with the following elements:

- A label "E-Mail :" followed by a single-line text input field.
- A label "Mesaj:" followed by a multi-line text area.
- Two buttons: "Temizle" (Clear) and "Gönder" (Send).

Şekil 4.7. Basit bir geri besleme form uygulaması

```
CREATE TABLE `geribesleme` (
  `id` int(11) NOT NULL auto_increment,
  `email` varchar(255) NOT NULL default "",
  `mesaj` mediumtext NOT NULL,
  `saat` datetime NOT NULL default '0000-00-00 00:00:00',
  PRIMARY KEY (`id`)
);
```

Kod parçasıyla veritabanında tablo oluşturulmaktadır. Bu kısımdan sonra iki adet sayfa oluşturulmaktadır. Şekil 4.7.'de gösterilen form için kullanılacak olan form.html dosyasının kodları aşağıdaki gibi olacaktır. Daha sonra bu dosya ile gönderilen verileri veritabanına işleyecek olan mesajgonder.php dosyası hazırlanacaktır.

```
<form name="form" id="form" method="POST" action="mesajgonder.php">
```

```
<label for="email">E-Mail</label>
```

```
<input name="email" id="email" type="text" />
```

```
<br /><br />
```

```
Mesaj<br />
```

```
<textarea name="mesaj" id="mesaj" rows="5" cols="30"></textarea>
```

```
<br />
```

```
<input name="sil" type="reset" id="sil" value="Temizle" />
```

```
<input name="gonder" id="gonder" type="submit" value="Gönder" />
```

```
</form>
```

```
<div id="sonuc"></div>
```

mesajgonder.php dosyamızın içeriği de aşağıdaki gibi olacaktır.

```
<?php
```

```
$baglan=mysql_connect("localhost","admin","password");
```

```
mysql_select_db("db",$baglan);
```

```
mysql_query("INSERT INTO geribesleme(email,mesaj,saat)
```

```
VALUES('$REQUEST[email]','$REQUEST[mesaj],NOW())" or die("hata  
olustu : " . mysql_error());
```

```
?>
```

```
<html> <head><title>GeriBesleme</title></head><body>
```

```
<h1>Mesajınız Kaydedildi</h1> Mesajınız için teşekkür ederiz. En kısa  
zamanda mesajınız yanıtlanacaktır.
```

```
</body>
```

```
</html>
```

Klasik yöntem ile oluşturduğumuz geri besleme formumuzda ilk önce *form.html* dosyamız çalışacak, bu formda gönder butonuna basıldığı anda sayfa yenilecek ve yerine *mesajgonder.php* dosyası çalışarak mesaj veritabanına kaydedilmiş olacaktır. Aynı form tasarımının Ajax tekniği ile yapılması aşağıdaki şekilde mümkündür.

```
<script type="text/javascript">

// İlk önce XHR nesnesi oluşturulur

function XMLHttpRequest() {

var xhr = false;

if (window.XmlHttpRequest) {

xhr = new XmlHttpRequest();

} else if (window.ActiveXObject) {

try {

xhr = new ActiveXObject("Msxml2.XMLHTTP");

} catch (e) {

try {

xhr = new ActiveXObject("Microsoft.XMLHTTP");

} catch (e) {

xhr = false;

}

}

}
```

```
}  
  
return xhr;  
  
}  
  
var Xhr = new XMLHttpRequest();  
  
// Bu fonksiyon formun gonder butonuna basıldığı anca çağırılacaktır  
  
function kaydet () {  
  
var mesaj = document.getElementById("mesaj").value;  
  
var email = document.getElementById("email").value;  
  
//open deyimi ile istek hazırlanmaktadır. True parametresi iletişimin  
//Eşzamansız olacağını göstermektedir.  
  
xhr.open("GET","mesajgonder.php?mesaj=" +mesaj + "&email=" + email +  
"&return=onay",true);  
  
xhr.onreadystatechange = function() {  
  
// eğer sunucudan yanıt gelmişse ve hata yoksa gelen veriyi alıyoruz  
  
if(xhr.readyState == 4) {  
  
    if(xhr.status == 200) {  
  
        var result = xhr.responseText;  
  
        var msg = ""  
  
        if(result == '1') {
```

```
        msg = "Mesajınız için teşekkür ederiz. En kısa zamanda  
mesajınız yanıtlanacaktır. ".result;
```

```
    } else {
```

```
        msg = "Hata oluştu : " . result;
```

```
    }
```

```
    document.getElementById("form").innerHTML = msg;
```

```
    }
```

```
    }
```

```
    }
```

```
    http.send(null);
```

```
    return false;
```

```
    }
```

```
</script>
```

mesajgonder.php dosyasında mysql_insert deyiminin hemen ardından aşağıdaki kod parçası eklenerek işlem tamamlanır.

```
if($_REQUEST['return'] == 'onay') {
```

```
    print 1;
```

```
    exit;
```

```
}
```

Aşağıdaki bir diğer uygulama örneğinde şehir ve ilçe alanları dolduruldukça posta kodunun otomatik olarak yazılmasını sağlamaktadır.

The image shows a web form with three input fields. The first field is labeled 'Şehir:' and has a width of 20 characters. The second field is labeled 'İlçe:' and also has a width of 20 characters. The third field is labeled 'Posta Kodu:' and has a width of 5 characters. The form is enclosed in a black border.

Şekil 4.8. Otomatik doldurma için form tasarımı

Şehir ve ilçe alanına her metin girişi olduğunda postakoduogren() fonksiyonu sunucuda çalıştırılacaktır.

```
<form>
```

```
<p>Şehir: <input type="text" name="sehir" id="sehir" size="20"
```

```
onChange="postakoduogren();" /></p>
```

```
<p>İlçe: <input type="text" name="ilce" id="ilce" size="20"
```

```
onChange="postakoduogren();" /></p>
```

```
<p>Posta Kodu: <input type="text" name="zipkodu" id="zipkodu" size="5" />
```

```
</p></form>
```

Oluşturulan XMLHttpRequest nesnesinden sonra sunucuya gönderilecek tüm verileri almak, sunucuya bağlanmak, sunucudan gelen bilgiler için ne yapılması gerektiğine karar vermek gerekmektedir. Aşağıdaki kod örneğinde bir formdan alınan şehir ve ilçe bilgilerini sunucuya gönderen bir istek fonksiyonu görülmektedir.

```
<script language="JavaScript">
```

```
// İlk önce XHR nesnesi oluşturulur
```



```
function XMLHTTPObject() {  
  
    var xhr = false;  
  
    if (window.XmlHttpRequest) {  
  
        xhr = new XmlHttpRequest();  
  
    }else if (window.ActiveXObject) {  
  
        try {  
  
            xhr = new ActiveXObject("Msxml2.XMLHTTP");  
  
        } catch (e) {  
  
            try {  
  
                xhr = new ActiveXObject("Microsoft.XMLHTTP");  
  
            } catch (e) {  
  
                xhr = false;  
  
            }  
  
        }  
  
    }  
  
    return xhr;  
  
}  
  
var Xhr = new XMLHTTPObject();  
  
function postakoduogren() {
```

```
// Formdan sehir ve semt bilgilerini aliyoruz

var sehir = document.getElementById("sehir").value;

var ilce = document.getElementById("ilce").value;

// sehir ve ilce alanları doluysa sunucuyla bağlantı kurulacaktır

if ((sehir == null) || (sehir == "")) return;

if ((ilce == null) || (ilce == "")) return;

// Bağlantı yapılacak url belirleniyor

var url = "/zipkodu.php?sehir=" + escape(sehir) + "&ilce=" + escape(ilce);

// sunucu ile bağlantı sağlanıyor. True ifadesi iletişimin eşzamansız olacağını gösteriyor.

xhr.open("GET", url, true);

// Sunucu hazır olduğunda hazır() isimli fonksiyonun çalışacağı belirtiliyor

xhr.onreadystatechange = hazır;

// istek gönderiliyor

xhr.send(null);

// Xhr nesnesinin durumu 4 olduğunda yanıt sunucudan gelmiş demektir.

function hazıriz() {

if (xhr.readyState == 4) {

var yanıt = xhr.responseText;
```

// Sunucudan alınan sonuç bilgisi form kodunda zipkodu ile belirtilen alana yazılıyor

```

    document.getElementById("zipkodu").value =yanit;

}
}

</script>

```

Yukarıdaki kodlarda da görüldüğü gibi *form.html* dosyamızda yer alan şehir ve ilçe alanlarının içeriği değiştiği anda postakodu verileri *zipkodu.php* dosyasından gelecektir. Bu dosyanın içeriğinde veritabanı ile bağlantı kurulmuş ve “*select..from*” mysql deyimi ile veriyi filtreleme komutları yer almalıdır.

```
var url="/zipkodu.php?şehir="+escape(sehir)+"&ilçe="+escape(ilce);
```

Örnek kod içinde yer alan `escape()` ifadesi düz metin olarak gönderilemeyen karakterler için kullanılır. Örnekteki ilçe isminde boşluk karakteri var ise bu karakter `%20` olarak ifade edilir ve adres satırına eklenir.

Yukarıda belirtilen hazır durumlarından sıklıkla 1 ve 4 durumları kullanılır. İşlem gerçekleştiği süre içerisinde ekranda “yükleniyor” ifadesi belirmesi istenirse, “1” durumunun kullanılması gerekmektedir.

Bu çalışmanın ekler kısmında test aşamasında kullanılan web sitesinin kodlarının bir bölümü mevcuttur. Bu kodlar içindeki Ajax uygulaması başka bir sayfa içeriğinin aktif sayfada gösterilmesine örnek teşkil etmektedir.

4.4. Ajax Kütüphaneleri

Ajax tekniđi daha önce belirtildiđi gibi javascript temeline dayanmaktadır. Geliřtiriciler uygulamalarda kullanılmak üzere fonksiyonlarını yazarak javascript kütüphane dosyalarını oluřturmaktadır.

Her geen gün Ajax için yazılan kütüphane sayısı hızla artmaktadır. Ancak rastgele bir kütüphane ile alıřmak dođru deđildir. Ajax tekniđinin avantajlarından yararlanmak için kütüphanelerin ilgili bölümleri uygulamalarda kullanılmalı, ihtiyaç duyulmayan kısımları kaldırılmalıdır. Bu bölümde en ok kullanılan ve tercih edilen Ajax kütüphanelerinden örnekler sunulmuřtur.

Adobe Spry Framework for Ajax: Adobe firması tarafından Ajax için geliřtirilmiř olan Spry framework, web tasarımcıları için, kullanıcılarına daha zengin bir deneyim yařayacakları ortamlar hazırlamalarına yardımcı olabilecek ve pek fonksiyonel bir javascript kütüphanesidir. web adresi: <http://labs.adobe.com/technologies/spry/>

Prototype: Prototype, dinamik web sitesi geliřtirmeyi kolaylařtırmak amacıyla oluřturulmuř bir JavaScript framework'tür. web adresi: prototype.conio.net

Rico : Rico Ajax talep iřleyicilerini (Ajax request handler) ve de HTML elementlerini ya da JavaScript nesnelerini Ajax cevap nesneleri (Ajax response object) olarak register etmek için kullanılacak ok basit bir arayüz sađlıyor. oklu elementler ve/ya da nesnelere bir Ajax talebinin sonucu olarak güncellenebilirler. web adresi: <http://openrico.org/rico/home.page>

MochiKit: MochiKit, ok iyi dokümanede edilmiř ve test edilmiř, pek ok řeyi hızlı bir řekilde yapmanıza olanak sađlayacak JavaScript kütüphanesi. web adresi: <http://www.mochikit.com/>

Dojo Toolkit Dojo, JavaScript'i destekleyen, dinamik web siteleri ya da başka ortamlar hazırlamanızı kolaylaştıran açık kaynak kodlu bir toolkittir. web adresi: <http://www.dojotoolkit.org/>

Moo.FX: Moo.FX, prototype.js ile yazılmış oldukça küçük bir JavaScript efekt kütüphanesidir. web adresi: <http://moofx.mad4milk.net/>

Scriptaculous: prototype.js ile kodlanmış, Javascript görsel efektlerinden oluşan bir kütüphanedir. Çok popülerdir. web adresi: <http://script.aculo.us/>

Sajax: Sajax programlama web sitesi yapmak için kullanılacak bir framework'tür. Bütün sunucu taraflı platformlara çağrı gönderebilme yeteneğine sahiptir: ASP/ ColdFusion/ Io/ Lua/ Perl/ PHP/ Python/ Ruby. web adresi: <http://www.modernmethod.com/sajax/> [15].

5. PERFORMANS TESTLERİ VE DEĞERLENDİRMESİ

Bu bölüme kadar Ajax tekniğinin gereksinimleri, kullanım alanları, nasıl kullanıldığı, avantaj ve dezavantajlarından bahsedilmiştir. Çalışmanın bu kısmında, giriş bölümünde belirtilen Ajax tekniği zaman ve performans açısından tasarruf sağlar tezinin ispatı üzerinde durulmuştur. Veri trafiğini rahatlatmasının ne ölçüde olduğu test edilmiştir.

5.1. Web Sitesi Performans Testi

Bu aşamada Ajax'lı ve klasik yolla hazırlanmış web sitesinin performansı zaman ve bant genişliği açısından ölçülmüştür. Sitenin iki ayrı uyarlamasında da aynı iş yükü yüklenmiştir.

Test ortamı için aşağıdaki donanım ve yazılımlar kullanılmıştır:

Yazılımlar:

- Web sitelerinin performansını ölçmek için tasarlanmış Wapt ve WebPerformance Suite yazılımları
- Ajax tekniği ve klasik yöntem kullanılarak hazırlanmış aynı içerikteki iki versiyonlu web sitesi
- WindowsXP + SP2 işletim sistemi

Donanımlar:

- 1024Kb Dsl internet hattı
- AMD 2000 işlemcili 768 mb bellekli bilgisayar
- Tek kullanıcı USRobotics Adsl Modem

Öncelikle Resim 5.1. de bir bölümü görülen, bir pansiyonun tanıtımı için iki farklı teknikle web sitesi hazırlanmıştır. Aynı içerikli bu iki versiyon hem klasik yöntem ile hem de Ajax tekniği ile ayrı ayrı çalıştırılacak sonuçlar karşılaştırılacaktır. İki farklı versiyon da internet ortamına taşınmıştır.



Resim 5.1. Test sitesi

Wapt ve WebPerformance Suite performans yazılımları yardımıyla iki farklı teknikle çalışmakta olan sitelerde aynı işlemler gerçekleştirilecektir.

İşlem basamakları şu şekilde belirlenmiştir:

- Tüm bağlantıların çalıştırılması
- Galeri sayfasından dört adet resmin tıklanarak gerçek boyutunun gösterilmesi
- İletişim sayfasından aşağıdaki bilgilerle mail gönderilmesi

İsim: Ajax

Email : Ajax@Ajax.com

Telefon: 555 55 55

Mesajınız: Rezervasyon yaptırmak istiyorum.

Geleceğiniz Tarih: 25 Haziran

Kaç kişi geleceksiniz: 5 Kaç gün kalacaksınız: 10

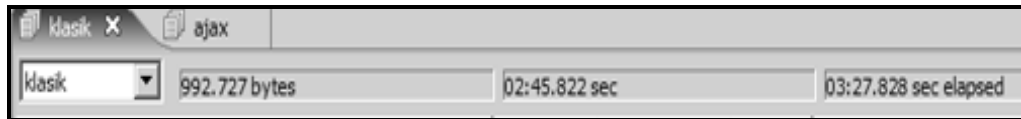
Bu test için kullanılan webperformance suite programı site üzerinde yapılan her işlemi kayıt altına alır. Yapılacak işlem basamakları bu sayede yazılım tarafından algılanmış olur. Daha sonra yazılım sanal ziyaretçiler oluşturarak, daha önce kayıtlı işlem basamaklarını sırasıyla uygulamaya başlar.

Performans test yazılımıyla elde edilen ilk test sonuçları aşağıdaki gibi çıkmıştır:

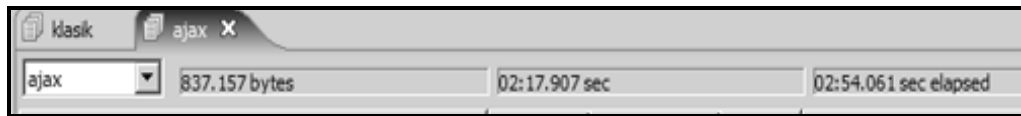
Çizelge 5.1. İlk test sonuçları

	Klasik Teknik	Ajax Teknik
Bant genişliği (byte)	1 345 535	1 164 324
İşlem süresi (dk)	01:38:711	01:25:993

Birinci aşama test bittikten sonra çok kullanıcıli ikinci bir test aynı maddelerle tekrarlanmıştır. Farklı olarak bağlantı hızı 56K olarak seçilmiş ve galeriden sadece bir resim orijinal boyutlarında görüntülenmiştir. Bu test uygulamasının ardından elde edilen sonuçlar Resim 5.2. ve Resim 5.3.'te gösterilmiştir.



Resim 5.2. Klasik tekniğin performans test sonucu



Resim 5.3. Ajax tekniğinin performans test sonucu

Hazırlanan uygulama üzerinde yapılan bir çok test sonucunda elde edilen verilerin ortalamaları aşağıdaki çizelgede belirtilmiştir.

Çizelge 5.2. Test sonuçları ortalaması

	Klasik Teknik	Ajax Teknik
Bant genişliği (byte)	992 727	837 157
İşlem süresi (dk)	03:27:828	02:54:061

Yapılan her denemenin sonunda, Ajax tekniğinden elde edilen sonuçların klasik yöntemle göre daha az çıktığı gözlenmiştir.

5.2. E-Posta Servislerinin Performans Testleri

Gmail'in Ajax tekniği ile kullanıcılara sunulmasının ardından rakip şirketler Microsoft ve Yahoo'nun bu atağa karşılık vermeleri gerekirdi. İki şirkette Ajaxlı yeni uyarlamasıyla, kullanıcılarının karşısına çıkmıştır. Bu bölümde her iki mail uygulamasının performans testleri aynı iş yükü ile ayrı ayrı test edilmiştir.

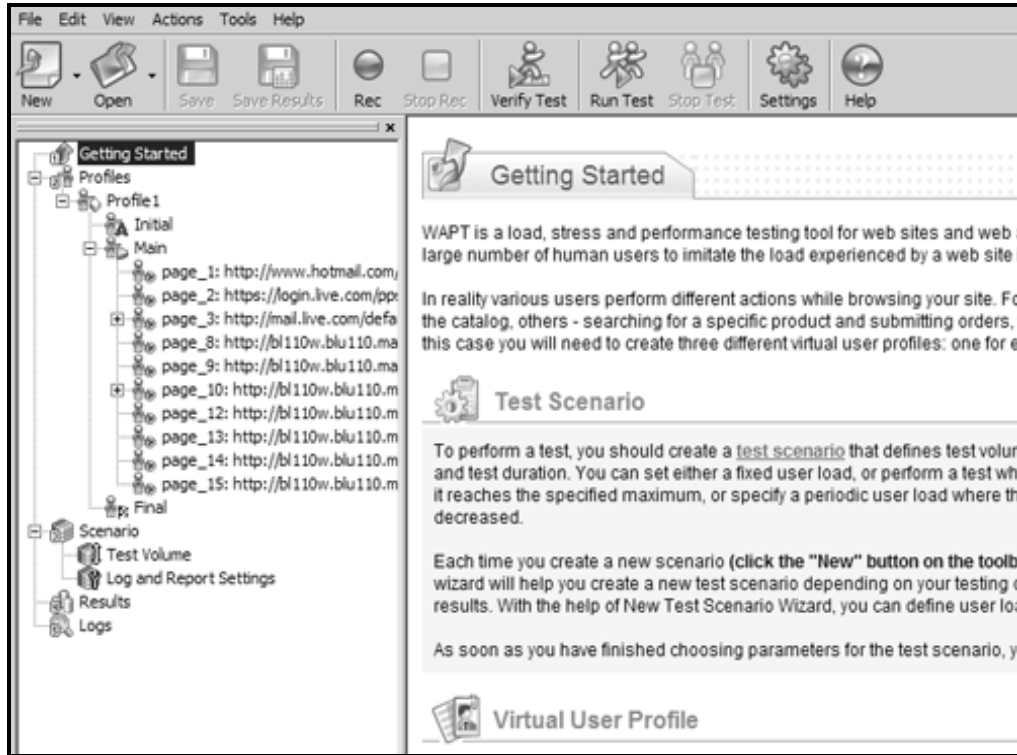
Bölüm 5.1.'de bahsedilen test ortamı aynen korunmuştur. Aynı koşullarda test tekrar edilmiştir.

Test süresinde her yapılan işlem yazılım tarafından kaydedilmektedir. Bu nedenle hatayı en aza indirmek aynı işlemleri her iki versiyonda yapılmak zorunluluğu vardır. Yapılacak işlemler şu şekilde belirlenmiştir:

- Gelen bir e-posta hiç açılmadan silinir,
- Gelen bir e-posta açılarak okunur,
- Yeni bir e-posta oluşturulur, başka bir kullanıcıya postalanır.

Okunacak , silinecek ve gönderilecek e-postalar aynıdır. Ajax'lı uyarlaması test edilirken silinen bir e-posta test sonunda tekrar gelen e-posta bölümüne taşınmış, klasik versiyon test edilirken de aynı e-posta tekrar silinmiştir. Test aşamasında farklı işlemler yapılması söz konusu değildir.

Wapt, SoftLogica firmasının geliřtirmiş olduđu bir web sitesi performans test yazılımıdır. Programın sanal ziyaretçiler oluşturarak, istenilen sıklıkta, istenen nitelikte web sitelerini test etme özelliđi bulunmaktadır. Ziyaretçi başına gönderilen, alınan veri miktarları, toplam kullanılan bant genişliđi, sayfaların yüklenme süreleri gibi ayrıntılı sonuçlar vermektedir. Ayrıca güvenli giriş SSL içeren sitelerinde test edilmesine olanak sağlamaktadır.



Resim 5.4. Wapt performans test yazılımının ara yüzü

1024Kb Dsl bağlantı kullanılmasına rağmen, test aşamasında hız limiti 56Kb olarak belirlenmiştir. Bu sayede hatlarda meydana gelebilecek olası zaman kaybını en aza indirilmiştir. Test süresi bir dakika olarak ayarlanmış ve yirmi sanal ziyaretçi ile web sitesine bağlantı kurulması sağlanmıştır.

İlk yapılan test sonuçlarında aşağıdaki veriler elde edilmiştir.

Çizelge 5.3. Live mail ilk test sonucu

	Klasik Teknik	Ajax Teknik
Giden Veri (Kb)	672	266
Alınan Veri (Kb)	1,548	494

İlk test sonuçlarına göre hotmail, Ajax tekniğini iyi kullanmış ve klasik yöntemle göre avantaj elde etmiştir.

Diğer test aşamasında farklı boyuttaki bir e-posta ile işlem yapılmış iki versiyonda da test edilmiştir.

Çizelge 5.4. Live mail test sonucu

	Klasik Teknik	Ajax Teknik
Giden Veri (Kb)	746	642
Alınan Veri (Kb)	2,162	1,520

Aynı test uygulamaları yahoo mail için uygulanmış test sonuçları aşağıdaki gibi olmuştur:

Çizelge 5.5. Yahoo mail test sonuçları

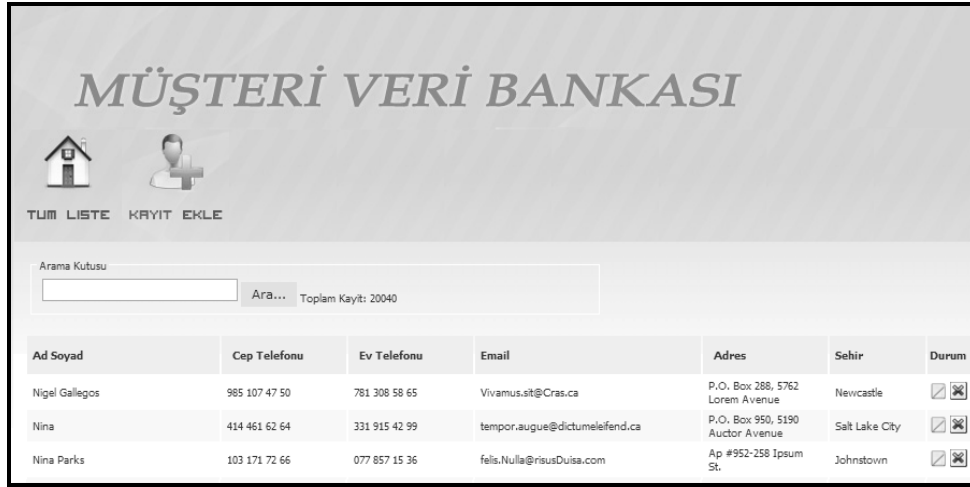
	Klasik Teknik	Ajax Teknik
Giden Veri (Kb)	308	273
Alınan Veri (Kb)	4,823	4,247

Bu test denemeleri birçok kez uygulanmış her seferinde Ajax teknik verileri klasik verilerden daha iyi çıktığı görülmüştür.

Burada test edilen e-posta servislerinin hangi Ajax kütüphanesi ile çalıştıkları bilinmemektedir. Kullanılan kütüphanelere göre sonuçlarda değişim olmaktadır. Bu nedenle Ajax tekniği ile elde edilmiş veriler ile klasik teknikle elde edilmiş veriler arasındaki fark değişebilmektedir. Ancak Ajax tekniğinin veri trafiğini ve çalışma süresini düşürerek klasik teknik ile aynı işi yapabildiğini söylemek mümkündür.

5.3. Veritabanı Uygulamalarında Ajax Performans Testi

Ajax tekniğinin web siteleri üzerindeki bant genişliği ve zaman tasarrufu etkisinin yanında web tabanlı veritabanı uygulamalarındaki performansını da ölçmek için müşteri veri bankası adında bir uygulama hazırlanmıştır. Aşağıdaki resimde bu uygulamanın ara yüzü görülmektedir.



Ad Soyad	Cep Telefonu	Ev Telefonu	Email	Adres	Sehir	Durum
Nigel Gallegos	985 107 47 50	781 308 58 65	Vivamus.st@Cras.ca	P.O. Box 288, 5762 Lorem Avenue	Newcastle	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Nina	414 461 62 64	331 915 42 99	tempor.augue@dictumelefend.ca	P.O. Box 950, 5190 Auctor Avenue	Salt Lake City	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Nina Parks	103 171 72 66	077 857 15 36	felis.Nulla@risusDuisa.com	Ap #952-258 Ipsum St.	Johnstown	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

Resim 5.5. Veritabanı uygulaması ara yüzü

Performans testi için geliştirilen veritabanı uygulama programı ile kayıt ekleme, silme, arama ve düzenleme işlemleri yapılabilmektedir. Uygulama programında, gerçek dünyaya uygun olması bakımından başlangıçta 20,000 kayıtlı veritabanı kullanılmıştır. Daha sonra sırasıyla veritabanının kayıtları 30,000 ve 40,000 olarak artırılmıştır. Performans testi, uygulamanın Ajax ve klasik yöntemle yapılmış iki farklı uyarlaması üzerinde ayrı ayrı uygulanmıştır.

Kayıt sayısı arttıkça Ajax tekniğinin sunucu üzerinde göstereceği performans test edilmiştir.

Uygulama programlarında kullanılan dosya boyutları aşağıdaki tablodaki gibidir.

Çizelge 5.6. Veritabanı uygulamasının dosya boyutu

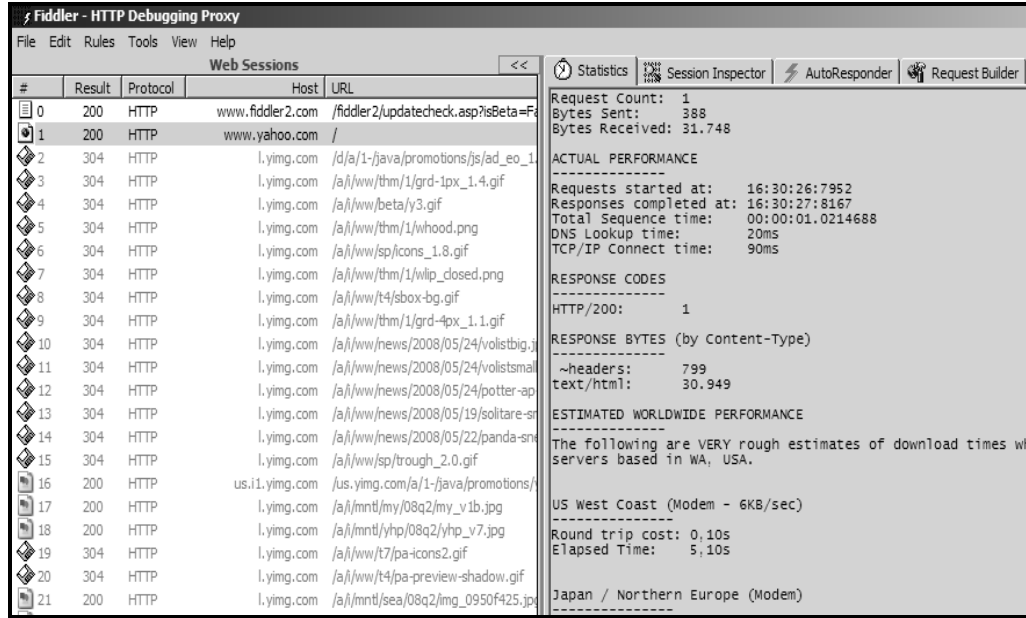
	Ajax Uygulama (Kb)	Klasik Uygulama (Kb)
Grafik Dosyaları	67,11	67,11
Kod Dosyaları	20,90	32,20
CSS Dosyaları	3,84	1,63
Javascript Dosyaları	34,50	-
Toplam	126,34	100,93

Ajax sunucu ile iletişim için javascript dosyalarına ihtiyaç duyar. Bu dosyalar sunucu ile iletişimi sağlamak ve veriyi arka planda sunucuya gönderip gelen sonucu istemciye iletmek için kullanır. Genel toplamda Ajax uygulamasının dosya boyutu, klasik uygulamaya göre daha çok yer tutmaktadır.

Ajax'ın veritabanı uygulamasındaki performans testi için geliştirilen müşteri veri bankası isimli uygulama sitesinde, 30 kişilik denek grubundan aşağıda belirtilen işlemleri yapmaları istenmiştir.

- Veritabanına üç farklı kişiye ait bilgileri girme
- Kayıtlarda arama yapma
- Arama sonucu bulunan kayıtlardan birini düzenleme
- Kayıtlardan birini silme

Performans testi sırasında istemci – sunucu arasındaki veri akışını takip edebilmek için Microsoft destekli Fiddler Web Debugger adlı yazılım kullanılmıştır.



Resim 5.6. Fiddler Web Debugger arayüzü

Bu yazılım test için kullanılan her bir bilgisayara yüklenerek, istemcinin sunucu ile iletişimi sırasındaki veri akışı ayrıntısıyla takip edilmiştir.

Test; senaryoda belirtilen işlemler esas alınarak 30 kişilik bir denek grubu ile aynı anda gerçekleştirilmiştir. Elde edilen sonuçların ortalamaları alınmış ve sonucu ulaşılmıştır.

Aşağıdaki çizelgede 20,000 kayıtlı veritabanı üzerinde ilk beş kişiye ait sunucuya gönderilen ve alınan verilerin byte olarak sonuçları görülmektedir.

Çizelge 5.7. Beş deneye ait performans testi sonuçları

Kişi Sayısı	Ajax Uygulama			Klasik Uygulama		
	Giden Veri (byte)	Gelen Veri (byte)	Süre (dk)	Giden Veri (byte)	Gelen Veri (byte)	Süre (dk)
1	12,494	378,049	02:25	11,307	650,637	03:03
2	14,682	373,449	02:30	12,745	648,364	03:10
3	13,352	392,903	02:14	9,800	651,367	02:58
4	12,407	264,613	02:28	9,906	633,038	03:01
5	10,335	315,056	02:33	10,565	580,306	02:57
Ortalama	12,654	344,814	02:26	10,864	672,742	03:01

Otuz kişilik denek grubu belirlenen görevlere bağlı kalarak 20,000 kayıtlı veritabanı uygulaması üzerinde testleri yapmış ve sonuçlar toplanmıştır. Sonuçların ortalamaları çizelgede gösterilmiştir.

Çizelge 5.8. Yirmibin kayıtlı veritabanı uygulamasında performans testi sonucu

Kayıt Sayısı	Ajax Uygulama			Klasik Uygulama		
	Giden Veri (byte)	Gelen Veri (byte)	Süre (dk)	Giden Veri (byte)	Gelen Veri (byte)	Süre (dk)
20,000	12,650	351,898	02:31	9,974	614,878	03:12

Belirlenen görevlere bağlı kalmak koşuluyla veritabanındaki kayıt sayısı 30,000'e çıkarılmış ve deneklerden elde edilen sonuçların ortalamaları alınmıştır.

Çizelge 5.9. Otuzbin kayıtlı veritabanı uygulamasında performans testi sonucu

Kayıt Sayısı	Ajax Uygulama			Klasik Uygulama		
	Giden Veri (byte)	Gelen Veri (byte)	Süre (dk)	Giden Veri (byte)	Gelen Veri (byte)	Süre (dk)
30,000	12,219	428,927	02:51	10,135	837,039	03:50

Belirlenen görevlere bağlı kalmak koşuluyla veritabanındaki kayıt sayısı 40,000'e çıkarılmış ve deneklerden elde edilen sonuçların ortalamaları alınmıştır.

Çizelge 5.10. Kırkbin kayıtlı veritabanı uygulamasında performans testi sonucu

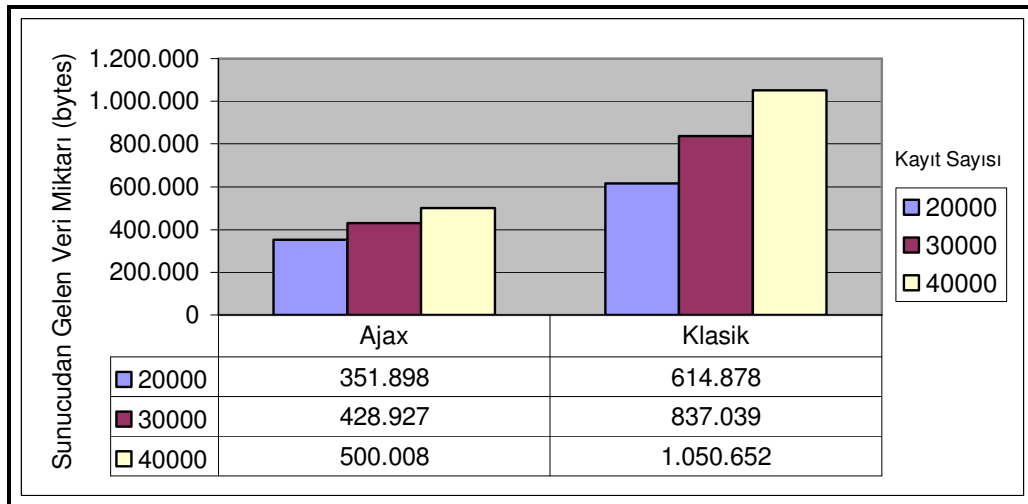
Kayıt Sayısı	Ajax Uygulama			Klasik Uygulama		
	Giden Veri (byte)	Gelen Veri (byte)	Süre (dk)	Giden Veri (byte)	Gelen Veri (byte)	Süre (dk)
40,000	12,428	500,008	03:13	10,542	1,050,652	04:55

Tüm bu sonuçlara bakıldığında Ajax'ın yapısı gereği kullandığı javascript dosyaları nedeniyle sunucuya gönderdiği veri miktarının biraz daha fazla olduğu görülmektedir. Ancak sunucudan alınan veri miktarına bakıldığında bu dezavantajını fazlasıyla telafi ettiği sonucuna varılmaktadır. Ayrıca Ajax'ın görsellik açısından kullanışlılığı da bunun yanında önemli bir özelliktir.

Veritabanındaki kayıt sayısı arttıkça her iki teknikte de sunucudan gelen veri miktarında ve işlem süresinde artış gözlenmektedir. Testlerde uygulanan işlem basamakları aynı olduğundan, sunucuya gönderilen veri miktarında önemli bir değişim olmamıştır.

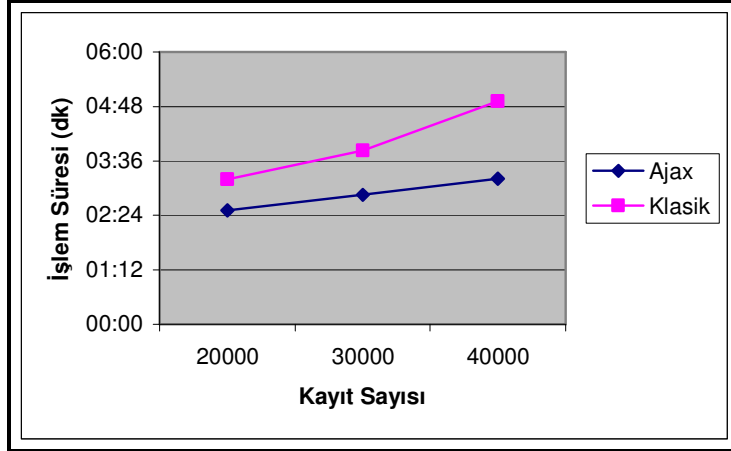
Klasik yöntemle yapılan uygulamada, sunucudan alınan veri miktarındaki artış, veritabanındaki kayıt sayısı arttıkça daha da artmaktadır.

Ajax yönteminde de bu artış gözlenmekte ancak klasik yöntemdeki artış miktarına göre gerilerde kalmaktadır. Şekil 5.1.'deki grafikte sunucudan gelen veri miktarlarının, kayıt sayılarına göre ortalamaları gösterilmektedir. Buna göre yaklaşık % 51,18 oranında bir değişim görülmektedir.



Şekil 5.1. Kayıt sayısına göre sunucudan gelen veri miktarlarının ortalamaları

Şekil 5.2.'de gerçekleştirilen işlem basamaklarına göre elde edilen işlem süresine ait grafik gösterilmektedir. Bu işlemler sonunda işlem süresinde yaklaşık % 30'luk bir zaman kısalması görülmüştür.



Şekil 5.2. Test sonucunda elde edilen ortalama işlem süreleri

Ajax tekniği, sunucudan daha az veri alarak klasik yöntem ile aynı işi yapmaktadır. Elde edilen yaklaşık %51,18'lik tasarruf, büyük ölçekli web uygulamaları için çok önemli sayılabilecek bir değerdir. Web sitelerinin kullandıkları bant genişliğine göre host şirketlerine ödeme yaptıkları düşünüldüğünde, önemli derecede tasarruf imkanı da sağlamaktadır. Ancak Ajax tekniğinin yerinde ve doğru kullanıldığında bu tasarrufu sağladığını da unutmamak gerekir.

6. SONUÇ VE ÖNERİLER

İnternet kullanıcıların gelişen teknoloji ile birlikte artan taleplerini karşılamak için geliştirilen ve geliştirilmekte olan web 2.0 döneminin önemli bir özelliği de Ajax tekniğidir.

Bu çalışmada; yapılan literatür taramaları ve performans testleri sonucunda Ajax masaüstü uygulamalarını internet ortamına sunmada en kolay ve masrafsız bir yol olduğu görülmüştür. Literatüre göre Google, Yahoo, Microsoft gibi internet devlerinin çalışmalarında bu tekniğe yer vermeleri ve devamlı kullanmaları ile ilerleyen zamanlarda Ajax'ın daha da gelişeceğini işaretleridir.

Google Docs, eyeos.com gibi internet uygulamalarında da görüldüğü gibi masaüstü uygulamalarında yapılan bir çok işlemin artık web ortamına taşınabildiği görülmüştür.

Klasik web anlayışımız olan; bir bağlantıya tıklandıktan sonra sayfanın tamamen ortadan kaybolup, yeni sayfanın yüklenmesi için geçen süre olarak belirtilen tıkla- bekle süreci Ajax tekniğinde ortadan kalkmıştır.

Ajax tekniğinin sunucu ile eşzamansız iletişim yeteneği sayesinde aynı zaman dilimi birden çok işlem yapılabilirdiği görülmüştür.

Ajax, sağladığı görsel özelliği bakımından internet kullanıcılarına, bant genişliği kullanımındaki tasarrufu ve istemci sunucu arasındaki işlem süresini düşürmesiyle de web sitesi sahiplerine yararı vardır.

Sayfa içeriğinin, yinelenmeden değişebilmesi, arama yaparken önerilerde bulunabilmesi, form alanlarının doldurulmasında yapılan hataların anında bildirilmesi, büyük ölçekli resimleri, haritaları izlemekteki kolaylık, sohbet uygulamaları gibi görsel zenginlikleriyle Ajax kullanıcı dostu olmuştur.

Bu çalışma sonucunda; Ajax tekniğini kullanan web sayfalarında istemci sunucu arasındaki veri trafiğinin azalması ve işlem süresinin düşmesiyle de sunucu performansında %51,18'e varan artış sağlanmıştır. Bu artış internet şirketleri için göz ardı edilemez.

Çalışma sonucu elde edilen bu oran; Clinton W. Smullen III ve Stephanie A. Smullen'in Mart 2008'de yayımladıkları makalelerinde belirttikleri sonuç ile paralellik sağladığı görülmüştür. Makalede elde edilen tasarruf oranı yaklaşık %54 olarak belirtilmiştir.

Yapılan performans testleri sonucuna göre, Ajax tekniği hem görsellik açısından hem de sunucu performansı ve ekonomisi açısından kullanımı faydalıdır.

Çalışma boyunca yapılan araştırmalara göre, bugün etkileşimli uygulamalar oluşturmada en güvenilir, en hızlı, en performanslı desteği Ajax'lı asp.net vermiştir. Microsoft'tan elde edilebilen kullanılabilir bedava, ticari geliştirme araçlarıyla iyi görünümlü ve performanslı web siteleri oluşturmak kolaydır.

.Net Ajax üzerine yoğun şekilde çalışmakta ve bu dalda zengin kütüphaneler sunmaktadır. Ajax üzerine çalışma yapmak isteyenler programcıların öncelikle javascript dili ile asp.net programlamaya önem vermeleri daha uygun olacaktır.

Web uygulamalarında Ajax tekniğine yer vermek isteyen programcılar, kullanacakları kütüphanelerin seçiminde dikkatli olmaları gerekmektedir. Basit bir uygulama için yoğun içerikli bir kütüphanenin kullanılması uygulamayı gereğinden fazla yoracaktır. Bu da Ajax'ın sağladığı hiçbir tasarruftan yararlanılamaması anlamına gelir.

Ajax tekniği Java ve Flash ile kıyaslanmaktadır. Java ve Flash'ın çalışabilmesi için ek yazılımlara ihtiyaç duyması nedeniyle Ajax bu konuda bir adım daha öndedir. Bu çalışma da Ajax ile Flash ve Java karşılaştırılması yapılmamıştır. Ajax'ın video, ses desteği olmaması nedeniyle tüm Flash ve

Java uygulamalarını Ajax ile yapmak mümkün değildir. Ancak birçok uygulama Ajax ile yapılabilir. Sonraki çalışmalarda Ajax ile bazı Flash ve Java uygulamaları geliştirilerek elde edilecek sonuçlar karşılaştırılarak sunucu performansları test edilebilir.

KAYNAKLAR

1. Teague, J.C., "Css, Dhtml & Ajax", **Peachpit Press**, Berkeley, 428-429 (2006).
2. Paulson, L. D., "Building Rich Web Applications with Ajax", **IEEE Computer Society**, 38(10):14-17 (2007).
3. Holzner, S., "Ajax For Dummies", **Wiley Publishing Inc.**, Indiana, 1-3 (2006).
4. Zakas, N. C., Mcpeak J., Fawcett J., "Professional Ajax 2nd Edition", Özveri, G., Al, K., **Alfa Yayınları**, İstanbul, 2-14 (2007).
5. Zepeda, J. S., Chapa, S. V., "From Desktop Applications Towards Ajax Web Applications", **4th International Conference on Electrical and Electronics Engineering (ICEEE 2007)**, Mexico City, 193 (2007).
6. Smullen, C. W., Smullen, S. A., "An Experimental Study of AJAX Application Performance", **Journal of software**, 3(3): 30 – 37 (2008).
7. Keith, J., "BulletProof Ajax", Kaya, M., **Açık Akademi**, İstanbul, 9-73, 161-202 (2007).
8. Darie, C., Brinzarea, B., Cherecheş, F., Bucica, M., "Building Responsive Web Application Ajax and Php", **Packt Publishing**, Birmingham, 16-18 (2006).
9. Johnson, D., White, A., Charland, A. "Enterprise Ajax Strategies for Building High Performance Web Applications", **Prentice Hall**, Massachusest, 22 (2007).
10. Sahillioğlu, E., "Web Uygulamalarında Yeni Yaklaşım: Ajax", **Akademik Bilişim Seminerleri**, Denizli, 10-12 (2006).
11. Jacobs, S., "Beginning Xml with DOM and Ajax ", **APress**, New York, 1-3 (2006).
12. Taşkın, G., Kabakçı, Ş., "Elektronik Ticaret", **Yeni Çizgi Yayınları**, Ankara, 70-103 (2003).
13. Pekgöz, N., "Webmaster için Javascript", **Pusulula Yayınları**, İstanbul, 83-85 (2001).
14. Crane, D., Pascerello, E., James, D., "Ajax in Action", **Manning Press**, Greenwich, 33-36 (2006).

15. Eernisse, M., "Build Your Own Ajax Web Applications", **Sitepoint Press**, Amerika Birleşik Devletleri, 283-285 (2006).

EKLER

Ajax test aşamasında yapılan sitede Ajax.js adında bir javascript dosyası oluşturuldu. Bu javascript dosyası XHR nesnesi oluşturup, sunucu ile arka planda bağlantıyı sağlayacak Ajax motorunu kontrol etmektedir.

Sayfalarda içeriğin görüntülenebilmesi için `<div id="icerik">....</div>` etiketi kullanılmaktadır.

index sayfası içerisinde `<script type="text/javascript" src="Ajax.js"> </script>` ifadesiyle Ajax dosyamız sayfaya dahil edilmektedir. Aşağıdaki javascript dosyasında da görüldüğü gibi bağlantılar,

`javascript:Sayfa('galeri.html');` şeklinde çağırılmaktadır.

Ajax.js

```
var http;
```

```
var browser = navigator.appName;
```

```
getNewXMLHttpRequest()
```

```
function getNewXMLHttpRequest() {
```

```
    var obj;
```

```
    try {
```

```
        http = new ActiveXObject('Microsoft.XMLHTTP');
```

```
    }
```

```
    catch(e) {
```

```
        try {
```

```
            http = new XMLHttpRequest();
```

```
        }
```

```
        catch (e) {
```

```
            http = false;
```



```

    } } return http; }

function Sayfa(adres){
    _icerik=document.getElementById("icerik");
    http.open('get', adres);
    http.onreadystatechange = function(){
        if(http.readyState == 4){
            _icerik.innerHTML = http.responseText;
        }else if(http.readyState == 0){
            _icerik.innerHTML = "Bir sorun oluřtur.";
        } }
    http.send(null);
}

```

Ajax'lı form kısmında ise gönder butonuna onclick="submitForm()" komutu eklenerek ařağıdaki javascript dosyası çağırılarak çalıştırılmaktadır.

form.js

```

function submitForm()
{
    var content = convertFormDataToPostContent(window.document.form);
    doPost('gonder.php', content, 'processResult');
}

function processResult(result)
{
    document.getElementById('icerik').innerHTML = result;
}

```

```
function doPost(url, content, callback_name)
{
    var async_request = false;
    if (window.XMLHttpRequest)
    {
        async_request = new XMLHttpRequest();
    }
    else if (window.ActiveXObject)
    {
        async_request = new ActiveXObject("Microsoft.XMLHTTP");
    }
    async_request.open('POST', url, true);
    async_request.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
    async_request.onreadystatechange = function()
    {
        if (async_request.readyState == 4)
        {
            response_content = async_request.responseText;
            eval(callback_name + '(response_content);');
        }
    }
    async_request.send(content);
}
```

```
function convertFormDataToPostContent(form_name)
{
    var content_to_submit = "";
    var form_element;
    var last_element_name = "";
    for (i = 0; i < form_name.elements.length; i++)
    {
        form_element = form_name.elements[i];
        switch (form_element.type)
        {
            case 'text':
            case 'hidden':
            case 'password':
            case 'textarea':
            case 'select-one':
                content_to_submit += form_element.name + '=' +
                escape(form_element.value) + '&'
                break;
            case 'radio':
                if (form_element.checked)
                {
                    content_to_submit += form_element.name + '='
                    + escape(form_element.value) + '&'
                }
        }
    }
}
```

```
break;

case 'checkbox':

if (form_element.checked)

{

    if (form_element.name == last_element_name)

    {

        if (content_to_submit.lastIndexOf('&') == content_to_submit.length - 1)

        { content_to_submit = content_to_submit.substr( 0,
content_to_submit.length - 1);

        }

        content_to_submit += ';' + escape(form_element.value);

    } else

    {

        content_to_submit += form_element.name + '='

        + escape(form_element.value);

    }

    content_to_submit += '&';

    last_element_name = form_element.name;

}

break; }

}

content_to_submit = content_to_submit.substr(0, content_to_submit.length
- 1);

return content_to_submit;

}
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : KAYA, Mehmet Salih

Uyruğu : T.C.

Doğum tarihi ve yeri : 29.04.1980 Ankara

Medeni hali : Bekar

E-posta : msalikhkaya@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Yılı
Yüksek Lisans	Gazi Üniversitesi Bilişim Enstitüsü	2008
Lisans	Gazi Üniversitesi E.S.E.F Bilgisayar Eğitimi	2002
Lise	Batıkent Süper Lisesi	1998

İş Deneyimi

2002-2003	Zübeyde Hanım P.İ.O Balya Balıkesir – Bilgisayar Öğrt.
2004-2007	Batıkent End. Mes. Lisesi Ankara – Bilgisayar Bölüm Şefi
2007-	TED Ankara Koleji Vakfı Özel Lisesi Ankara – Bilg. Öğrt.

Yabancı Dil

İngilizce

Hobiler

Bilgisayar teknolojileri, masa tenisi, gokart, sinema, satranç