

**İSTANBUL ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ**

# GELENEKSEL WEB TABANLI UYGULAMALARDAN AJAX İLE DESTEKLENEN ZENGİN İNTERNET UYGULAMALARINA GEÇİŞ

**Cüneyt KURU**

**Fen Bilimleri Ana Bilim Dalı**

**Enformatik Programı**

**Danışman**

**Doç.Dr. Sevinç GÜLSEÇEN**

**Eylül, 2008**

**İSTANBUL**



**İSTANBUL ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ**

# GELENEKSEL WEB TABANLI UYGULAMALARDAN AJAX İLE DESTEKLENEN ZENGİN İNTERNET UYGULAMALARINA GEÇİŞ

**Cüneyt KURU**

**Fen Bilimleri Ana Bilim Dalı**

**Enformatik Programı**

**Danışman**

**Doç.Dr. Sevinç GÜLSEÇEN**

**Eylül, 2008**

**İSTANBUL**

Bu çalışma 14/10/2008 tarihinde aşağıdaki jüri tarafından ENFORMATİK Anabilim Dalı ENFORMATİK programında Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Jürisi

Doç. Dr. Sevinç GÜLSEÇEN (Danışman)
İstanbul Üniversitesi

Prof. Dr. Sıddık YARMAN
İstanbul Üniversitesi

Doç. Dr. Selim YAZICI
İstanbul Üniversitesi

Yrd. Doç. Dr. Zuhal TANRIKULU
Boğaziçi Üniversitesi

Yrd. Doç. Dr. Zerrin AYVAZ REİS
İstanbul Üniversitesi

# ÖNSÖZ

Yüksek lisans öğrenimim sırasında ve tez çalışmalarım boyunca gösterdiği her türlü destek ve yardımlarından dolayı, değerli hocam Doç.Dr.Sevinç GÜLSEÇEN'e en içten dileklerimle teşekkür ederim.

**Eylül, 2008** **Cüneyt KURU**

## İÇİNDEKİLER

**ÖNSÖZ ..... I**

**İÇİNDEKİLER ..... II**

**ŞEKİL LİSTESİ ..... V**

**TABLO LİSTESİ ..... VI**

**ÖZET ..... VII**

**SUMMARY ..... VIII**

**1. GİRİŞ ..... 1**

**2. WEB STANDARTLARI ..... 3**

**2.1 W3C STANDARTLARI ..... 3**

**2.2 ECMA STANDARTLARI ..... 6**

**2.3 WEB STANDARTLARINI KULLANMA GEREKLİLİĞİ ..... 7**

**2.4 YAPI VE SUNUM KATMANLARI ..... 8**

**3. WEB UYGULAMALARI GELİŞTİRME TEKNİKLERİNİN EVRİMİ ..... 9**

**3.1 CGI (COMMON GATEWAY INTERFACE) ..... 10**

**3.2 APPLETLER ..... 10**

**3.3 JAVASCRIPT ..... 11**

**3.4 SERVLET, JSP, ASP VE PHP ..... 12**

**3.5 FLASH ..... 15**

**3.6 DHTML ..... 16**

**3.7 XML VE TÜREVLERİ ..... 17**

**4. ZENGİN İNTERNET UYGULAMASI NEDİR ? ..... 20**

**4.1. KULLANICI DENEYİMİ TASARIMI VE KULLANICI ODAKLI GELİŞTİRME.....21**

**4.1.1 Kullanıcı Deneyimi Tasarım Metodolojisi.....22**

**4.2. ZENGİN İNTERNET UYGULAMALARI GELİŞTİRME ARAÇLARI ...23**

**5. AJAX NEDİR ? .....25**

**5.1. GELENEKSEL WEB TABANLI UYGULAMALAR İLE AJAX TEKNİĞİNİN KARŞILAŞTIRILMASI.....28**

**5.2. ÇEVRİMİÇİ VE ÇEVRİMDIŞI DENEYİMLERİN BİRLEŞTİRİLMESİ 32**

**5.3. XMLHTTPREQUEST NESNESİ .....32**

**5.3.1 Yordam ve Özellikleri .....33**

**5.3.2 XMLHTTPRequest Nesnesinin Ajax Yaklaşımı İçerisindeki Kullanımı34**

**5.4 ÖRNEK KULLANIM VE GELENEKSEL YÖNTEMLERLE KARŞILAŞTIRMA .....36**

**5.4.1 İkili seçim listesi uygulaması (Double-Combo Script) : .....36**

**5.5. AJAX ALTYAPI VE ARAÇ TAKIMLARI.....38**

**5.6 AJAX'IN KULLANIM ALANLARI, AVANTAJLARI VE DEZAVANTAJLARI .....40**

**5.6.1 KULLANIM ALANLARI.....40**

**5.6.2 AVANTAJ VE DEZAVANTAJLARI .....41**

**5.7 AJAX YAKLAŞIMINI KULLANAN ÖRNEK UYGULAMALAR .....43**

**5.7.1 Google Maps .....44**

**5.7.2 Google Suggest.....45**

**5.7.3 Flickr.....46**

**5.7.4 Facebook .....47**

**5.8 AJAX VE GÜVENLİK .....48**

**5.8.1 XMLHTTPRequest Açıkları....48**

**5.8.2 Arttırılmış Saldırı Alanı....48**

**5.8.3 SQL Injection....49**

**5.8.4 Çapraz Site Betik Saldırıları (Cross-site Scripting Attacks)....49**

**5.8.5 Hizmetin Reddi Saldırıları (Denial Of Service Attack- DoS)....50**

**5.8.6 Ajax Köprüleme....50**

**5.8.7 Tarayıcı Tabanlı Saldırılar....50**

**5.8.8 Sonuç....51**

**5.9 AJAX VE WEB 2.0....51**

**6. ÖRNEK BİR UYGULAMA:ÖĞRENC-IM....53**

**6.1 AMAÇ....53**

**6.2 TEORİ VE METODOLOJİ....53**

**6.2.1 Operator (yetkili) modülü....54**

**6.2.2 Öğrenci Girişi....54**

**6.3 TEKNOLOJİ....57**

**6.3 VERİTABANI YAPISI....58**

**7. SONUÇ....59**

**KAYNAKLAR....61**

**ÖZGEÇMİŞ....64**

# ŞEKİL LİSTESİ


**Şekil 3.2** : Örnek bir xml dökümanı. 4
**Şekil 3.3** : Örnek bir XHTML dökümanı. 5
**Şekil 3.4** : Örnek bir CSS dökümanı. 5
**Şekil 2.1** : Sun firmasının sitesinden saat Appleti örneği 12
**Şekil 4.1** : Ajax'ın dört ana bileşeni ve çalışma mantığı. 28
**Şekil 4.2** : Klasik Web tabanlı uygulama modeli [29]. 28
**Şekil 4.4** : Ajax uygulamalarındaki asenkron etkileşim modeli. 30
**Şekil 4.5** : Ajax yaklaşımlı form uygulaması ardışıklık diyagramı. 31
**Şekil 4.6** : İkili seçim listesi uygulaması istemci taraflı çözüm. 36
**Şekil 4.7** : İkili seçim listesi uygulaması sunucu taraflı çözüm 37
**Şekil 4.8** : İkili seçim listesi uygulaması ajax temelli çözüm 38
**Şekil 5.1** : Goggle Maps uygulamasından bir görüntü 44
**Şekil 5.2** : Google Maps ile kiralık ev ilanları uygulamasından bir görüntü. 45
**Şekil 5.3** : Google Suggest uygulamasından bir görüntü. 46
**Şekil 5.4** : Facebook uygulamasından bir görüntü. 47
**Şekil 6.1** : Öğrenc-im uygulaması operator ekranları akışı. 55
**Şekil 6.2** : Öğrenc-im uygulaması öğrenci ekranları akışı. 56
**Şekil 6.3** : Öğrenc-im uygulaması veritabanı yapısı. 58

# TABLO LİSTESİ

**Tablo 2.1** : Ajax tekniğini oluşturan temel elemanlar ve açıklamaları. ..... **26**
**Tablo 4.1** : XMLHTTPRequest nesnesi yordamları..... **33**
**Tablo 4.2** : XMLHTTPRequest nesnesi özellikleri. ..... **34**
**Tablo 4.3** : İkili seçim listesi örneğinde kullanılan üç yaklaşımın karşılaştırılması. ..... **38**

# ÖZET


## GELENEKSEL WEB TABANLI UYGULAMALARDAN AJAX İLE DESTEKLENEN ZENGİN İNTERNET UYGULAMALARINA GEÇİŞ

Geleneksel sunucu-istemci modelli Web tabanlı yazılım geliştirme yaklaşımı, uygulamaların kullanımında zorluklara neden olmakta, eğitim giderlerini artırmakta, yatırım genel geri dönüşünü ise azaltmaktadır. Diğer taraftan, internet kullanıcıları Web sayfasının da masaüstü programları gibi çalışılabilmesini istemektedir.

Bunu sağlayan ve "Zengin İnternet Uygulamaları-ZİU (Rich Internet Applications-RIA)" olarak adlandırılan yeni kuşak yaklaşım, masaüstü uygulamalarının zengin grafiksel arayüzünü Web'in evrensel erişimiyle bir araya getirerek kullanıcıların Web sayfaları ile olan etkileşimini mümkün kılmakta, bant genişliği kullanımını ve sunucu yükünü azaltmaktadır; dolayısıyla, son kullanıcı deneyimini iyileştirmektedir.
ZİU yaklaşımında en önemli görevi üstlenen Ajax tekniğidir. Önceleri, Dynamic HTML ve Remote Scripting adları ile de anılmakta olan Ajax, aynı zamanda Web 2.0 yaklaşımının da bir bileşendir.

Bu tez çalışmasında, geleneksel sunucu-istemci modelli Web tabanlı yazılım geliştirme yaklaşımının evrimi ile ilgili bilgiler verildikten sonra, ZİU yaklaşımı ve araçları tüm ayrıntıları ile anlatılmakta; Ajax ile geliştirilmiş olan belli başlı uygulamalar özetlenmektedir.

Ajax ile geliştirilmiş ve son kullanıcılar tarafından büyük ilgi görmüş olan; google maps (harita uygulaması), flickr (fotoğraf paylaşım ve telifleme sitesi), google suggest (alternatif arama sonuçlarının önceden listelenmesini sağlayan uygulama) gibi uygulamalardan yine tez içerisinde bahsedilmektedir.

Bu tez kapsamında, Web üzerinden anlık mesajlaşma imkanı sağlayan ve Ajax tekniğiyle geliştirilmiş olan Öğrenc-im isimli örnek bir uygulama yer almaktadır. Uygulama şimdilik sadece Enformatik Bölümü öğretim üyeleri ve Yüksek Lisans öğrencilerinin verileriyle çalışacak şekilde hazırlanmış ve prototip olarak geliştirilmiştir; ihtiyaçlar çerçevesinde genişletilebilecektir.

# SUMMARY


## TRANSITION FROM TRADITIONAL WEB-BASED APPLICATIONS TO RICH INTERNET APPLICATIONS POWERED WITH AJAX

Traditional server-client model web based software development approach cause difficulties in the utilization of applications, increase training costs, and reduce general return of investment. On the other hand, internet users want web pages run as desktop programs.

The new generation approach enabling this is called Rich Internet Applications (RIA), which enable interaction of users with web pages by combining rich graphical interfaces of desktop applications with universal connectivity of the web, reduce bandwidth uses and server loads; they therefore enhance end-user experience.

The Ajax technique plays the most important role in the RIA approach. Formerly referred to as Dynamic HTML and Remote Scripting as well, Ajax is at the same time a component of the Web 2.0 approach.

In this thesis study, after providing background information on the evolution of the traditional server-client model web based software development approach, the RIA approach and tools are discussed in detail and main applications developed using Ajax are outlined.

Applications such as Google Maps (map application), Flickr (photo sharing and copyrighting website), and Google Suggest (an application that prelists alternative search queries as a search query is being typed) are mentioned in the study.

This study comprises an instant messaging application called Ogrenc-im which is implemented using Ajax. The application which is in its prototype stage is running by using the data related with the students and faculty members of Informatics Department

# 1. GİRİŞ

İnternet, soğuk savaşın devam ettiği 60'lı yıllarda, olası felaketlerden (doğal afet, bomba vs.) etkilenmeyecek bir iletişim ağı ihtiyacından dolayı çıkmıştır. Amerika Birleşik Devletleri Savunma Bakanlığı Gelişmiş Araştırma Projesi Dairesi'nin geliştirdiği (Department of Defense's Advanced Research Project Agency) (ARPA) internet, savunmayla ilgili bütçe desteği alan tüm üniversitelere yerleşmiş ve zaman içerisinde eğitim amaçlı bir iletişim ağına dönüşmeye başlamıştır. Sonraki yıllarda iş dünyasının da internete el atmasıyla birlikte, internet kullanımı daha da artmıştır. Elektronik posta, sohbet, grafik ve resim alış verişini mümkün kılan internet, PC (Personal Computer - masaüstü bilgisayarların genel adı) fiyatlarının da ucuzlamasıyla birlikte neredeyse her eve girmiştir.

İnternetin kalbi olarak kabul edilen WWW (World Wide Web) ise ilk olarak Mart 1989 da, İsviçre'de Tim Berners-Lee [1] tarafından, CERN (Conseil Européen pour la Recherche Nucléaire- Avrupa nükleer araştırma merkezi) fizik labarotuvarlarında, farklı bölümlerde araştırma yapan öğretim görevlilerinin, araştırma sonuçlarını etkin bir şekilde paylaşabilmeleri gereksiniminden yola çıkılarak, geliştirilmeye başlanmıştır. Bu geliştirme çalışmaları sonucunda ortaya çıkan dil, bilgisayarda yeralan dökümanların, anahtar kelimeler vasıtasıyla birbirlerine bağlantılı hale getirilmesini sağlamış, sonrasında, birbirine bağlı olarak organize edilmiş dökümanlar, farklı bilgisayarlar arasında internet'e bağlıyken değiş-tokuş yapılabilir hale gelmiştir. Dökümanları birbirine bağlamak için kullanılan bu dile HTML (Hypertext Markup Language) adı verilmiş ve Web sayfaları 1992 yılına kadar sadece statik metin içeriklerinden oluşacak şekilde kalmışlardır. HTML, bir kaynaktan dağıtılan ve ortak kullanıma açık olan hiperortam bilgi sistemleri için uygulama seviyesinde bir iletişim kuralıdır. Sonraki yıl ise; İllinois Üniversitesi, Süper Hesaplama Uygulamaları Ulusal Merkezi'nde (National Center for Supercomputing Applications) ilk çapraz geçişli grafiksel tarayıcı olan Mosaic ortaya çıkmış ve böylece günümüz internet tarayıcılarının geliştirilmesine yönelik ilk adım atılmıştır.

Bugün, internet üzerinden ulaşılabilir bilginin hacmi çok büyük boyutlardadır.

Kısa zamanda büyük bir değişim ve gelişim evresi geçiren Web, yazılım geliştirme teknikleri açısından da, ihtiyaca yönelik olarak sürekli yenilenmektedir. Bilgiye ulaşım için kullanılan internet, erişilecek bilgiyi kullanıcının kendisinin oluşturabilmesi imkanının yaratılması ile birlikte yeni bir döneme girmiştir. O' Reilly ve MediaLive International firmalarının yaptıkları bir beyin fırtınası konferası sırasında ortaya çıkan ve Web 2.0 olarak adlandırılan bu dönem, son kullanıcılar için Zengin İnternet Uygulaması (ZİU) geliştirme tekniklerinin ortaya çıkmasını da beraberinde getirmektedir.

Bu tez çalışmasında ZİU'na geçiş süreci açıklanmaya çalışılmış; bu amaçla ZİU öncesi Web ve uygulamalarının zaman içerisinde değişmesi ve gelişmesi anlatılmıştır. Tezin ilk bölümünde Web uygulaması geliştirme tekniklerinin evriminden bahsedilmiş ve bu teknikler konusunda bilgiler verilmiştir. Tezin ikinci bölümünde ise, ZİU nedir sorusunun cevabı verilerek, geliştirme tekniklerinden bahsedilmiş, bu tekniklerden biri olan Ajax tekniğiyle beraber, geleneksel Web uygulamaları geliştirme yaklaşımının nasıl değiştiği anlatılmıştır. Ayrıca, Web standartları konusuna değinilmiş ve nasıl standartlara uygun kod yazımı gerçekleştirilebileceği konusunda bilgi verilmiştir. Üçüncü bölümde, Ajax kullanımının getirdiği kazanç ve riskler aktarılarak, uygulama güvenliği konusunda bilgiler verilmiş ve Ajax'ın Web 2.0 yaklaşımı içindeki yeri tartışılmıştır. Tezin son bölümünde ise, Ajax tekniği kullanılarak hazırlanan, ve Web üzerinden anlık mesajlaşma imkanı sağlayan öğrec-im isimli örnek çalışma sunulmuştur.

# 2. WEB STANDARTLARI

Web standartları, W3C (World Wide Web Consortium-World Wide Web için uluslarası standartlar organizasyonu) adı verilen ve Web'in mucidi Tim Berners-Lee tarafından MIT (Massachusetts Institute of Technology-Camridge, Massachusetts'de bulunan, özel bir üniversite) ve CERN bünyesinde kurulmuş olan, uluslararası konsorsiyum tarafından belirlenmektedir. W3C, Web ve Web'i oluşturan teknolojilerin standartizasyonunu ve işlerliğini sağlamak için kurulmuştur. Apple, Adobe, IBM, Macromedia, Microsoft ve Sun gibi bilişim dünyasının devleri olarak kabul edilen kuruluşlar, W3C'ye üyedirler.

Web standartları ayrıca, Web tabanlı içeriğin yaratılmasında ve çözümlenmesinde kullanılan teknolojiler olarak da değerlendirilebilmektedir. Bu teknolojiler, gelecekte de doğru biçimde görüntülenebilecek dökümanların Web'de yayımlanmasını ve bu dökümanların olabildiğince çok insan tarafından erişilebilir olmasını sağlamak üzere tasarlanmışlardır.

Web üzerinden sunulan dökümanlara, farklı işletim sistemlerinde çalışan, birçok tarayıcı araç kullanılarak, değişik boyutlardaki ve kalitelerdeki monitörler aracılığıyla ve tarayıcılarının varsayılan metin büyüklüğünü ve diğer seçeneklerini de değiştirmiş kullanıcılar tarafından erişilmektedir. Web sayfaları hazırlayan herkez, kağıt üzerinde yayın çıkaranlar ya da televizyon için filmler üretenler gibi düşünülmesi gereken önkoşulların varolduğunu bilmek zorundadır.

Web standartları, bazı yapısal ve sunum dillerinin, nesne modellerinin ve betik (script) dillerinin standartizasyonlarının birleşmesinden oluşmaktadır.

Standartlar ve bileşenleri aşağıda incelenmiştir.

## 2.1 W3C STANDARTLARI

W3C aşağıda bahsedilmekte olan yapısal ve sunum dillerini ve nesne modellerini standartlaştırmıştır [4].

**HTML 4.0 - HyperText Markup Language (Zengin metin işaretleme dili) :**

HTML, internet üzerinde veri paylaşımı ve metin tabanlı dökümanları biçimlendirmek için kullanılan en yaygın, yapısal dildir. Tarayıcılar, HTML dökümanlarını yorumlamakta ve Web kullanıcılarının anlayabileceği formatta sunmaktadırlar.

**XML 1.0 - Extensible Markup Language (Genişletilebilir işaretleme dili) :**

XML, HTML gibi bir yapısal işaretleme dilidir. Temel işlevi veri depolamaktır. W3C tarafından standartlaştırılmıştır. XML günümüzde çoğunlukla farklı bilgisayar sistemleri arasında veri transferi için kullanılmaktadır.

Uygulama geliştiriciler tarafından oluşturulan, yeni elemanı yapısına eklemeye izin verdiği için, HTML'den daha esnek bir dildir. Büyük ölçekteki metin tabanlı verilerin saklanmasını sağlamaktadır. Şu an bir çok arama motoru ve veritabanı uygulamasında da kullanılmaktadır.

XML dökümanları ağaç yapısındadırlar. Şekil 3.2'de örnek bir XML dökümanının [26] görüntüsü verilmiştir.

```
<bookstore>
<book category="COOKING">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>
<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
<book category="WEB">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
</bookstore>
```

Şekil 3.2 Örnek bir xml dökümanı.

**XHTML 1.0 - Extensible Hypertext Markup Language (Genişletilebilir zengin işaretleme dili)**

XHTML, HTML in bir XML uygulaması olarak yeniden düzenlenmesinden ortaya çıkmış ve HTML'in yerini almaya çalışan yapısal bir dildir. HTML 4.0.1 sürümü ile

benzer birçok yönü olmasına karşın, HTML'den daha kolay anlaşılır ve kuralcı bir dildir.

HTML'den farklı olarak, döküman yapısı içerisine yeni elemanların eklenebilmesine izin verir.

XHTML dökümanları tarayıcılar tarafından HTML'e benzer bir şekilde gösterilmektedir.

XHTML'in gelişen yapısı, şu anda ve gelecekte, sadece kişisel bilgisayarlar için değil, cep telefonları ve avuç içi cihazlar için de bir standart olma yolundadır.

Basit bir XHTML örneği şekil 3.3'de gösterilmiştir [27].

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>simple document</title>
</head>
<body>
<p>a simple paragraph</p>
</body>
</html>
```

Şekil 3.3 Örnek bir XHTML dökümanı.

**CSS - Cascading Style Sheets (Biçim Şablonları)**

CSS, HTML veya XML dosyaları içerisinde yeralan elemanların, tanımlanan biçimlerde gösterilmesini sağlayan, tüm sitenin görünümünü değiştirebilen bir mekanizmadır. Basit, fakat daha yapısal ve kolaylıkla müdahele edilebilen Web sayfaları yaratmak için kullanılan en etkili ve kullanışlı sunum dilidir. CSS kullanımıyla ilgili basit bir örnek şekil 3.4'de verilmiştir (http://www.w3schools.com/css/css_howto.asp adresinden alınmıştır).

```
<head>
<style type="text/css">
<!--
hr {color: sienna}
p {margin-left: 20px}
body {background-image: url("images/back40.gif")}
-->
</style>
</head>
```

Şekil 3.4 Örnek bir CSS dökümanı.

**Document Object Model (Belge Nesnesi Modeli)**

İnternet tarayıcıları girilen internet sayfasını bir belge, bu belge (sayfa) içerisinde bulunan tüm elemanları da nesne olarak kabul eder. Buna göre resim, yazı, form gibi tüm elemanlar nesnedir. DOM, sayfa içindeki herhangibir nesnenin özelliğine müdahele edilebilmesi için bir arayüz sağlar. Bunun yapılabilmesi için ise, JavaScript gibi istemci taraflı çalışan bazı betik (script) dillerin kullanılması gerekmektedir. HTML ve CSS le hazırlanmış bir dökümanın elemanlarına kolaylıkla mudahele edilebilmesi, DOM sayesinde gerçekleşebilmektedir.

Kısaca DOM, bir web sayfası dökümanının, hiyerarşik ve mantıksal yapısını tanımlayan ve dökümanın elemanlarına ulaşmayı ve değiştirmeyi sağlayan bir uygulama geliştirme arayüzüdür (Application Programming Interface - API).

## 2.2 ECMA STANDARTLARI

ECMA (European Computer Manufacturers Association), avrupa bilgisayar üreticileri birliği olarak bilinen ve bilgi ve iletişim teknolojilerine yönelik standartlaştırma çalışmaları yapan, 1961 yılında kurulmuş bir teşkilattır.

ECMA'nın bilgi teknolojileri için hazırladığı standartlar 350'den fazla ve hazırladığı teknik raporlar ise 85'i bulmuştur [4]. ECMA tarafından hazırlanan standartlardan büyük bir çoğunluğu, ISO (International Organization for Standardization - Uluslararası Standartlar Teşkilatı) tarafından kabul görmüştür. ECMA' nın ISO tarafından onaylanmış standartlarına http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html#ISOCopyrightforthefreelyavailablestandards adresinden ulaşılabilir.

**ECMAScript (JavaScript Standartizasyonu)**

ECMAScript, betik dilleri standartlaştırmakta, büyük ölçüde Netscape'in JavaScripti ve Microsoft'un Jscript'ine dayanmaktadır. ECMAScript standartı ECMA'nın teknik komitesi tarafından tanımlanmıştır ve http://www.ecma-international.org/memento/TC39.htm adresinden ulaşılabilmekterdir.

ECMAScript'in ana kullanımı, Döküman Nesne Modeli (DOM) olarak tanımlanmış olan Web sayfalarında ki nesnelerin manipüle edilmesidir. Bu nesneler, (Web sayfası yapımında kullanılan tüm öğeler) değiştirilebilir, silinebilir, taşınabilir ya da özellikleri

değiştirilebilir nesnelerdir. Güncel ECMAScript şartnamesine http://www.ecma-international.org/publications/standards/Ecma-262.htm adresinden ulaşılabilir.

## 2.3 WEB STANDARTLARINI KULLANMA GEREKLİLİĞİ

Johansson'a göre [5] Web standartları aşağıdaki sebeplerden dolayı kullanılmalıdır;

**Daha kolay geliştirme ve bakım:** Daha semantik ve yapılandırılmış HTML kullanmak başka birisi tarafından yazılmış kodu anlamayı kolaylaştırmakta ve hızlandırmaktadır.

**Gelecekte ortaya çıkacak tarayıcılarla uyumluluk:** Tanımlanmış standartlar kullanılarak geçerli kod yazımı sayesinde, yeni çıkacak tarayıcıların dökümanları anlamaması riski azaltılmaktadır.

**Web sayfalarının daha hızlı indirilmesi ve çözümlenmesi:** Daha az HTML kodu daha küçük dosya boyutu ve daha hızlı indirilme demektir. Modern tarayıcılar, sayfaları, standart haldeyken daha hızlı çözümleyebilmektedirler.

**Daha fazla erişilebilirlik:** Semantik HTML, yani yapının sunumdan ayrıldığı HTML, tarayıcı araçların ve alternatiflerinin içeriği daha kolay algılamasını sağlamaktadır.

**Daha yüksek arama motoru sıraları:** İçeriğin ve sunumun ayrılması içeriğin sayfa boyutunun çok daha fazlasını temsil etmesi anlamına gelmektedir. Semantik kodlarla birlikte bu durum, sayfanın arama motoru sırasını yükselten bir etken olmaktadır.

**Daha kolay uyum:** Düzgün, standartlara uygun biçimde oluşturulmuş bir döküman, yalnızca farklı bir CSS dosyası ile ilişkilendirilerek yazdırma işlemi ya da el bilgisayarları veya cep telefonları gibi alternatif tarayıcı araçlarla çok daha kolay uyumlu hale getirilebilmektedir.

Web standartlarına uyulması yani standartlara uygun kod yazılması, Web sitesi yaratıcılarına para ve zaman kazandırır, ziyaretçilere ise çok daha iyi bir kullanıcı deneyimi yaşatır [5].

### Geçerleme (Validation)

Geçerleme, bir dökümanın, yazıldığı dilin kurallarına uyup uymadığını kontrol etme işlemidir. Bu işlem ile döküman içerisindeki metnin sözdizim kurallarına uyumluluğu kontrol edimiş olur.

Geçerleme, Web sayfaları geliştirme sürecinin çok önemli bir parçasıdır. Farkedilmesi zor birçok hata, geçerleme sırasında bulunur. Hatalar, yazım hataları kadar basit

olabileceği gibi yanlış kullanılan bir elaman ya da özellik nedeniyle çok daha ciddi de olabilmektedir.

Birçok tarayıcı hata vermek yerine, geçersiz HTML kodlarını ellerinden geldiğince düzgün biçimde çözümlemeye ve yazan kişinin ne kastettiğini becerebildiği kadar kestirmeye çalışmaktadır. Bu da web uygulaması geliştiricilerin çoğu zaman Web standartlarına uymamasını beraberinde getirmektedir. Bu tür kodlar kestirilemeyen sonuçlar ürettikleri ve Web tarayıcıların hata işleme yöntemlerine güvendiklerinden dolayı sorunludurlar.

Yerleşik geçerleme araçlarına sahip birçok HTML editörü bulunmaktadır. Bunun dışında, eğer kullanılan geliştirme aracının yerleşik geçerleme yöntemleri yoksa, W3C'nin çevrimiçi erişilebilen geçerleme servislerinden (W3C MarkUp Geçerleme Servisi http://validator.w3.org/ ve W3C CSS Geçerleme Servisi http://jigsaw.w3.org/css-validator/ )
yararlanılabilmektedir.

## 2.4 YAPI VE SUNUM KATMANLARI

Web standartlarından sözedilirken sıkça değinilen konulardan biri de yapıyı ve sunumu belirleyen kodların birbirinden ayrılmasıdır. Dökümanın semantik yapısını anlamak, yapı ve sunum katmanlarını ayırabilmek için çok önemlidir. Yapı ve sunum katmanları birbirinden ayrıldığında bir dökümanın sunumunu CSS ile kontrol etmek çok daha kolay hale gelmektedir.

Yapı bir dökümanın olmazsa olmaz parçaları ve içeriğinin semantik ve yapısal kodudur. Sunum ise içeriğe verilen biçimdir, dökümanın nasıl göründüğü ile ilgilidir.

Yapıyı ve sunumu mümkün olduğunca birbirinden ayırmak gerekmektedir. İdeal olarak, sadece yapıyı ve içeriği kapsayan bir HTML dökümanı ve sunumu kontrol etmeyi sağlayan bir de CSS dökümanı olmalıdır.

### Semantik HTML

Yapıyı sunumdan ayırmak için kullanılan diğer bir yöntemde, dökümanın yapısını kodlamak için semantik HTML kullanmaktır. Dökümanın ilgili bölümünde kullanıma uygun yapısal anlamı olan bir XHTML etiketi varsa onun yerine başka bir şey

kullanmaya yani bir HTML etiketini başka bir HTML etiketi gibi göstermek için CSS kullanımına gerek yoktur. (Örneğin; bir başlığı kodlamak için <h1> yerine  elemanını kullanmak vb.)

Semantik HTML kullanarak bir dökümanın farklı bölümlerinin tüm Web tarayıcıları için anlamlı olması sağlanabilmektedir. Bu tarayıcının, modern bir PC'deki en son grafik tabanlı Web tarayıcısı, CSS'i işleyemeyen eski bir tarayacı ya da Unix üzerinde metin tabanlı bir tarayıcı olması durumu değiştirmemektedir.

# 3. WEB UYGULAMALARI GELİŞTİRME TEKNİKLERİNİN EVRİMİ

İnternet'in ortaya çıktığı ilk zamanlarda, bütün Web sayfaları statik metin tabanlı dökümanlardan oluşmaktaydı. Kullanıcılar internet üzerinden bir kaynağa erişmek için istekte bulunduktan sonra, sunucudan dönen sonuç statik bir döküman olarak gösterilmekteydi. Web sayfaları, bir dökümanın elektronik kopyasından ve bu dökümanın kolaylıkla dağıtılmasından öteye geçememekteydi. Tarayıcıların ortaya çıkışının ilk zamanlarında, Web sayfalarının bu statik doğası bir sorun teşkil etmemekteydi; bilim adamları interneti araştırma sonuçlarını paylaşmak için, üniversiteler ise derslerle ilgili bilgileri çevrimiçi verebilmek adına kullanmaktaydılar [1]. Sadece, metin tabanlı bir işaretleme dili olan HTML vasıtasıyla veri paylaşımı sağlanan bu kanalı, iş ve ticaret çevreleri nasıl kullanacaklarını henüz anlayamamışlar ve başlangıçta sadece şirketleriyle ilgili irtibat bilgilerini ve bazı dosyaları sunmak için kullanmışlardır.

Web kullanıcılarının dinamik bir deneyim talep etmesi çok uzun zaman almamıştır. PC sayısı ve kullanımının günden güne artması, işletim sistemlerinin gelişen arayüz ve beraberlerinde getirdikleri masaüstü uygulamalarının yaygın kullanımı, kullanıcıların taleplerinin biranda artmasına neden olmuştur.

## 3.1 CGI (COMMON GATEWAY INTERFACE)

Web sayfalarını daha dinamik hale getirebilmek için ilk çözüm, türkçe karşılığı "ortak geçit arayüzü" olan Common Gateway Interface (CGI) oldu. CGI, kullanıcı bir istekte bulunduğunda çalışan, veritabanı sunucusuna bağlanabilen ve sonuçları döndürebilen programlar yaratmaya olanak sağlamaktadır. CGI betikleri, Perl'den (Practical Extraction and Report Language - Pratik Çıkarım ve Raporlama Dili) Visual Basic'e kadar birçok dilde yazılabilmektedir. Visual Basic, Microsoft tarafından, Basic programlama dili üzerinde geliştirilmiş, olay yönlendirmeli, üst seviye, nesne tabanlı ve görsel bir programlama dilidir.

Tüm bunlara rağmen, CGI dinamik Web uygulamaları geliştirmede güvenli bir yöntem olarak kabul edilmemektedir. CGI ile birlikte, uygulama geliştirciler, kullanıcıların kendi sistemlerinde bir program çalıştırmasına izin vermişlerdi. Çoğu zaman bu durum büyük bir problem yaratmasa da, kötü niyetli bir kullanıcının sistemi sömüren, onun da ötesinde yok edebilecek bir uygulama çalıştırması olasıdır. Bu dezavantajına karşın CGI, günümüzde halen kullanılmaya devam etmektedir.

## 3.2 APPLETLER

1995 yılında piyasaya çıkışı ilan edilen Java programlama dilinin kullanımı kısa sürede yaygınlaştı. Netscape, tarayıcı uygulamalarında bu yeni dilin kullanımını destekleyeceğini açıkladı ve Java dinamik Web sayfaları oluşturmada yeni bir yöntem haline geldi [1].

Appletler, programcılara Web sayfalarına yerleştirilebilecek ufak uygulamalar yazmaya imkan sağlamıştır. Kullanıcılar, bilgisayarlarına yüklediklerini Java Virtual Machine (JVM) uygulaması sayesinde, Applet uygulamalarını çalıştırabilmekteydiler. Birçok şey yapabilmeye imkan sağlamasına rağmen Applet uygulamalarının da kısıtları bulunmaktaydı.

Örneğin istemci bilgisayarın dosya sisteminde, okuma ve yazma yapmaları engellenebiliyordu, bu yüzden yerel kütüphaneleri kullanamıyorlar; istemci tarafında herhangibir program başlatamıyorlardı.

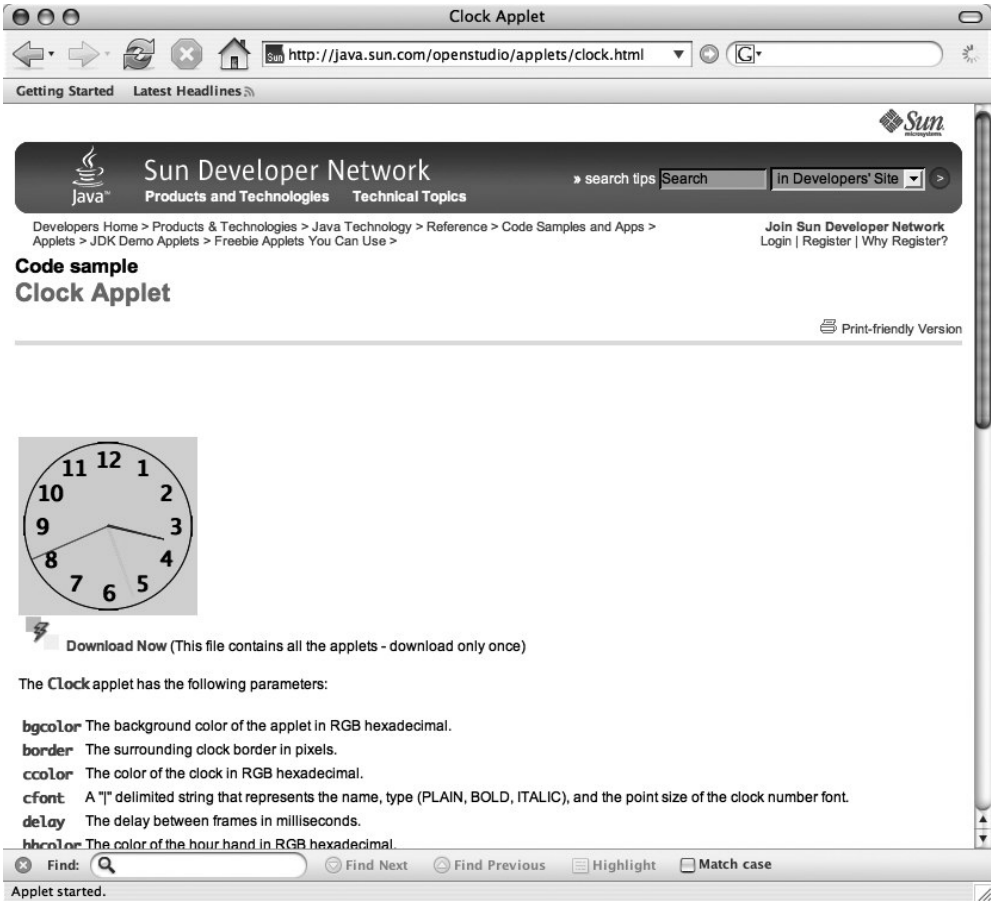
Oldukça fazla sayıda güvenlik kısıtı ve ayrıca geliştiricilerin hazırladıkları Java Appletinin doğru çalışabilmesi için tarayıcıda çalışan JVM uygulamasının sürümünün doğruluğunu kontrol etme gerekliliği olsa da, dinamik Web uygulaması geliştirmek

isteyen birçok yazılım geliştirici, Java diline ve Appletlere ilgi göstermiştir. Şekil 2.1'de Sun firmasının sitesinden alınmış bir Applet örneği gösterilmektedir.

### 3.3 JAVASCRIPT

Java'daki gelişmeler devam ederken, Netscape JavaScript adında yeni bir betik dil yaratmıştır (piyasaya çıkarılmadan önce, prototip zamanında Mocha, sonrasında LiveWire ve LiveScript olarak anılmaktaydı). JavaScript, Java ile ilgilenmeyen Web tasarımcıları ve programcılar için, yeni bir yol yaratmak adına tasarlanmıştı ve aynı zamanlarda Microsoft firması da VBScript'i çıkartmıştı. Kullanışlı hata mesajları ya da geliştirme araçları olmadığı halde, JavaScript dinamik Web uygulamaları geliştirmede tercih edilen bir araç haline gelmiştir. JavaScript, Web uygulaması geliştiricilerin istemcilere, zengin bir kullanıcı deneyimi yaratabilmelerini sağlayan güçlü bir platformdur. Tezin ilerleyen bölümlerinde anlatılacak olan döküman nesnesi modelinin (Document Object Model - DOM) ortaya çıkmasıyla birlikte, JavaScript ve DOM ayrılmaz bir ikili haline gelmişlerdir. DOM, bir Web sayfasının, tamamiyle nesne-yönelimli bir şekilde tasvir edilmesidir ve betik diller olan JavaScript ve VBScript ile yeniden düzenlenebilmesine olanak sağlamaktadır. World Wide Web Consortium (W3C) tarafından standartları belirlenmiş olan DOM ve European Computer Manufacturers Association (ECMA) tarafından, ECMAScript talimatnameleri adı altında standartlaştırılmış olan JavaScript, standartlara uygun olarak kodlandığı takdirde, hazırlanan web sayfaları, her tür web tarayıcısında aynı şekilde gözükmektedir. JavaScript kullanımındaki en önemli kısıtlar ise şunlardır:

- İstemcinin istediği zaman tarayıcı uygulama tarafında, JavaScript kullanımını kapatabilmesi;
- Farklı tarayıcılarda, aynı uygulamanın farklı sonuçlar döndürmesi;
- Dil geliştirme ortamlarının yetersizliğinden, uygulama geliştircilerin kod yazımında karşılaştığı zorluklar.




Şekil 2.1: Sun firmasının sitesinden saat Appleti örneği

### 3.4 SERVLET, JSP, ASP VE PHP

Çevirmeli ağ bağlantısı (dial-up) kullanımı zamanlarında, kullanıcıların ve yazılım geliştiricilerin en büyük sorunlarından birisi karmaşık bir Applet uygulamasının istemci bilgisayarlara yüklenmesiydi. Ayrıca, yazılım geliştiriciler istemci bilgisayarlarda çalışan Java sürümü ve Virtual Machine (sanal makine) uygulamasından bekledikleri sonucu alamamışlardı. İnternet aslında tamamen statik sayfalar sunmak için geliştirilmiş bir yapıdır [1]. Bu durumda en iyi çözüm, statik sayfaların dinamik olarak sunucu tarafında oluşturulması ve statik olarak istemciye ulaştırılması olarak görüldü. Java'nın tüm dünyada tanıtılmasından sonra Sun firması Servlet'leri geliştiricilerin kullanımına sundu. Bu sayede, Java kodları tarayıcılarda Applet olarak çalışmamakta, sunucularda çalışabilmekteydiler. Servletler, geliştiricilere mevcut hazırlamış oldukları Java uygulamalarını sunucuya taşımalarına izin verdikleri gibi, Java'nın sürümüyle ilgili endişelerinin sadece sunucu tarafındaki uygulamanın sürümünde güncelleme yaparak

sonlanmasını sağlamıştır. Bu durum, geliştiriciler için "birkez yaz, her yerde çalıştır" anlamına gelmektedir. Servletler, Sun firması tarafından CGI betiklerine alternatif olarak sunulmuşlardır.

Java'nın uygulama geliştirme arayüzüne sınırsız erişim ve HTTP (Hypertext Transfer Protocol- Yüksekmetin Aktarma İletişim Kuralı) işlemek için büyük bir kütüphaneye sahip olması, Servletlerin o yıllardaki kullanımını öne çıkarmıştır. Tüm bunlara rağmen, Servletler kullanım açısından mükemmel değildirler. Servletleri kullanarak arayüz tasarımı yapmak çok güçtür. Basit bir Servlet kodu bile, kapladığı alana göre, düşük kalitede bir sonuç üretebilmektedir (Şekil 2.2'de basit bir servlet kodu, Şekil 2.3'de ise bu kodun çıktısı gösterilmiştir). Geliştiriciler için önemli olan sadece iş katmanı değil ayrıca sunum katmanıdır ve Servlet'ler sunum katmanında başarısızdırlar. Bu durum, iş katmanıyla sunum katmanının birbirinden tamamen ayrılması gerçeğini ortaya çıkarmıştır. Java firması, bu sebeplerden dolayı, Java Server Page'leri (JSP) ortaya çıkarmıştır. Aslında JSP'ler Microsoft'un, Sun'ın Servletler konusunda yaptığı hatalardan yola çıkarak geliştirdiği Active Server Page (ASP) çözümünden sonra ortaya çıkmıştır [1]. JSP ve ASP aslında sundukları yapı bakımından eşdeğerdir, ikiside sunum ve iş katmanlarını birbirinden ayırmak için tasarlanmışlardır. Bazı teknik farklılıklar olsada, ikisi de Web tasarımcıların sunuma odaklanmasını, yazılım geliştiricilerin ise iş katmanına odaklanmasını sağlamaktadırlar. Şekil 2.4'de basit bir JSP kod bloğu Şekil 2.5'de bu kodun çıktısı verilmektedir.

Sunucu taraflı yazılım geliştirme ile sadece Microsoft ve Sun firmaları ilgilenmemektedir. PHP'den ColdFusion'a kadar birçok alternatif sunucu taraflı platform bulunmaktadır. Bunlardan bazıları çok kullanışlı araçlar sağlamakta, bazıları ise daha kolay yazılabilen dil yapılarından oluşmaktadır. Sonuçta tüm bu çözümler aynı amaca hizmet etmektedirler: dinamik olarak HTML yaratmak ve içeriği sunucuda oluşturmak ki bu da dağıtım problemlerini çözmektedir. Öte yandan, salt HTML ile sağlanabilecek bir kullanıcı deneyimi Applet'ler ile karşılaştırıldığında sönük kalmaktadır.

```
response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet SimpleServlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Hello World</h1>");
        out.println("<p>Imagine if this were more complex.</p>");
        out.println("</body>");
        out.println("</html>");

        out.close();
```

Şekil 2.2: Basit bir servlet kodu.




Şekil 2.3: Şekil 2.2 de verilen servlet kodunun çıktısı.

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Hello World</title>
    </head>
    <body>

    <h1>Hello World</h1>
    <p>This code is more familiar for Web developers.</p>

    </body>
</html>
```

Şekil 2.4: Basit bir JSP kod bloğu.




Şekil 2.5: Şekil 2.4 de verilen JSP kodunun çıktısı.

### 3.5 FLASH

Microsoft ve Sun dinamik Web sayfaları oluşturmak adına çözüm üreten tekel şirketler değildirler. 1996 yazında FutureWave firması, Java tabanlı bir animasyon uygulaması olan FutureSplash'ı piyasaya tanıtmıştır. FutureSplash, Macromedia firması tarafından FutureWave'in satın alınmasından sonra, Flash olarak yeniden markalandırılmıştır.

Flash, tasarımcılara etkileyici ve dinamik uygulamalar geliştirebilmek için doğal bir ortam sağlamaktadır. Bu sayede firmalar, yüksek etkileşimli uygulamaları Web üzerinden yayınlayabilir hale gelmişlerdir. Appletler, Servletler ve CGI betiklerinden farklı olarak, Flash programlama yeteneği gerektirmeyen ve kolay öğrenilen bir platformdur. Öte yandan, bu kolaylık beraberinde maliyeti de getirmektedir. Şekil 2.6'da örnek bir flash uygulaması gösterilmektedir.

Diğer birçok araçta olduğu gibi, Flash da istemci bilgisayarlarda bir yazılımın çalışmasına gereksinim duyar. ShockWave Player uyumlu eki (plug-in) olarak adlandırılan bu uygulama kısa zamanda bütün tarayıcı uygulama ve işletim sistemlerinde desteklenir hale gelmesine, ve bedava olmasına karşın pekçok kullanıcı virus korkusu nedeniyle bu yazılımın bilgisayarlarına yüklenmesini reddetmektedir.

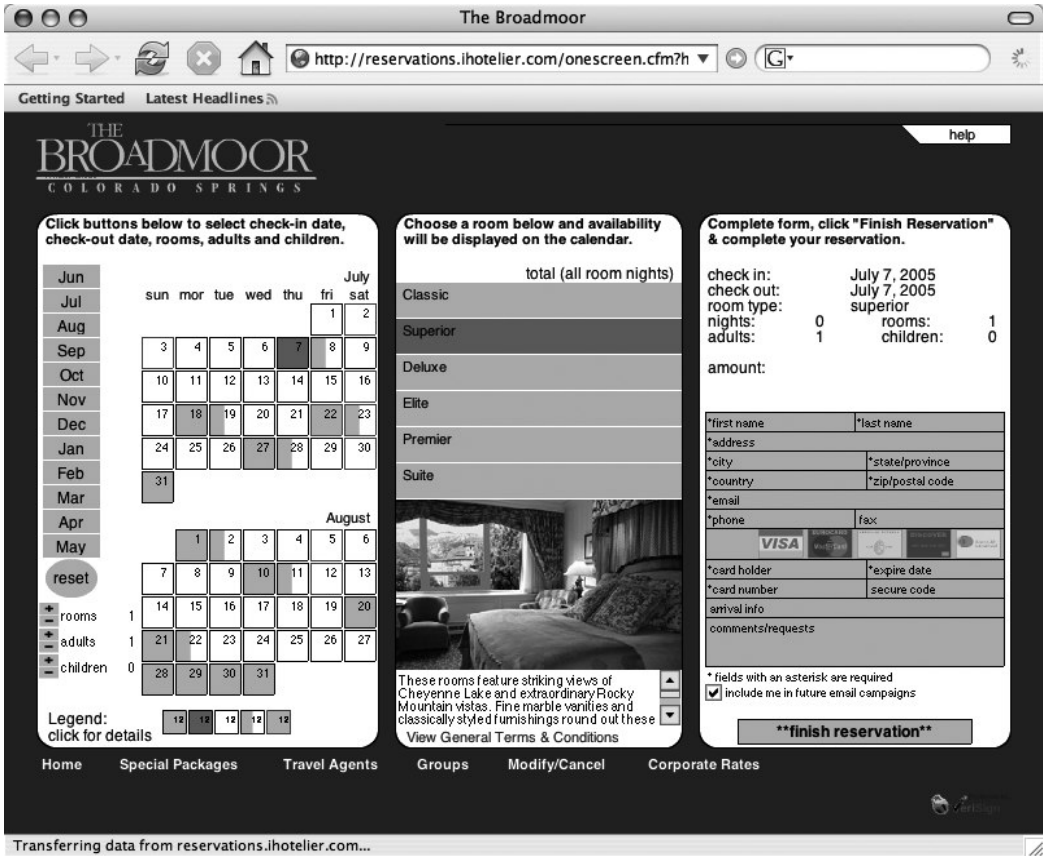
Flash uygulamaları doğru bir şekilde çalışabilmek için önemli ölçüde ağ bant genişliğini kullanırlar ve istemci bilgisayarlara yüklenmeleri gereklidir. Bu nedenle; birçok firma Web sitesinin istemci bağlantı hızına uygun birden fazla sürümünü ayrı ayrı seçenekler olacak şekilde hazırlamaktadır.

Flash uygulamaları yaratmak için uygun yazılım ve tarayıcı eklentilerine ihtiyaç vardır. Tam bir Flash araç takımına (toolkit) sahip olmak binlerce dolarlık maliyet anlamına gelmektedir. Java appletlerinin basit döküman editör (text editor) uygulamalarında ya da bedava sahip olunabilen Java geliştirme takımları (JDK) vasıtasıyla geliştirilebildiği düşünüldüğünde bu maliyet azımsanmayacak niteliktedir. Asleson'a [1] göre engellenemez olmadıkları halde, bu faktörler Flash'ın yaygın olarak kullanımını yavaşlatmaktadır.

### 3.6 DHTML

Microsoft ve Netscape tarayıcı uygulamalarının Sürüm 4'ünü çıkardıklarında, Web geliştiriciler için yeni bir tercih imkanı daha ortaya çıkmış oldu: dynamic HTML (DHTML). DHTML bir W3C standartı değildir. Gerçekte DHTML, HTML, Cascading Style Sheets (CSS), JavaScript ve DOM'un birleşiminden oluşmuştur. Bu teknolojilerin birleşimi, geliştiricilere istenilen zamanda bir Web sayfasının yapısını ve içeriğini değiştirebilme imkanı sunmaktadır.

İnternet Explorer ve Netscape tarayıcı uygulamalarında DHTML'i desteklese de, kod geliştirme bu iki tarayıcı için farklı bir şekilde yapılmaktadır. Bunun anlamı, geliştiriciler için istemcinin hangi tarayıcıyı kullandığını biliyor olmak ve geliştirme maliyetinin artması demektir. Geliştiriciler açısından DHTML kullanmakla ilgili bir diğer tereddütte, halen resmi olarak standartlaştırılmamış olmasından kaynaklanmaktadır.




Şekil 2.6: Bir Flash uygulaması.

### 3.7 XML VE TÜREVLERİ

Farklı türdeki ve yapıdaki dökümanların tanımlanabilmesi için kullanılan ilk işaretleme dili SGML'nin (Standard Generalized Markup Language) bir altkümesi (subset) olarak ortaya çıkan XML (eXtensible Markup Language) yaygın kullanımı olan bir işaretleme dilidir. W3C tarafından standartlaştırılan XML, SGML'nin küçültülmüş bir türü olmakla beraber, Web ortamında uygulanması çok zor olan SGML zorluklarını kullanılabilir kılmaktadır.

Günümüzde, XML türevi diller Web uygulamaları geliştirmede sıkça kullanılmaktadır. Örnek olarak yine W3C'nin standartlaştırdığı, XHTML, Mozilla'nın XUL'i, açık kaynak kodlu bir alternatif olan Java'nın geliştirdiği XAMJ, Macromedia'nın MXML'i ve Microsoft'un XAML'i verilebilir.

Mozilla, Netscape'in kaynak kodunu açmasıyla ortaya çıkan, ve ilk kurulduğundan beri Netscape'in Ağ tarayıcıları için kullandığı kod ismi olan "Mozilla"yı markalaştıran, açık kaynaklı yazılım ailesidir.

Aşağıda bu dillerden kısaca bahsedilmektedir [1].

**XUL:** XUL, ("ZUUL" olarak telaffuz edilmektedir.) bir XML kullanıcı arayüz tanımlama dilidir (XML User Interface Language). Mozilla projesinin bünyesinde geliştirilmiştir. Firefox Web tarayıcısı, Thunderbird e-posta istemcisi XUL kullanılarak yazılmıştır. DHTML'e yakın bir dil olan XUL, geliştiricilerin kolayca öğrenebileceği bir şekilde tasarlanmıştır. XUL, geliştiricilere internete bağlı olunsun ya da olunmasın, zengin uygulamalar (rich applications) yaratma imkanı tanımaktadır. Tek başına bir standart olmamasına karşın HTML 4.0, CSS, DOM, XML ve ECMAScript gibi birçok standartı içinde barındırmaktadır.
XUL kullanımının en önemli dezavantajı, Gecko motoruna ihtiyaç duyması ve İnternet Explorer tarayıcısının bu eklentiyi desteklememesidir. Gecko, Mozilla tabanlı yazılımların merkezinde bulunan ve sayfaları yorumlayıp ekrana yansıtan ana bileşendir. Desteklediği kurallar ve uygulamalar arasında HTML 4.0, CSS 1/2 (ve yakında 3), W3C Document Object Model (Belge Nesnesi Modeli), XML 1.0, RDF ve JavaScript yer alır. Firefox'un tarayıcı kullanımında azımsanmayacak oran sahibi olmasına karşın, İnternet Explorer desteğinin olmaması XUL'in büyük ölçüde yaygınlaşmasını ve uygulama geliştirmede ki kullanımını düşürmektedir.
**XAML:** XAML, ("ZAMEL" olarak telaffuz edilmektedir.) Microsoft'un geliştirdiği yeni işletim sistemi Vista'nın bir bileşenidir. XAML (Extensible Application Markup Language), Vista üzerinde kullanıcı arayüzü yaratmadaki standartları tanımlayan, HTML'e benzer bir işaretleme dilidir. XAML, geliştiricilere XML kullanarak tüm .NET kullanıcı arayüz elemanlarını (buton, textbox vb..) kontrol edebilme yeteneği sağlar. XAML'in dezavantajı ise, Microsoft'un ürünü olduğu için sadece Microsoft'un bir işletim sisteminde kullanılabiliyor olmasıdır.
**MXML:** Macromedia'nın Flex teknolojisi içerisinde kullanılabilmesi için tasarladığı işaretleme dilidir. Flex, Macromedia firmasının çıkardığı MXML ve flash (swf) teknolojilerini kullanan ve zengin web uygulamaları hazırlamaya yarayan bir platformdur. MXML (Magic Extensible Markup Language), HTML gibi tasarlanmış, ActionScript ile birlikte zengin internet uygulamaları yaratabilmek için arayüz tasarlamaya yarayan bir dildir. ActionScript, Flash geliştiricilerinin sunum seviyesi mantığını tasarlamak için kullandıkları, ECMAScript üzerine inşa edilmiş bir betik dilidir. Pahalı geliştirme araçlarına ihtiyaç duyduğu için, MXML kullanımı ve

yaygınlığı bakımından Flash'la aynı kısıtlara sahiptir. .NET 'in gelecekte desteklemesi beklenen MXML, Flex'in günümüzde sadece Java 2 Enterprise Edition (J2EE) uygulama sunucularında (TomCat ve IBM'in Websphere'i gibi) çalışması nedeniyle tam olarak benimsenememiştir.

**XAMJ:** XAMJ, (Xml Application Markup Language for Java) Java'nın, XML türevli arayüz tasarımı dünyası için geliştirdiği yeni bir dildir. XAMJ, yakın geçmişte Web uygulaması geliştiricilerinin platformlar arası seçenekleri arasında yerini almıştır. Bu türev, nesne tabanlı dillerinden biri olan Java'nın bütün gücünü sağlayabilecek şekilde inşa edilmiştir. XAMJ temelde, belirli bir altyapıya (framework) bağımlı olmayan ya da yüksek internet bağlantı hızı istemeyen; böylece daha güvenli bir seçenek olarak görülen XAML ya da HTML tabanlı uygulamalara bir alternatif olarak ortaya çıkmıştır. XAMJ, "clientlet" mimarisi üzerine inşa edilen ve derlenen bir dildir; ve kendi başına çalışabilen uygulamaların geliştirilmesi mümkün olmasına rağmen, genel olarak XAMJ kökenli programlar Web tabanlıdır. Clientlet, XAMJ dökümanlarının dönüştürüldüğü özel bir tür Java sınıfıdır. Bu tez yazıldığı esnada, XAMJ gerçekçi bir şekilde eleştirilmek için fazlasıyla yeni olmasına rağmen takip edilmeyi haketmektedir.

**XFORMS:** XForms, W3C tarafından standartlaştırılmış, daha zengin kullanıcı arayüzleri tasarlamaya ve veri ile sunum katmanlarını birbirinden ayırmaya imkan sağlayan, klasik HTML form uygulamalarının XML'deki karşılığı olarak tanımlanabilir. XForms'un en önemli özelliklerinden biri, XML formatında ki form verilerinin sunucuya gönderilmesinden sonra üretilen verinin yine XML olmasıdır. Böylelikle bu XML formatında ki veri, diğer XML teknolojileriyle(XML Events, XML Schema, Xpath vb..) birlikte rahatlıkla kullanılabilir hale gelmektedir. XForms ile klasik HTML form uygulamalarının yapabildiklerinden daha fazlası gerçekleştirilebilmekte; form verilerinin doğruluğu betik bir dil (JavaScript gibi) kullanılmadan kontrol edilebilmekte, Web servisleriyle kolayca entegrasyon yapılabilmektedir. XForms ile sadece iş mantığı belirlenmekte, formun nasıl görüntüleneceği tamamen istemciye bırakılmakta bu da farklı işaretleme dilleri, uygulamalar, işletim sistemleri ve makinelerde kullanılabileceği anlamına gelmektedir.

# 4. ZENGİN İNTERNET UYGULAMASI NEDİR ?

Zengin İnternet Uygulamaları (Rich Internet Applications) terimi ilk olarak Macromedia firması tarafından 2002 yılında ortaya atılmıştır [3]. Bu konsept, aslında daha önceki yıllarda birçok adla anılmıştır;

- Remote Scripting (Microsoft tarafından 1998 yılında üretilen, Java Applet ve JavaScripti bir arada kullanan bir web uygulamaları geliştirme çözümü.)
- X Internet [2] (Executable Internet, Bağımsız, teknoloji ve pazar araştırma şirketi, Forrester Research tarafından, Ekim 2000)
- Rich (web) clients, (Zengin (web) istemcileri)
- Rich web applications, (Zengin web uygulamaları)

Zengin istemci arayüzleri oluşturmak, birçok açıdan bir Web sayfası tasarlamaktan farklıdır.

Geleneksel Web tabanlı uygulamalar, tüm aktivitilerini sunucu-istemci mimarisi etrafında şekillendirmişlerdir. Bu mimaride, tüm işlemler sunucu bilgisayar tarafında çalışan uygulamada sonlandıktan sonra, istemciye statik içerik HTML olarak gönderilmektedir. Bu doğrultuda, mimarinin en büyük dezavantajı, uygulama ile olan tüm etkileşimin, istenen verinin sunucuya bildirilmesi ve gelen cevabın istemci tarafında sayfa tamamen tekrar yüklendikten sonra gösterilebilmesidir. İstemci taraflı teknolojiler, tüm bu etkileşimin istemci bilgisayarında yapılmasını sağlamaktadır.

ZİU bu geleneksel mimariden kaynaklı yavaşlığın ve eş zamanlı döngülerin aşılmasını sağlamakta ve istemci ile olan etkileşimi kolaylaştırmaktadır.

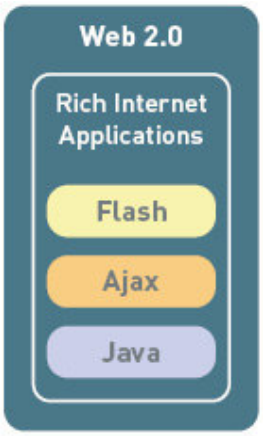
Web standartları, bu teknik ve yaklaşımları yavaş da olsa, sürekli olarak içine almakta ve geliştirmiktedir. Bu yüzden, zengin internet uygulamaları kavramını, neyin oluşturup, oluşturmadığını kesin olarak anlamak ve ayırmak kolaylıkla mümkün değildir.

Bütün zengin internet uygulamalarının taşıdığı karakteristik bir özellik vardır: çoğunlukla “istemci motor” (Client Engine) olarak adlandırılan ve istemci ile sunucu arasında çalışan kod içerisinde, ara bir katman kullanmak. Bu “istemci motoru” çoğunlukla uygulamanın başında istemci tarafından indirilir ve uygulama çalıştıkça da

yeni kod indirmeleriyle de desteklenebilir. İstemci motoru, tarayacının bir ek özelliği gibi davranır ve çoğunlukla da uygulamanın kullanıcı arayüzü ve sunucu arasında iletişim kurmasını üstlenir.

İstemci motoru, bunu eş zamanlı olarak değil, istemci tarafından tetiklenen bir olay neticesinde asenkron olarak yapabilmektedir.

Web'in hızlı gelişimi, birçok kavramın gerek uygulama gerekse iş geliştiriciler tarafında kafa karışıklığı yaratmasına neden olmuştur. Zengin internet uygulamaları kavramıda bu karmaşadan etkilenmiştir. Halen bir çok geliştirici için ZİU denildiğinde akla gelen sadece Ajax ya da Flash olmaktadır. Şekil 3.1'den de anlaşılabileceği gibi Ajax, ZİU'nun bir alt kümesi, Web2.0 ise ZİU'nun bir üst kümesini oluşturmaktadır [14].




Şekil 3.1 Web 2.0, ZİU, Ajax ve Flash'ın birbirleriyle olan ilişkisi diyagramı.

## 4.1. KULLANICI DENEYİMİ TASARIMI VE KULLANICI ODAKLI GELİŞTİRME

Kullanıcı deneyimi (user experience) kavramı, kullanıcının belirli bir proje ya da ürünle etkileşime geçtiği anda, sahip olduğu deneyimlerin tümünü ve bunların kalitesini ifade etmek için kullanılmaktadır.

Başlangıçta, insan-bilgisayar etkileşimlerine (Human-Computer Interactions) ilişkin bir kavram olarak kullanılan bu terim ki halen bu disiplinleride büyük ölçüde içinde bulundurmaktadır, şu anda her türlü insan-tasarım etkileşimlerini (Human-Design Interactions) (Örneğin, bir dijital cihazdan bir satış işlemine kadar olan herşeyi) belirtmektedir. "Kullanıcı Deneyimi", organizasyondan organizasyona farklılık

göstermekle beraber, bazı firmalarda pazarlama departmanının sorumluğunda bazılarında ise bilişim teknolojileri (IT) departmanının sorumluluğu altındadır.

Kullanıcı deneyimi tasarımı; kullanıcı odaklı tasarım (user-centered design), kullanışlılık ve kullanıcı arayüzü tasarım disiplinlerinin kökünü oluşturur. Fleming'e göre [6], bir kullanıcı deneyimi, ürünün farkında olmakla başlar, kullanıcının amacına ulaşmak için, uygulamayla ilgili olan tüm etkileşimini tüm açılarıyla içerir. En iyi kullanıcı deneyimleri, müşterinin ihtiyacına en kısa ve basit yoldan erişebilmesi ile sağlanmaktadır.

Garret'a göre [19], kullanıcı deneyimi geliştirme işlemi, kullanıcının ürünle olan tüm etkileşiminin tamamen site sahibinin bilgisi dahilinde gerçekleşmesiyle sağlanabilmektedir. Bunun anlamı, kullanıcının Web sitesi üzerinde gerçekleştirebileceği tüm hareketlerin ve her adımındaki olası beklentilerinin önceden saptanması demektir.

Kullanıcı deneyimi ile kullanıcı odaklı tasarım arasında çok ince ve anlamsal bir fark olmasına rağmen temelde iki kavram da aynı şeyi ifade etmektedir. Kullanıcı odaklı tasarım, kullanıcıyı uygulamanın merkezine koymakta, kullanıcı deneyimi tasarımı ise deneyimi vurgulamakta ve iş amaçları (business goals) ve kullanıcı ihtiyaçları arasındaki dengeyi göz önünde bulundurmaktadır. Kullanıcı odaklı geliştirme maliyetleri azaltır ve kullanıcı memnuniyeti ve verimliliği arttırır [20].

### 4.1.1 Kullanıcı Deneyimi Tasarım Metodolojisi

Kullanıcı deneyimi tasarımı işlemi; kullanıcının amaç tanımını, "akıllı" uygulama davranışı tasarımını, kullanıcı arayüzü tasarımını, sistem tasarımını ve bazı iş ve pazarlama konularını kapsamaktadır [19].

İdeal bir kullanıcı deneyimi tasarımı aşağıdaki adımları içermelidir:

**Gereksinim ve sınırlamaları saptamak**

- Uygulamanın ve yapılacak işin gereksinimlerinin belirlenmesi
- Zaman planı ve kapsamın çıkartılması
- Teknik ve tasarımsal sınırlamaların belirlenmesi

**Ayrıntılı kullanıcı analizi**

- Kullanıcı ve kullanıcıların amaçlarını tanımlamak

- Demografik araştırma hazırlıkları yapmak, pazarlama odak gruplarıyla toplantılar yapmak ve kullanıcılarla konuşmak (örneğin, demografik profilleri uygun kullanıcılarla)
- Her tipteki kullanıcı için bir profil yaratmak ve her profil için bir kullanıcı hikayesi yazmak

**Etkileşim ve navigasyon tasarımı**

- Kullanıcı amaçlarını, iş ve görevleri analiz etmek ve önceliklendirmek
- Kullanıcı iş akışını tanımlamak
- Kullanıcı ve uygulama arasındaki etkileşimi tanımlamak
- Bilgilendirme nesnelerini ve navigasyon akışını tanımlamak
- Bir davranışsal yönetmelik çıkarmak

**Kullanıcı arayüzü tasarımı**

- Uygulamanın kağıt üzerinde bir prototipini yaratmak
- Prototipi kullarak, kullanılabilirlik testini tasarlamak ve yürütmek
- Prototipi revize etmek ve kullanıcılarla üzerinden tekrar geçmek
- Uygulamanin detaylandırılmış bir kullanıcı arayüzünü geliştirmek
- Detaylandırılmış kullanıcı arayüzünü destekleyen bir arayüz yönetmeliği yazmak ve teknik gereksinim ve diger meseleleri tanimlamak

**Uygulama Geliştirme ve Test**

- Uygulamanın çalışan bir prototipini kullanarak kullanilabilirlik testinin yapılması
- Çalışan prototipi revize etmek ve kullanıcılarla üzerinden tekrar geçmek
- Uygulama geliştiricilerin yaptıklarını gözden geçirmek
- Uygulama geliştiricilerle birlikte test edip, uygulamayı teslim etmektir

## 4.2. ZENGİN İNTERNET UYGULAMALARI GELİŞTİRME ARAÇLARI

ZİU, olarak tanımlanabilecek uygulamalar halen geliştirme ve kullanıcı uyum sürecinin başındadırlar. Bu noktada, aşağıda da açıklanmış olan birçok kısıtlama ve gereksinim bulunmaktadır [7].

**Tarayıcı uyum süreci:** Birçok ZİU çalışmak için modern Web tarayıcılarına ihtiyaç duymaktadır. Zengin kullanıcı arayüzleri oluşturabilmek için, XMLHTTPRequest

(XHR) gibi istemci-sunucu iletişimini sağlayan nesnelerin tarayıcı tarafından desteklemesi gerekmektedir.

**Web standartları:** Tarayıcılardaki farklılıklar, ZİU yazımını zorlaştırmaktadır. Örneğin, Java platformunun tutarlılığı sayesinde bu sorun Java Applet'lerin kullanımı vasıtasıyla aşılabilmektedir.

**Geliştirme araçları:** ZİU oluşturmak için, bazı Ajax altyapıları ve entegre çözümler sunan bazı ürünler (Curl, Adobe Flex ve Microsoft Silverligt vb.) geliştirilmiştir.

**Erişim kısıtları:** Uygulama erişimindeki limitler, bazı teknik yaklaşımların ortaya çıkmasını gerektirmektedir.

**Kullanıcı uyum süreci:** Kullanıcılar, içinde tarayıcının tüm özelliklerini kullanabildikleri standart Web uygulamaları beklemektedirler. (Örneğin: ileri-geri butonu problemleri)

ZİU ile ilgili başlıca geliştirme yöntem ve araçları aşağıda incelenmiştir.

**JavaScript / Ajax :** İlk ve en çok kullanılan istemci taraflı dil ve teknoloji olan JavaScript, bilinen ve en çok kullanılan tarayıcı programlarla uyumlu olarak çalışmaktadır. Başlangıçta diğer istemci taraflı teknolojilere nazaran daha limitli olarak kullanılsa da, DHTML kullanımıyla birlikte, ZİU'nun vazgeçilmez bir parçası haline gelmiştir. Ajax tekniğinin kullanılmasıyla birlikte, JavaScript'in önemi ve kullanımı daha da artmış, fakat büyük ölçekteki uygulamaların hazırlanması, JavaScript ile kod yazımının zor olması, ve tarayıcı uyumluluğu gereksinimi gibi nedenlerle zor olduğundan, bu durum birçok, açık kaynak kodlu, hazır, Ajax altyapılarının ortaya çıkmasına neden olmuştur.

**Appcelerator :** Appcelerator, ZİU ve servis tabanlı mimarinin kullanımına imkan veren, entegre bir platformdur. ZİU, zengin Web 2.0 kullanıcı arayüzlerini, servis tabanlı mimari ise, ZİU tarafından kullanılan sunucu bazlı servisleri ifade etmektedir. Appcelerator, yeni nesil ZİU ve servis tabanlı mimari uygulamalarını yaratmayı sağlayan güçlü bir geliştirme platformdur. Uygulamaya, http://www.appcelerator.org/ adresinden erişilebilmektedir.

**OpenLaszlo :** OpenLaszlo, Laszlo Systems Inc. tarafından geliştirilen, açık kaynak kodlu bir ZİU geliştirme altyapısıdır. LZX programlama diliyle, OpenLaszlo sunucusuna bağlı bir platformudur. LZX, XML ve JavaScript karışımı bir dildir.

OpenLazslo ile birçok platform için (Flash, DHTML, J2ME) uygulama geliştirilebilmektedir [8]. Kaynak kodda yapılacak küçük bir değişiklikle uygulamanın tüm yaygın tarayacı ve işletim sistemlerinde çalışması sağlanabilmektedir.

**Microsoft Silverlight :** Windows Presentation Foundation'in (WPF) bir alt kümesi olarak ortaya çıkan Microsoft Silverlight, uygulama geliştiricilerin kolaylıkla, ZİU geliştirmelerini sağlayan bir platformdur. WPF, .NET Framework 3.0 'ın grafik altyapısını oluşturan XAML bir tabanlı sistemdir. XAML (Extensible Application Markup Language), Microsoft tarafından geliştirilmiş olan, değer ve nesnelerin ilk değerlerini vermek için kullanılan XML tabanlı bir dildir. WPF de olduğu gibi Silverlight'da XAML'i kullanır. Bu yüzden, önceki geliştirme tecrübelerinde .NET Framework 3.0 ve XAML kullanan uygulama geliştiriciler, Silverlight'ı daha tanıdık, çekici ve kolay kullanılabilir bulmaktadırlar.

**ActiveX Controls :** HTML içerisine Activex kontrolleri gömmek, ZİU geliştirmede çok etkili yollardan biridir. Activex kontrolleri, diğer Web tarayıcılarının Activex kontrol desteği olmaması nedeniyle, sadece Internet Explorer tarayıcısı üzerinde düzgün bir şekilde çalışmaktadır. Ayrıca, bilgisayar virüsleri ve Malmware'ler için potansiyel hedef olduklarından bu durum yüksek derecede güvenlik açığı yaratmaktadır. Malmware, bilgisayar sistemlerine zarar vermek için tasarlanan bir tür yazılımdır. Yalnızca, bir şirketin iç ağında standart tarayıcı olarak İnternet Explorer'ı (Microsoft firması tarafından geliştirilen Web tarayıcısı) tercih etmesi durumunda Activex kontrollerini kullanmak iyi bir seçim olarak gözükmektedir.

# 5. AJAX NEDİR ?

Ajax, Asenkron JavaScript ve XML kelimelerinin baş harflerinden oluşan ve Web tarayıcısının içinde çalışan, sunucuya bağlanıp çeşitli metodlarla bilgi alış verişi yapabilen, sonuçlar kullanıcıya ulaşana kadar durumununu takip eden ve en sonunda sunucudan alınan bilgileri kullanmayı sağlayan, bir tekniktir. Kullanıcıların Web sayfalarıyla olan etkileşiminde devrim yaratan bir yöntem olarak nitelendirilmektedir. JavaScript, internet tarayıcı uygulamalarının içerisinde gelen istemci taraflı (client-side) olarak yorumlanan bir dildir. Java dili ile ilişkili değildir. XML (Extensible Markup

Language-Genişletilebilir İşaretleme Dili) ise, verinin tanımlanması ve tutulması için kullanılan bir işaretleme dildir.

Ajax terimi, ilk olarak 2005 yılında, Adaptive Path firmasının kurucusu Jesse James Garret tarafından yazmış olduğu bir makalede kullanılmıştır [9]. Ajax tekniği önceleri, Dynamic HTML ve Remote Scripting adları ile tanınmaktaydı.

Garret'a [9] göre, Ajax tek başına geliştirilmiş bir teknoloji olmayıp aşağıda belirtilen teknoloji ve yöntemlerin birleşmesinden oluşmuştur;

- XHTML ve CSS (standartlara uygun gösterim için)
- DOM (dinamik ve etkileşimli görünümler için)
- XML ve XSLT (kurallı veri transferi için)
- XMLHTTPRequest nesnesi (asenkron veri erişimi için)
- JavaScript

XHTML, HTML'in kirli, dağınık yapısını ortadan kaldırmak için kullanılan standartlaşmış bir XML uygulamasıdır.

CSS (Cascading Style Sheets - Geçişli Stil Kağıtları), Web dökümanlarına stil (font, renk vs..) eklemek için kullanılan basit bir mekanizmadır.

DOM (Document Object Model - Belge Nesnesi Modeli), bir web tarayıcısının hiyerarşik nesne yapısını, özelliklerini ve metodlarını belirleyen bir haritadır.

XSLT (Extensible Stylesheet Language Transformations - Genişletilebilir Biçimlendirme Dili Dönüşümleri), XML dosyalarını farklı formattaki dosyalara çeviren bir biçimlendirme dilidir.

Tablo 2.1 de, tüm bu teknojiler ve Ajax tekniğini oluşturmadaki rolleri özetlenmiştir [10].

Tablo 2.1 : Ajax tekniğini oluşturan temel elemanlar ve açıklamaları.

| ***Eleman*** | **Açıklama** |
|---|---|
| Java Script | Javascript, web uygulamaları içerisine kolaylıkla gömülebilecek şekilde tasarlanmış olan, istemci taraflı, bir uygulama geliştirme dilidir. Ajax uygulamaları javascript kullanılarak yazılır. |
| Cascading Style Sheets | Web sayfasını oluşturan elementlerin, yeniden kullanılabilir ve değiştirilebilir, görsel stillere sahip olması için kullanılan basit |

| | |
|---|---|
| Document Object Model(DOM)(Belge Nesnesi Modeli) | ama güçlü bir yoldur. Ajax uygulamalarında, kullanıcı arayüzlerinin, etkileşimli olarak yeniden oluşturulması CSS kullanılarak sağlanmaktadır. DOM, web sayfasını oluşturan tüm elementlerin hiyerarşik yapısını ifade eden bir modeldir. Javascript tarafından bu elementlere kolayca ulaşılabilerek, modifiye edilebilmektedir. Ajax kullanarak, sayfanın herhangibir parçası anında ve etkin birşekilde DOM nesnelerine ulaşılarak değiştirilebilmektedir. |
| **XMLHttpRequest Nesnesi** | XHR nesnesi, web sunucusuna yapılan isteklerin ve dönen verinin alınabilmesinin, arka planda bir aktivite olarak yapılabilmesini sağlayan, nesnedir. Gönderilen ve alınan verinin formatı genelde XML'dir fakat XHR nesnesi, her türlü metin tabanlı veri üzerinde rahatlıkla çalışmaktadır. Ajax'ın temelini oluşturan sunucu-istemci aktiviteleri, bu nesne kullanılarak yapılmaktadır. |

Ajax tekniği, JavaScript kodları aracılığı ve XHR nesnesi yardımı ile bilgi transferini, takibini ve elde edilen bilgilerin kullanımına imkan sunmaktadır. Ajax, tarayıcı üzerinde çalışan bir JavaScript komut kümesinin, Web sayfası üzerinde meydana gelen olaylara (tıklama vs.) göre, sunucudan istekte bulunarak, sayfanın tamamının ya da bir kısmının gelen veriye göre değiştirilmesini sağlayan bir yöntemdir. Bunların yanında, DHTML, CSS ve HTML dillerini de kullanmaktadır.

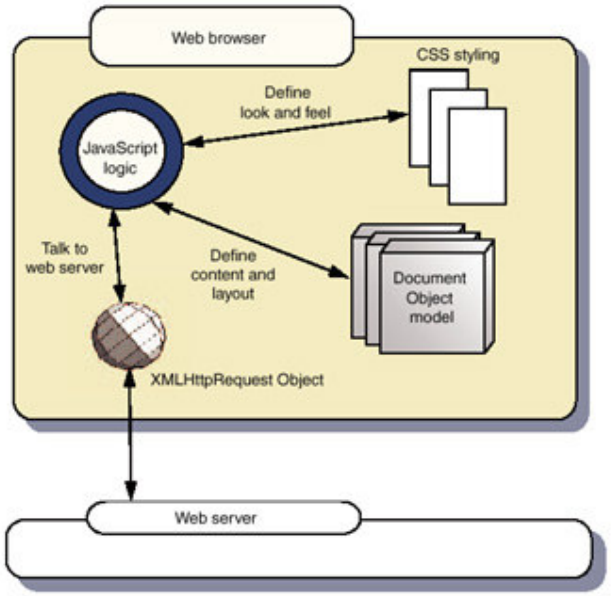
Ajax'ın dört önemli bileşeninin kullanımı şu şekilde gerçekleşmektedir:

JavaScript, işin kurallarını ve program akışını tanımlamakta, DOM ve CSS ise, XHR nesnesi yardımıyla, arka planda veriye erişimin sağlanmasında, sayfanın görünüşünün yeniden düzenlenmesinde kullanılmaktadır. Şekil 4.1'de Ajax'ın dört ana bileşenin çalışma mantığı gösterilmektedir [10].

Dört bileşenden üçü (CSS, DOM ve JavaScript) birlikte Dynamic HTML'i kısaca DHTML'i ifade etmektedir. DHTML, Web sayfaları için etkileşimli arayüzler oluşturmayı sağlamaktadır. Ajax'ın gücünü kullanarak, tüm sayfanın yeniden yüklenmesi gerekmeden, Web sunucusuna yapılan asenkron isteklerle, DHTML kullanılarak Web sayfalarının uzun ömürlü ve zengin görünümlü olmaları sağlanmıştır.

Ajax, gelişmiş, zengin, daha hızlı, daha dinamik ve eş zamanlı çalışan Web uygulamaları geliştirmek için kullanılan bir yaklaşımdır. Ajax tekniği, birçok Web teknolojisinin bir arada kullanılması ile Web sayfaları üzerinde, birbirinden özgür,

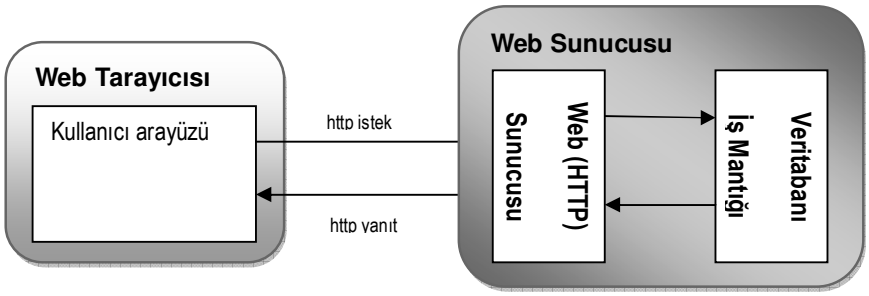
dinamik alanlar oluşturulmasına ve bu alanların ayrı ayrı takip edilebilmesini sağlamıştır.




Şekil 4.1: Ajax'ın dört ana bileşeni ve çalışma mantığı.

## 5.1. GELENEKSEL WEB TABANLI UYGULAMALAR İLE AJAX TEKNİĞİNİN KARŞILAŞTIRILMASI

Şekil 4.2'de de görüldüğü gibi [29], geleneksel Web tabanlı uygulamalarda, kullanıcı istekleri önce sunucuya ulaşmakta, sunucu tarafında çalışan uygulama vasıtasıyla, gelen isteğe bağlı olarak oluşturulan cevaplar istemciye gönderilmekte, bu da zaman ve hız kaybına neden olmaktadır [21].




Şekil 4.2: Klasik Web tabanlı uygulama modeli [29].

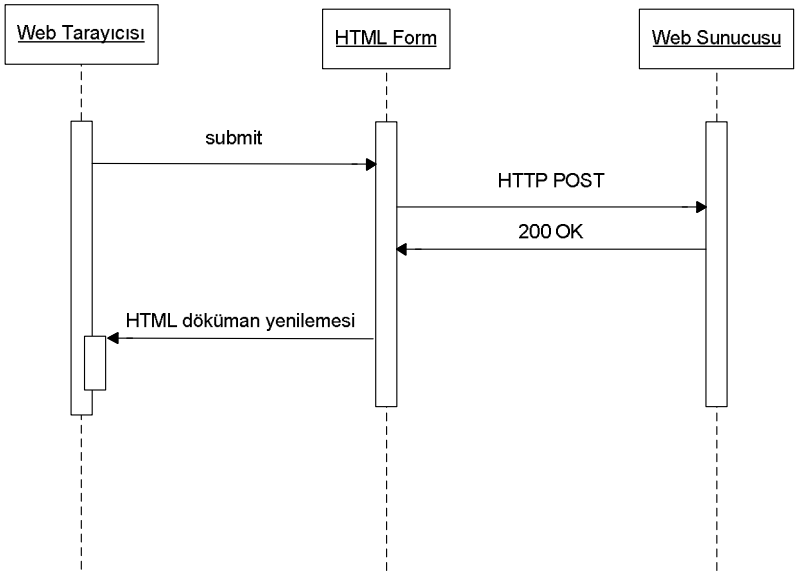
Daha basit bir örnekle anlatmak gerekirse, Web sayfaları içerisinde kullanılan HTML Form elemanları (anket, insan kaynakları formu gibi uygulamalar HTML form uygulamalarıdır.), kullanıcı tarafından girilen verileri, Web sunucusu üzerinde çalışan uygulamaya, çeşitli metodlarla gönderirler (post, get) ve sunucudan dönen sonucuda sayfanın tamamı yeniden yüklendikten sonra kullanıcıya yansıtabilmektedirler. Şekil 4.3 de gösterilen ardışıklık (sequence) diyagramında da bu durum anlatılmak istenmiştir [9][12].

Bu klasik yaklaşım ve uygulama uzun zamandan beri kullanılmaktadır. En belirgin dezavantajı, sayfanın istemci tarafında her veri gönderiminde tekrar tekrar yüklenmesi ve bunun büyük ölçüde zaman kaybına yol açmasıdır. Sunucu tarafında sayfa, istemciden gelen her talepte yeniden oluşturulmaktadır. Bir diğer dezavantajı ise, istek yapıldıktan ve sayfa yeniden yüklendikten sonra, tarayıcının, kaydırma ve imleç elemanlarının pozisyonları kaybolmakta, bu da kullanıcı tarafında işlemin tamamlanmasını, sayfanın neresiyle ilgilendiğiyle ilgili odağın kaybolmasına sebep olmaktadır [12].

Bazen Web uygulaması geliştiricilerin, bir sayfa içerisinde birden fazla form elemanı kullanması durumu söz konusu olabilir. Bu durumda, her bir formun ayrı ayrı gönderilmesi (submit) gerekmektedir çünkü gönderilen form dışındaki diğer formlara yapılan veri girişleri, sayfanın yeniden yüklenmesinden (postback) etkilenmekte, ve kullanıcı tarafından bu formlara girilen veriler kaybolmaktadır.

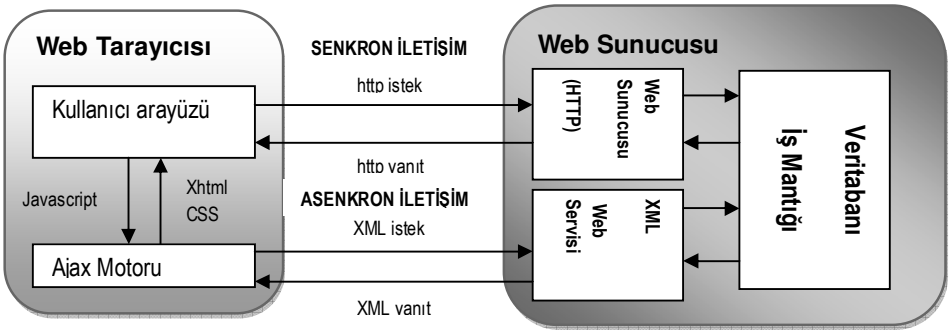
Ajax tekniği, klasik Web tabanlı uygulamalarda ki, *başla-dur* etkileşimini ortadan kaldırmayı hedeflemektedir. Bu amaçla sunucu ve istemci arasına *Ajax Motoru (istemci motoru)* denilen bir katman oluşturulmaktadır.

Kullanıcı, ilgili Web sayfasına giriş yaptığında, tarayıcı öncelikle sayfayı yüklemek yerine JavaScript ile yazılmış olan Ajax Motoru'nu yüklemektedir. Tarayıcıya yüklenen Ajax Motoru hem kullanıcının göreceği arayüzü oluşturmakta hem de kullanıcının sayfa üzerindeki hareketleri sonucunda oluşan olaylara göre, sunucuyla iletişim kurabilmektedir.




Şekil 4.3: Geleneksel modelde html form verisi gönderim ve alımı ardışıklık diyagramı.

Ajax tekniğinde, sunucu etkileşimi asenkron gerçekleşmektedir. Bu sebeple, istemci ve sunucu arasındaki *başla-dur* etkileşimi ortadan kalkmaktadır [12]. Klasik Web tabanlı uygulamalar senkron olarak çalışırken, Ajax uygulamaları asenkron olarak çalışmaktadırlar. Bu durum, Şekil 4.4'de gösterilmiştir.

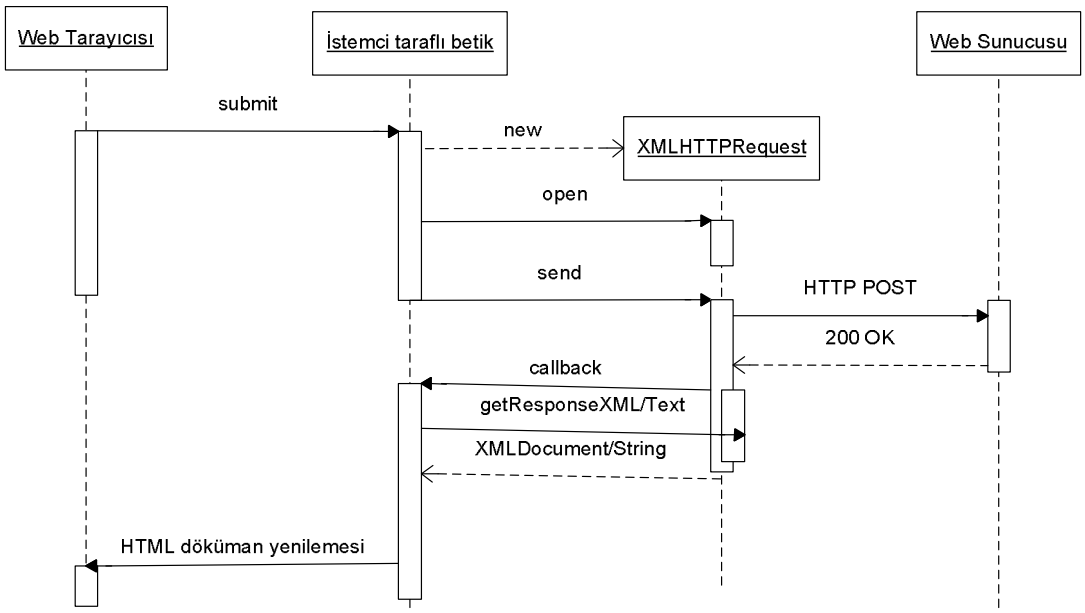



Şekil 4.4: Ajax uygulamalarındaki asenkron etkileşim modeli.

Ajax tekniği ile, yapılan her kullanıcı işlemi "Ajax Motoru" vasıtasıyla, bir HTTP isteği oluşturularak sunucuya gönderilir ve cevabı sunucundan alınır. Sunucunun döndüreceği cevaplar, sayfanın tarayıcı tarafında, gidip gelmesine neden olmaz, sunucudan dönen cevapları "Ajax Motoru" karşılar. Ajax Motoru sunucuya olan istekleri genellikle XML kullanarak ve kullanıcının arayüz ile olan etkileşimini kesmeden asenkron olarak yapar.

Web uygulamalarında XHR nesnesi aracılığı ile Ajax arabirimi oluşturulur. Daha sonra oluşturulan bu nesne aracılığıyla post, get vb. metodlar kullanılarak veriler web sunucusunda çalışan uygulamaya gönderilir ve gelen sonuçlar da bu nesnenin değişimi takip edilerek işlenir.

Ajax yaklaşımı ile birlikte, geleneksel form uygulamaları da, XHR nesnesi kullanılarak, asenkron bir şekilde sunucuya istek gönderebilecek duruma gelmişlerdir [11]. Bu da kullanıcı tarafında sayfanın yeniden yüklenmesini engellediği için zamandan tasarruf ve iyi bir kullanıcı deneyimi anlamına gelmektedir. Şekil 4.5 de yeralan ardışıklık diyagramında, Ajax yaklaşımlı form uygulamaları çalışma mantığı aktarılmıştır.




Şekil 4.5: Ajax yaklaşımlı form uygulaması ardışıklık diyagramı.

Özetlemek gerekirse; geleneksel Web sitelerinde, istemci sayfanın tamamı için, Web tarayıcısını kullanarak sunucuya bir istekte bulunur, sonrasında kullanıcı sayfa içerisinde herhangibir linke tıkladığında yada form verisi göndermek gibi başka bir aksiyon gerçekleştirdiğinde, sunucuya yeni bir istekte bulunulur ve sunucu farklı bir sayfa göndererek bu isteği karşılar.

Ajax tekniği ise; kullanıcı sayfayla etkileşime geçtiğinde (bir linke tıkladığında, form verisi gönderdiğinde vs.) sayfanın tamamının bu istek sonrasında etkilenmemesini,

sayfada talep edilen verinin, sayfanın ilgili parçasında güncellenmiş olarak sunulmasını sağlamaktadır [12].

## 5.2. ÇEVRİMİÇİ VE ÇEVRİMDIŞI DENEYİMLERİN BİRLEŞTİRİLMESİ

Kısa bir süre öncesine kadar, çevrimdışı çalışan masaüstü uygulama kullanımı ile tarayıcı tabanlı Web uygulamaları kullanımı arasında oldukça önemli farklar vardı: etkileşimli, zengin grafik kullanıcı arayüzleri, "sürükle ve bırak" uygulamaları, Web uygulamalarında sıklıkla karşılaşılan kavramlar değildi. Bu farklar, tarayıcıların izin verdiği ölçüde geliştirilen ZİU ile kısa zamanda kapanmıştır.

Çevrimiçi ve çevrimdışı deneyimlerin birleştirilmesi, büyük fırsatlar yarattı ve günümüzün Ajax benzeri teknikler ile desteklenen, tarayıcı tabanlı Web uygulamaları, kullanıcı etkileşimlerini, sürükle ve bırak uygulamalarını ve zengin araçları içinde barındıran Web uygulamalarının ortaya çıkmasını sağladı. Sonuç olarak, kullanıcı memnuniyeti ve kullanıcı deneyimlerinin kaliteleri artmıştır [13].

## 5.3. XMLHTTPREQUEST NESNESİ

XHR nesnesi, Ajax terimi ortaya atıldığından bu yana isminden en çok söz edilen, istemcinin, sunucuyla olan etkileşimini sağlayan, güçlü ve kullanışlı bir nesnedir. Jesse James Garrett [9], bu nesneyi zihninde düşünürken, Ajax kavramını ortaya atarak literatüre girmesini sağlamıştır..

Bu nesne, ilk olarak Microsoft tarafından, Exchange Mail Sunucusunun Web arayüzü olan Outlook Web Access 2000 (OWA) için geliştirilen bir Activex nesnesi olarak tasarlanmıştır. Microsoft tarafından XMLHTTP olarak adlandırılan nesne, istemci tarafında, Internet Explorer Web tarayıcısının 5.0 sürümünde, JavaScript ya da VBScript kullanılarak erişilebilen bir nesne özelliği taşımıştır.

Bu noktada, henüz, Web istemcileri açısından standart bir özellik olmamasına karşın birçok önemli, çok kullanılan Web tarayıcısı, nesneyi geliştirdikleri tarayıcılarına dahil etmişlerdir. (Mozilla 1.0 in 2002, Safari 1.2, Konqueror, Opera 8.0 etc..)

World Wide Web Consortium (W3C) tarafından henüz Web standartlarına konulmamasına rağmen taslak dökümanları oluşturulmaya başlanmış ve nesne ile ilgili standartlaşmaya gidilmiştir.

İlgili dökümanlara, http://www.w3.org/TR/XMLHttpRequest adresinden erişilebilir.

XHR nesnesi AJAX yaklaşımının temelini oluşturmaktadır. XHR nesnesi, sunucuya bir HTTP isteği göndermek için istemci tarafındaki JavaScript koduna bir yol sağlar. Yani Web uygulamalarında bir JavaScript nesnesi olan XHR nesnesi aracılığı ile Ajax birimi oluşturulur. Daha sonra oluşturulan bu nesne aracılığıyla çeşitli yordamlar kullanılarak bilgiler sunucu tarafında çalışan uygulamaya gönderilir ve gelen sonuçlar da bu nesnenin değişimi takip edilerek işlenir [15].

### 5.3.1 Yordam ve Özellikleri

XMLHttpRequest nesnesi için tanımlanmış olan yordamlar aşağıda yeralan tablo 4.1'de gösterilmiştir.

Tablo 4.1 : XMLHTTPRequest nesnesi yordamları.

| ***Yordam*** | **Açıklama** |
|---|---|
| abort() | Geçerli isteği iptal eder. |
| getAllResponseHeaders() | Tüm HTTP başlık etiketlerini ve değerlerini "string" olarak elde eder. |
| getResponseHeader( "Başlık Etiketi" ) | Verilen HTTP başlık etiketinin taşıdığı değeri "string" olarak elde eder.. |
| open( yöntem, adres ) | İstek gönderilmeden önce, isteğin gönderileceği adresi, gönderi yöntemini ve diğer seçime bağlı nitelikleri belirler. |
| open( yöntem, adres, eş zamanlılık ) | |
| open( method, URL, async, userName ) | Gönderi yöntemi "GET", "POST", "HEAD", "PUT", "DELETE", yada W3C standartlarında yeralan diğer HTTP istek yöntemlerinden biri olabilir. |
| open( method, URL, async, userName, password ) | |
| | Eş zamanlılık parametresi isteğin eş zamanlı bir şekilde mi, yoksa sunucudan gelen cevabı beklemeden işleme devam edileceğini ele alır. "true" script in işleme send() yordamı çağırıldıktan sonra, sunucudan dönecek cevabı beklemeden, devam edeceğini, "false" ise script'in sunucudan gelen cevabı bekledikten sonra işleme devam edeceğini belirler. |
| send( "içerik" ) | İsteği gönderir. Seçime bağlı olarak gönderilebilen verileri ve DOM(Document Object Model) ile yaratılmış verileri de gönderebilir. |

| | |
|---|---|
| **setRequestHeader( “etiket”, “değer”)** | Gönderilen isteğe bir etiket/değer başlığını iliştirir. |

XMLHttpRequest nesnesinin özellikleri ise tablo 4.2’de açıklamalarıyla beraber gösterilmiştir.

Tablo 4.2 : XMLHTTPRequest nesnesi özellikleri.

| ***Özellik*** | Açıklama |
|---|---|
| Onreadystatechange | İstekle ilgili durum değişikliğini kontrol eder |
| ReadyState | Tam sayı olarak istek durum bilgisini belirtir:<br>0 = uninitialized (başlatılmadı)<br>1 = loading (yükleniyor)<br>2 = loaded (yüklendi)<br>3 = interactive (etkileşimli)<br>4 = complete (tamamlandı) |
| ResponseText | Dönen veriyi "string" biçiminde alır. |
| ResponseXML | Dönen veriyi W3C standartlarında XML olarak alır. |
| Status | İstek yapılan sunucunun durumunun sayısal değerini belirtir:<br>404: Not found (Bulunamadı)<br>200: OK (Tamam) |
| StatusText | İstek yapılan sunucunun durumuyla ilgili text değeri getirir.(”OK”, “NOT FOUND” etc..) |

### 5.3.2 XMLHTTPRequest Nesnesinin Ajax Yaklaşımı İçerisindeki Kullanımı

Aşağıdaki JavaScript fonksiyonları, XHR nesnesinin, Ajax yaklaşımı içerisinde, destekleyen tarayıcılarda oluşturulması ve kullanılmasını örneklemektedir.

```
<div id="ajax_ciktisi">
        İçeriğin yapılan istek sonrası güncellenmesi bekleniyor..
</div>

<script type="text/javascript">
 var xmlHttp = null; // xmlHttp adında bir değişken tanımlanır.
 // Farklı tarayıcılar için doğru nesnenin yaratılması sağlanır.
 try {
   // Firefox, Opera 8.0+, Safari, IE7+
```

```
   xmlHttp = new XMLHttpRequest(); // xmlHttp değişkeni artık XMLHttpRequest nesnesidir.
 } catch (e) {
   // Internet Explorer tarayıcısı için
   try {
     xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
   } catch (e) {
     xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
   }
 }
 xmlHttp.onreadystatechange = function() {
   if (xmlHttp.readyState == 4) // Sunucuya yapılan isteğin, tamamlanıp/tamamlanmadığına bakılır
     try { // istek yapılan sunucunun durumuna bakılır
       if (xmlHttp.status == 200) {
         // istek sonrası sunucundan dönen cevap, sayfa(DOM) içerisindeki ilgili nesneye yerleştirilir.
           document.getElementById("ajax_output").innerHTML
             = xmlHttp.responseText;
       }
     } catch (e) {
       document.getElementById("ajax_output").innerHTML
         = "Hata durumu: " + e.description; // Sayfa içerisindeki nesneye hatanın açıklaması basılır.
     }

 }
 var reqUrl="adres.html"; // İsteğin yapılacağı adres, reqUrl değişkeninde tutulur.
 xmlHttp.open("get",reqUrl);
 xmlHttp.send(null); // İstek sunucuya gönderilir(gönderi yöntemi "POST" olması durumunda null yerine ilgili veriler gelecektir.)
</script>
```

Örnekte yeralan adres.html dosyasına yapılan istek sonucu dönen veri, "ajax_ciktisi" belirteçli div nesnesi içerisine yerleştirilmektedir. İstek durumunu takip edebilmek için XHR nesnesinin onreadystatechange olayından faydanılmakta, readyState özelliği "Tamamlandı" yani "4" değerini döndürdüğünde ve isteğin yapıldığı sunucudan "HTTP 200" dönüşü alındığında, XHR'ın responseText özelliğiyle, ilgili div'e dönen veri gömülmektedir [15].

## 5.4 ÖRNEK KULLANIM VE GELENEKSEL YÖNTEMLERLE KARŞILAŞTIRMA

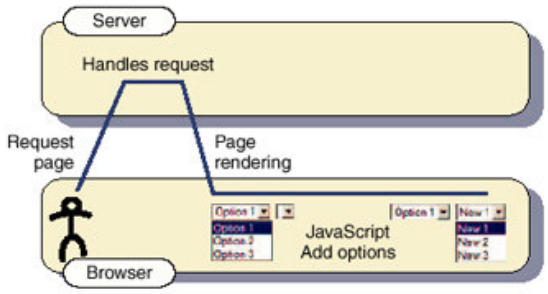
Bu bölümde, Ajax ve diğer geleneksel yöntemleri karşılaştırmak amacı ile, basit örnekler hazırlanarak, geleneksel yaklaşımlarla olan karşılaştırılması yapılmakta, avantaj ve dezavantajları üzerinde durulmaktadır.

### 5.4.1 İkili seçim listesi uygulaması (Double-Combo Script) :

Bu örnekte hazırlanacak olan Web uygulamasının birbirine bağlı iki seçim listesinden oluşması, ilk seçim listesinden seçilen seçeneğe bağlı olarak güncel listelenin ikinci seçim listesine doldurulması beklenmektedir.

**- İstemci taraflı çözüm ve limitleri**

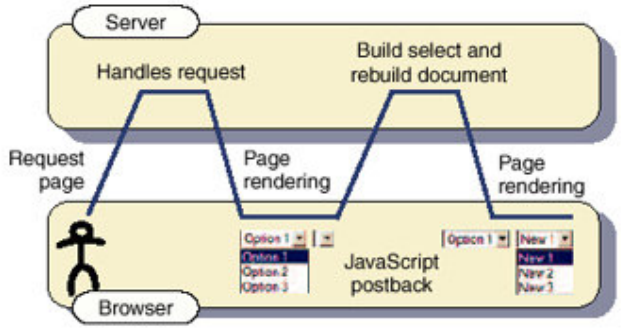
Bu uygulamanın hazırlanmasındaki ilk seçenek, uygulama geliştiriciler için tamamen istemci taraflı yaklaşımın kullanılması olabilmektedir. İstemci taraflı yaklaşım ve JavaScript kullanılarak dinamik ve değişken seçim listeleri hazırlanmak istendiğinde, tüm listenin, JavaScript dizisi kullanılarak kodun içerisine statik olarak yazılması gerekmektedir. Bu durumda, herhangibir sunucu taraflı bağlantı olmadığından seçim listeleri içerisindeki verilerin güncel olması sağlanamamaktadır. Şekil 4.6'da da görüldüğü gibi, sayfa ilk yüklemesini tamamladığında, ikinci seçim listesinde yeralacak tüm veriler önceden JavaScript dizisi içerisine yerleştirildiğinden, birinci seçim listesi seçildiği anda dizinin içerisindeki ilgili kayıtlar anında istemci tarafında görüntülenebilmektedir. Diğer taraftan, böylesine basit bir uygulamada verinin güncelliği de talep edilemediğinden, JavaScript kullanmanın tek yararı kodun ilk yüklemedeki zaman kaybını en aza indirgemek ve istemcinin istediği veriye en kısa zamanda ulaşabilmesini sağlamak olmaktadır [10].




Şekil 4.6: İkili seçim listesi uygulaması istemci taraflı çözüm.

**- Sunucu taraflı çözüm ve limitleri**

Diğer bir geleneksel çözümde, sunucuya ilgili form verilerinin gönderilmesidir. "Page Postback" olarak adlandırılan bu yöntemde, ilk seçim listesininin *onchange* adlı olayı tetiklenerek, seçili veri sunucuya *submit* methodu vasıtasıyla gönderilmekte, sunucu tarafında çalışan uygulama, istemci tarafından gönderilen seçimi alarak, veritabanında önceden hazırlanmış olan ilgili kayıtlar içerisinde bir sorgu çalıştırmakta, ve cevap olarak dönen sonuçlar, ikinci seçim listesi içerisine şekil 4.7'de gösterildiği gibi, dinamik olarak yerleştirilebilmektedir.



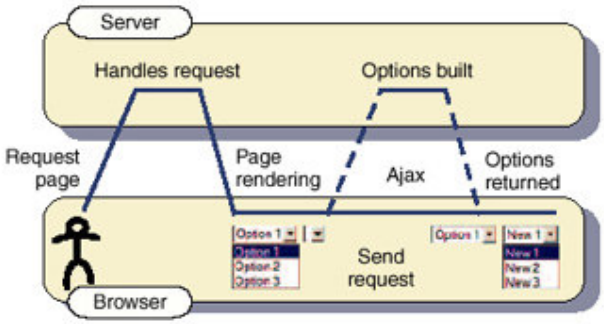

Şekil 4.7: İkili seçim listesi uygulaması sunucu taraflı çözüm.

Bu yöntemin dezavantajı ise, birinci seçim listesinin seçilmesinin herseferinde sunucu tarafına yapılan gidiş-dönüşler meydana getirmesi, dolayısiyle, tüm sayfanın sunucu tarafında tekrar tekrar oluşturulması anlamına gelmesidir. Bunun yanısıra, sunucu üzerinde fazladan trafik oluşturmak ve kullanıcının isteğine zamanında cevap verememekte önemli birer dezavantajdır..

**- Ajax temelli çözüm**

İstemci ve sunucu taraflı yaklaşımların dezavantajlarından kurtulmak için, birinci seçim listesinden yapılan seçim sonrası, Ajax yaklaşımı kullanılarak, sadece talep edilen veri için sunucu tarafına istekte bulunulabilmektedir. Sonuç sayfanın tamamı yeniden yüklenmeden, ilgili ikinci seçim listesine doldurulabilmektedir. İstemci taraflı çözüm ile karşılaştırıldığında, ikinci seçim listesinin oluşturulabilmesi için, tüm ve güncel verinin istemci tarafında ki dizide tutulmasından kaynaklı sayfa yüklemelerinde ki zaman kaybı böylelikle giderilmekte, sunucu taraflı çözüm ile karşılaştırıldığında da, tüm sayfanın her seferinde yeniden yüklenmesinden oluşan, sunucu taraflı trafik ve zaman kaybının önüne geçilmiş olunmaktadır. Ajax yaklaşımında, tüm sayfa içerisinde, sadece istenen

veri için sunucu tarafına istekte bulunulmakta ve sadece sayfanın ilgili bölümü bu isteğin sonucundan etkilenmektedir.




Şekil 4.8: İkili seçim listesi uygulaması ajax temelli çözüm.

Şekil 4.8' de de görüldüğü gibi;

Kullanıcı, ilk seçim listesini kullanarak bir seçim yaptığında, istemci, sunucu tarafına seçilmiş olan kaydı gönderir, sadece kendisini ilgilendiren bilgiyi talep eder ve dönen liste, ikinci seçim listesine dinamik olarak doldurulmaktadır.

Tablo 4.3'de yapılan örnekle ilgili avantaj ve dezavantajlar listelenmiştir.

Tablo 4.3 : İkili seçim listesi örneğinde kullanılan üç yaklaşımın karşılaştırılması.

| ***Metod*** | ***Avantajı*** | **Dezavantajı** |
|---|---|---|
| Seçim listesine değerlerin statik olarak yerleştirilmesi | Sunucu taraflı işlem gerektirmez. | Seçenekler dinamik olarak güncellenemez. |
| Değerlerin seçim listesine sunucu taraflı script kullanılarak doldurulması. | Seçenekler, veritabanı kullanılarak, dinamik olarak güncellenebilir. | Sunucu tarafında fazladan işlem yapılmasına yol açar. |
| **Değerlerin seçim listesine doldurulması için ajax yönteminin kullanılması.** | Sadece, istenen veri için sunucuya gidilir, tüm sayfa yeniden yüklenmez. | Sunucu tarafında fazladan kod yazılmasını gerektirir. |

## 5.5. AJAX ALTYAPI VE ARAÇ TAKIMLARI

Altyapı (framework), yazılım geliştirmeye yardımcı olmak için hazırlanmış kütüphaneler olarak tanımlanabilir. Örnek olarak, bir geliştiricinin en çok ihtiyaç duyduğu veritabanı erişimleri için, tekrar tekrar kod yazmak yerine hazırlanacak bir

altyapı sayesinde hızlı ve kolay bir şekilde bu basit işlemi yapabilmesi sağlanabilmektedir. Bir yazılım altyapısı hazır kod kütüphanelerini, bir betik dili ya da geliştirmeye yardımcı olan ya da bir yazılım projesinin farklı bileşenlerini birbirine bağlan yazılımları içerebilir.

Ajax geliştirmelerinde kullanılabilecek bazı popüler altyapılar aşağıda açıklamalarıyla beraber listelenmiştir.

**Adobe Spry Framework :** Adobe firması tarafından, Ajax için geliştirilmiş olan Spry framework, Web tasarımcıları için, kullanıcılarına daha zengin bir deneyim yaşayacakları ortamlar hazırlamalarına yardımcı olabilen çok fonksiyonel bir JavaScript kütüphanesidir. Adobe, Spry Framework ile Ajax'la geliştirme yapan kullanıcılar için Adobe Labs Web adresinde ( http://labs.adobe.com/technologies/spry/ ) etkileyici içerik ve dökümantasyon desteği sunmaktadır.

**Prototype:** Prototype, dinamik web uygulamaları geliştirmeyi kolaylaştırmak amacıyla oluşturulmuş bir JavaScript frameworktür. Javascript ile, nesne-sınıf tabanlı uygulama geliştirmek için araç takımı (toolkit) ve zengin bir Ajax kütüphanesini içinde barındırmaktadır. Birçok firma, Protoype aracını Web tabanlı uygulamalarında kullanmaktadır. (cnn.com, nbc.com, NASA.gov, apple.com vs.. ) Framework'ün en son sürüm ve dökümantasyonlarına http://www.prototypejs.org/ adresinden ulaşılabilmektedir.

**Dojo Toolkit**: Toolkit (araç takımı), uygulama geliştirmeyi kolaylaştıran araçların bulunduğu settir. JavaScript'i destekleyen, dinamik Web uygulamaları ya da başka zengin ortamlar hazırlamayı kolaylaştıran açık kaynak kodlu bir araç takımıdır. http://www.dojotoolkit.org adresinden ulaşılabilmektedir [16]. **Zephyr:** PHP5, uygulama geliştiricileri için bir ajax tabanlı, öğrenmesi kolay, uygulaması basit ama güçlü bir altyapıdır. http://zephyr-php.sourceforge.net adresinden erişilebilmektedir.

**Google Web Toolkit :** GWT, Java programcıları için, Ajax uygulamarı yaratmada kullanılan kullanışlı ve açık kaynak kodlu bir altyapıdır. Java, Sun tarafından geliştirilmeye başlanmış açık kodlu, nesneye yönelik, platformdan bağımsız, yüksek performanslı, çok işlevli, yüksek seviye, adım adım işletilen (interpreted) bir dildir. Programcı, Java üzerinde uygulamayı hazırladıktan sonra GWT'nin araçları sayesinde, uygulamanın ihtiyaç duyduğu JavaScript kodları otomatik olarak üretilmektedir. [17]

GWT, bedava indirilebilmekte ve üzerinde sunucu-istemci taraflı uygulamalar hazırlanarak, altyapı ile birlikte dağıtılabilmektedir.

## 5.6 AJAX'IN KULLANIM ALANLARI, AVANTAJLARI VE DEZAVANTAJLARI

Ajax, özellikle etkileşim amaçlı kullanılan, ancak standart olmayan ve ek olarak gereksiz bant genişliği (bandwidth) harcatan, Flash ve Java yerine daha az kaynak tüketen, tasarruflu bir yaklaşımdır. Bant genişliği, iletişim kanalının kapasitesini belirler; bant genişliği ne kadar büyükse, belli bir süre içinde aktarılabilecek verinin hacmi de o kadar büyük olur. Ajax, istemci tarafından, sadece talep edilen veriler gidip geldiğinden, bant genişliği tasarrufu sağlanmaktadır.

Ajax bilinen tüm sunucu tabanlı Web programlama dilleriyle uyumlu çalışabilmektedir. (Php, ColdFusion, ASP, ASP.NET vb.. ) Ajaxla hazırlanmış, oldukça etkileyici ve zengin kullanıcı deneyimi içeren Flickr, Gmail, Google Maps gibi uygulamalar Ajaxın kullanım oranını ve ününü arttırmaktadır.

Flickr, http://www.filckr.com adresinde erişilebilen fotoğraf ve resim paylaşım sitesi, Gmail, http://www.gmail.com adresinden erişilebilen mail hesabı yaratma ve kullanım sitesi, Google Maps ise, http://maps.google.com adresinden erişilebilen bir harita uygulamasıdır.

Tüm bunların yanısıra Ajax'ın, Web 2.0 kavramının ünlenmesinde itici bir rolü olmuştur [18].

Web tabanlı uygulama geliştiriciler tarafından, hali hazırdaki teknolojilerin birarada kullanılabilmesi nedeniyle, öğrenmesi ve uygulaması kolay bir tekniktir. Hazır ve açık kaynak kodlu gelişmiş kütüphaneler sayesinden uygulama geliştiriciler kısa sürede mevcut uygulamalarında Ajax tekniğini kullanabilmektedirler.

### 5.6.1 KULLANIM ALANLARI

Ajax;

- Etkileşimli ve değişken yapılı, karmaşık ve denetimli form ve anket uygulamalarında,
- Hiyerarşik yapılar, sadece ihtiyaç duyulan veriyi çekme ve sunucuyu yormama prensibini kullanan uygulamalarda,

- Sürekli yenileme ve takip gerektiren, borsa, piyasa ve oy takibi benzeri uygulamalarda,
- Girdi ve içerik filtreleme mekanizmalı uygulamalarda eş zamanlı kontrollerin yapılmasında,
- İpuçları ve yardım metinlerinin, mevcut içeriğe göre dinamik olarak güncellenmesi ve gösterilmesinin sağlanmasında,
- Anlık mesajlaşma uygulamalarında,
- Harita uygulamalarında (google maps),
- Her harf girişinde sonuçların ilgili şekilde güncelleneceği, aktif arama ve otomatik tamamlama uygulamalarında (google suggest),
- Bir alanı değiştirildiğinde, ilgili ek alanların çıkmasını sağlayan etkileşimli form uygulamalarında, (ülke-şehir ilişkisisi gibi)

kullanılabilmektedir.

### 5.6.2 AVANTAJ VE DEZAVANTAJLARI

Her yeni nesil teknolojide olduğu gibi Ajax kullanımının da bazı avantaj ve dezavantajları mevcuttur.

**Avantajlar;**

- HTTP talepleri arka planda gidip geldiği için sayfa yeniden yüklenmeleri engellenir, bu durum hem ziyaretçi hem de sunucu tarafında gereksiz bant genişliği kullanımının azalmasını sağlar.
- Masaüstü uygulamalarında bulunan zengin etkileşimin, arka planda çalışan Ajax motoru sayesinde kullanıcılara yaşatılması sağlanmaktadır.
- Ajax, birçok Web tasarımcısının ve uygulama geliştiricisinin bildiği mevcut teknolojilerden oluşmaktadır. Bu durum geliştiricilerin Ajax'ı kısa zamanda öğrenmelerine ve etkin bir şekilde kullanmalarına imkan sağlamaktadır.
- AJAX, yokluğunda ortaya çıkmış olan ve karmaşık web uygulamaları için tasarlanan JAVA ve Flash benzeri, her ortamda standart olarak çalışmayan, tarayıcıları yoran ve trafik harcatan teknolojilerin yerine, daha yüksek ve gerçek zamanlı etkileşim sunmakta, daha az kaynak tüketmekte ve uygulama zamanı olarak çok daha az zaman almaktadır.

- Sunum ve çalışma biçimi açısından hiçbir zaman tam bir standart olamayan, IFRAME ve benzeri nesnelerin kullanımından doğan problemlerin ortadan kalkmasını sağlamıştır.
- AJAX, CSS, DOM, XHTML ve JavaScript benzeri teknolojilerin kullanımını gerektirdiği için, bu teknolojilerin ve standartlarının yayılmasına ve gelişmesine de katkıda bulunmaktadır.
- Veri güvenliği nedeniyle yapılan, kullanıcı tabanlı JavaScript denetimleri yerine sunucu tabanlı denetimler hızlıca ve basitçe sayfa yenilenmeden, kullanıcıya zaman kaybı yaşatmadan yapılabilmektedir.
- Dezavantajlar kısmında da bahsedilecek olan; durdur, ileri ve geri butonlarının çalışmaması, aslında bazı dinamik uygulamalarda sorunlara yol açan bu tarayıcı işlevlerinin çalışmaması için geliştiriciler tarafından yapılan ek çalışmaları ortadan kaldırmaktadır. Örneğin, kullanıcının geri veya ileri butonunu kullarak, oturum (session) kullanılan dinamik uygulama sayfalarında gezinmesi, oturumda tutulan verilerin tutarlılığı açısından istenmeyen bir durumdur ve geliştiriciler bu durum için tarayıcının sağladığı bu işlevlerin olmadığı popup pencereleri kullanılmaktadır.
- Ajax uygulamalarının dinamik içerik üretme özelliklerinden dolayı, dezavantaj olarak görülen, üretilen içeriğin arama motorları tarafından taranamaması durumu, arama motorlarının ulaşması istenmeyen sayfalarda ek kod yazımı gereksinimini ortadan kaldırdığından, bazı durumlarda avantaj olarak görülebilmektedir.

**Dezavantajlar;**

- Veriler ve sayfalar daha çok kullanıcı tarafında işlendiği için, zaten yüksek miktarlarda işlemci ve bellek kullanan tarayıcılar daha yüksek miktarlarda işlemci ve bellek ihtiyacı duymaktadır. Bu da sınırlı imkanları ve yetenekleri olan bilgisayarlara sahip kullanıcıların Ajax uygulamalarına sahip siteleri gezerken problem yaşamaları anlamına gelmektedir.
- AJAX, Web tarayıcısı fonksiyonlarından durdur, geri ve ileri (Stop/Back/Forward) butonlarını kullanılamaz hale getirmektedir. Geri ve ileri butonları sayfanın tekrar yüklenmesi yani normal HTTP başlıkları ve talepleri gönderilmesi ile işlevsel hale gelmektedir. Ancak XHR nesnesi ile gönderilen

talepler geri ve ileri butonlarına etki etmediğinden bu butonları kullanmaya alışan tüm kullanıcılar ulaşmaya umdukları sayfaya gidemeyeceklerdir. Bu da doğal olarak sitenin iyi bir kullanıcı deneyimi yaratamamasına sebebiyet verecektir. (Bu problemle ilgili yazılım tarafında yapılabilecek birçok çözüm bulunmaktadır fakat bu çözümlerin çoğu istemci tarafında daha fazla kaynak tüketimi anlamına gelmektedir.)

- Bağlantıların ve sayfa içeriklerinin dinamik ve eşzamanlı üretilmesi aynı zamanda bu site içeriklerinin arama motorları tarafından taranamaması anlamına gelmektedir. Bu sorun, kayıt altına alınamayan içerik ve sayfalara, kullanıcılar tarafından erişimi zorlaştırmaktadır.
- İnternet Explorer tarayıcısında XHR nesnesi bir ActiveX nesnesidir. ActiveX bazı firma ve kullanıcı bilgisayarlarında varsayılan güvenlik ayarlarında engellenebilmektedir. Bu da ilgili bilgisayarlarda, AJAX uygulaması çalıştırılamaması anlamına gelmektedir.
- Kullanılabilirlik (usability) ve erişilebilirlik (accesibility) kurallarına ve standartlarına göre hazırlanması gereken Web uygulamalarının, hazırlandığı teknolojilerin varolmadığı tarayıcı türlerinde ve ya özelliklerinde çalıştırılabilmesi için uyumluluk çalışmaları yapılması gerekmektedir. Bu da fazladan zaman, emek ve maddi kaynak harcanması anlamına gelmektedir.
- Klasik web uygulamalarında kullanıcı profili ile ilgili veri toplamak, büyük ölçüde kullanıcıya bağlı bir durumdur. Ancak AJAX ile kullanıcılar farkında olmadan haklarında, özellikle Web sayfalarının üzerinde gerçekleştirdikleri hareketlerle ilgili oldukça fazla bilgi toplanabilme olanağı doğmuştur. Bu durum, Web dünyası çalışanları için, kullanıcı davranışlarına yönelik içerik gösterimi açısından yararlı olmaktadır. Kullanıcılar açısından ise davranışlarının takip edilmesi dezavantaj olarak görülebilmektedir.

## 5.7 AJAX YAKLAŞIMINI KULLANAN ÖRNEK UYGULAMALAR

Ajax'ın ününü ve kullanımını arttıran piyasadaki bazı örnek uygulamalar ve içerikleri aşağıda aktarılmıştır.

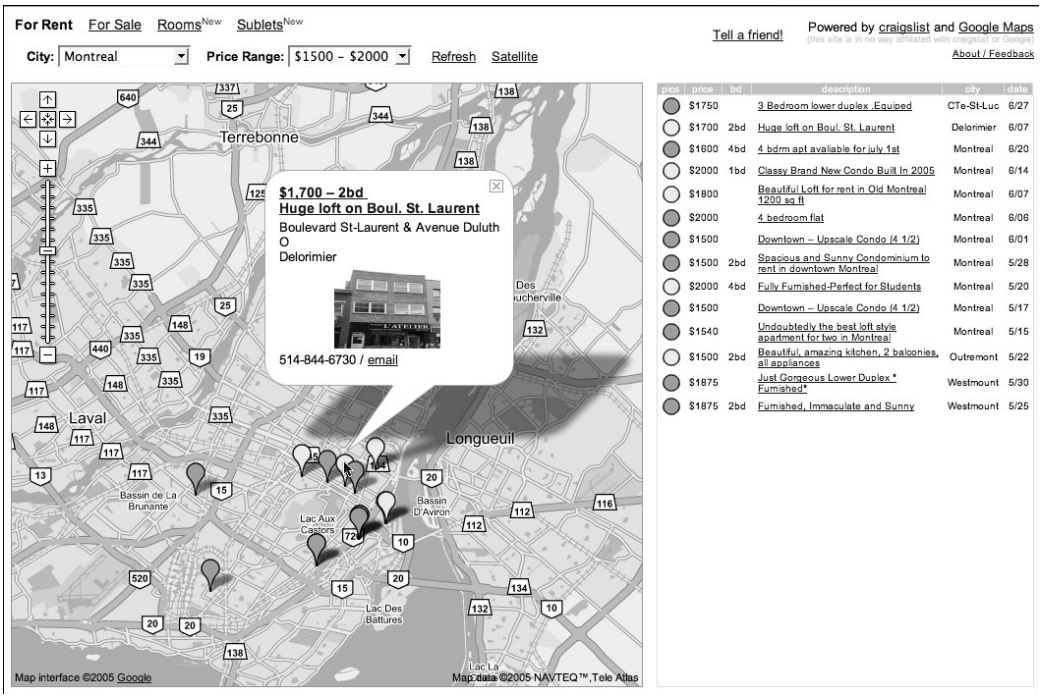
### 5.7.1 Google Maps

Google firması tarafından geliştirilmiş olan, Web üzerinde çalışan ve harita gösterimiyle arama motoru teknolojilerini entegre ederek, belirli bir lokasyonun bulunmasını ve görsel olarak kullanıcıya gösterilmesini sağlayan uygulamadır. Arama özellikleri ve fonksiyonları klasik Web uygulamalarında olduğu gibi sayfanın tamamının yeniden yüklenmesini gerektirse de, harita özelliği tamamiyle Ajax yaklaşımını kullanmaktadır. Kullanıcılar, sadece fare kullanarak tüm bir haritayı sürükleyerek, istedikleri lokasyonu bulabilmektedirler. Harita, küçük imajlardan oluşmaktadır. Kullanıcının harita üzerinde yapmış olduğu her bir hareket (yakınlaştırma, uzaklaştırma, kaydırma vs.) sonrası yeni imaj parçası, asenkron olarak yüklenmektedir. Şekil 5.1'de Google Maps uygulamasından bir görüntü verilmiştir.




Şekil 5.1: Goggle Maps uygulamasından bir görüntü.

Ayrıca Google, geliştirmiş olduğu harita uygulamasının API'ler vasıtasıyla farklı proje ve uygulamalar tarafından bağımsız olarak kullanılmasına da izin vermektedir. Örneğin, ABD'de bulunan bir emlak firması, kiralık ve satılık ev ilanlarını Google'ın harita servisini kullanarak işaretleyip, müşterilerine kendi sitesi (http://www.housingmaps.com) üzerinden kolaylıkla gösterebilmektedir. Bu yaratıcı fikir, zengin bir kullanıcı deneyimi yaratmakta ve örnek bir Web 2.0 uygulaması olarak değerlendirilmektedir [22]. Şekil 5.2'de bu uygulamayı gerçekleştiren Web sayfasından bir görüntü verilmiştir.




Şekil 5.2: Google Maps ile kiralık ev ilanları uygulamasından bir görüntü.

### 5.7.2 Google Suggest

Google'ın tipik arama motoru arayüzüne benzeyen bu uygulama, işlevselliğiyle dikkat çekmektedir (http://www.google.com/webhp?complete=1&hl=en adresinden ulaşılabilir.). Kullanıcının arama alanına, ilgili kelimenin ilk harfini girmesiyle birlikte arka planda çalışan Ajax istemci motoru, sunucu tarafında arama sonuçlarının indekslerinin bulunduğu uygulamaya bir sorgu yaparak, dönen sonuçları kullanıcıya aramasını tamamlayabilmesi için öneri olarak sunmaktadır. Harfler girildikçe otomatik olarak oluşan listeden kullanıcı kendi arama kelimelerinin bulunduğu seti seçebilmekte, arama motorunun ortalama kaç sonuç döndüreceği bilgisini görmekte ve aramaya

devam edebilmektedir [21]. Şekil 5.3'te Google Suggest uygulamasından bir ekran görüntüsü verilmiştir.



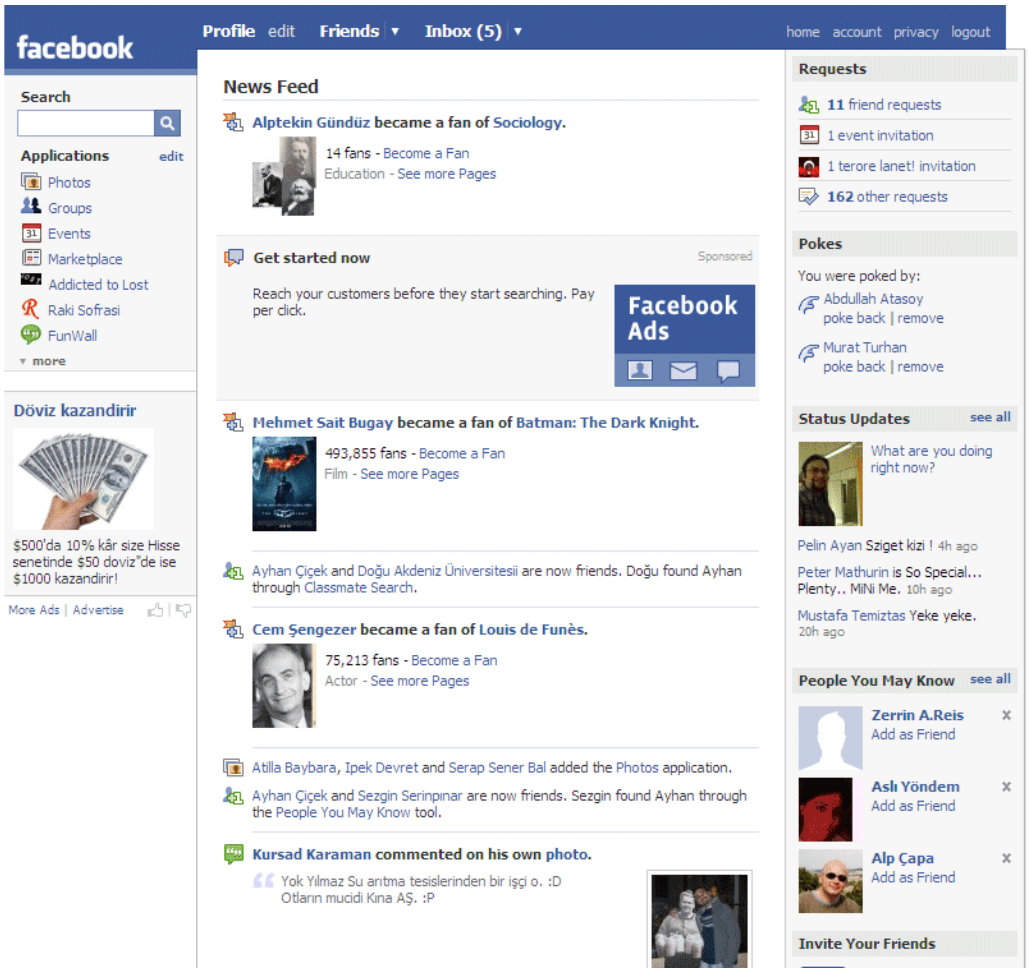

Şekil 5.3: Google Suggest uygulamasından bir görüntü.

### 5.7.3 Flickr

Bilinen en popüler dijital fotoğraf paylaşım sitesi ve bir çevrimiçi topluluk platfromu (online community platform) olan Flickr, uygulama mimarisi içerisinde Ajax yaklaşımını büyük ölçüde kullanmaktadır. Siteye yüklenen fotoğraflar, fotoğraf sahibinin seçimine göre, tüm hakları yükleyenin kendisinde kalacak şekilde ya da Creative Commons lisansıyla yayınlanabilmektedir. Creative Commons, kâr amacı gütmeyen, telif hakları alanında esneklik ve paylaşımı yaygınlaştırmak amacıyla kurulmuş bir düşünce hareketi ve organizasyonudur. http://creativecommons.org/ adresinden erişilebilir. Sistemin foto albüm oluşturma, arşivleme, filtreleme, arama, kullanıcı grubu oluşturma gibi fonksiyonları bulunmaktadır. Ajax tekniği sayesinde, zengin bir kullanıcı deneyimi sağlayan flickr sitesi, masaüstü fotoğraf yönetim uygulamalarının sağladığı tüm arayüz özelliklerinin fazlasıyla gerçekleştirilebildiği çevrimiçi bir platformdur. Bugün iki milyondan fazla kayıtlı kullanıcıya hizmet vermektedir ve şu an Yahoo firmasının bir parçası olan Flickr aynı zamanda Web 2.0 konseptli Web sitelerinin ilklerindendir [23].

### 5.7.4 Facebook

Geniş ölçekli bir sosyal ağ Web sitesi (social networking website) olarak tanımlanan facebook, kullanıcıların birbirleriyle iletişim kurmalarını ve bilgi alışverişinde bulunmalarını sağlamaktadır. Kullanıcılar ağa girişlerinde oluşturdukları profil bilgileri vasıtasıyla diğer kullanıcılarla iletişim kurabilmektedirler. Sitenin özellikleri arasında yeralan birçok uygulamada Ajax yaklaşımı kullanılmış bu da sitenin kullanışlılığını arttırmış ve daha fazla kullanıcıya ulaşmasında etkin rol oynamıştır. Harvard üniversitesi sınıf öğrencisi Mark Zuckerberg tarafından 2006 yılında kurulan facebook, 2007 yılında Microsoft tarafından %1,6 hissesi 240 milyon dolara satın alınmış bir firma haline gelmiştir [28]. Alexa istatistiklerine göre facebook Ekim 2007 itibarıyla; Dünya'nın en fazla ziyaret edilen yedinci Web sitesidir. Alexa, Web sitelerinin Web trafikleriyle ilgili bilgi veren bir istatistik firmasıdır. Alexa istatistikleri "Alexa Toolbar" adlı yazılımın kullanıcılardan topladığı verilerden derlenir.




Şekil 5.4: Facebook uygulamasından bir görüntü.

## 5.8 AJAX VE GÜVENLİK

Ajax, bir Web uygulamasının kullanışlılığını büyük ölçüde arttırsa da, uygulamanın geliştirici tarafından yanlış tasarlanması durumunda, saldırganlar için büyük fırsatlar ortaya çıkabilmektedir [24]. Aşağıda bu açıklardan ve saldırı tiplerinden genel olarak bahsedilmektedir.

### 5.8.1 XMLHTTPRequest Açıkları

Ajax, sunucu tarafında çalışan uygulamayla (sıklıkla bir web servis) olan tüm etkileşimler için XMLHTTPRequest (XHR) nesnesini kullanmaktadır. İstemci, belirli bir isteği, belirli bir URL adresine gönderir ve bir takım geri dönüşler alır. Bu geri dönen veri sıklıkla bir HTML parçası, XML, Javascript Object Notation (JSON), imaj ya da JavaScript'in işleyebileceği farklı bir format olabilir. JSON, basit ve kullanışlı veri tutuş formatıdır. XML'in Javascript ile kullanılmasının zor olmasından dolayı tercih edilmektedir, klasik DOM ağaçlarıyla aynı yapıya sahiptir. Bir Ajax sayfasına erişim protokolünün SSL (Secure Socket Layer – Güvenli Yuva Katmanı) olmadığı durumlarda, XHR sunucuya olan tüm talepleri şifrelenmemiş bir şekilde gerçekleştirir. Buna kullanılan sisteme giriş için gerekli olan kullanıcı bilgilerinin (login data) açık bir şekilde gitmesi de dahil olabilir.

SSL, internet üzerinden şifrelenmiş güvenli veri iletişimi sağlayan bir protokoldür. HTTPS/SSL kullanımı bu tarz açıklardan doğan saldırıları kapatacak nitelikte şifreleme imkanı sunan ve modern tarayıcıların desteklediği bir yöntemdir. XHR nesnesi, web üzerindeki tüm sunucuların bilgilerine erişme imkanı sunar, bu da farklı tipteki birçok saldırıya (SQL Injection, Cross Site Scripting (XSS) vb.) olanak sağlar.

### 5.8.2 Arttırılmış Saldırı Alanı

Ajax uygulamaları istemci-sunucu arasında çalıştıklarından, bu durum istemciye bazı güçler vererek uygulamaları saldırıya açık hale getirmektedir [25]. Geliştiriciler sınırlı işlevsellik gösteren sunucu taraflı bazı sayfalar yaratabilmekta ve bu durum saldırı yapacak biri için artırılmış bir saldırı alanı sağlayabilmektedir. Uygulamanın geri kalanı gibi bu sayfaların dikkatlice güvenlik kontrollerinin yapılması gerekmektedir. Bir başka kaygı verici alan da hatalarla başa çıkmadır (error handling). Geleneksel bir çok Web uygulamasında bu gözden kaçırılan bir alandır ve bir sonraki uygulamayı yaratmak için acele edilirken hatalarla başaçıkma zaman bakımından baskı altında olan uygulama

geliştiriciler tarafından ihmal edilebilmektedir [24]. Çok bir saldırı alternatifi olduğu için bir uygulamanın hata vermemesi için titizlikle çalışılması gerekmektedir.

### 5.8.3 SQL Injection

SQL injection, bir uygulamanın veritabanı katmanında gerçekleştirilen saldırı tekniğidir. Saldırgan veritabanına uygulama üzerinden erişerek veri güvenliğini tehdit edebilir. Genelde bu uygulama, Web sayfalarına tarayıcı üzerinden erişilirken URL'e yazılan bazı parametreler yardımıyla hata verdirilmesi, sonrasında hatadan yola çıkarak veritabanına ulaşılması ve yine sayfa üzerinden sql komutu çalıştırılması neticesinde oluşmaktadır.

Sql injection yapılarak sayfa üzerinden çalıştırılabilecek bir sorgu ile tüm bir veritabanı silinebildiği gibi, veritabanının çalıştığı makine üzerinde herhangi bir komut kümesinin de işletilmesi mümkündür.

### 5.8.4 Çapraz Site Betik Saldırıları (Cross-site Scripting Attacks)

XHR nesnesinin çapraz alan adı, çapraz port ya da çapraz protokol talepleri yapmasına izin verilmemiştir. Bu gelişi güzel saldırıları sınırlandırmasına rağmen, aynı zamanda çapraz site uygulamalarının gelişiminde de zorluklar çıkarabilmektedir.

XHR nesnesinin kullanımı ve aynı orijinal güvenlik politikası kısıtlamaları yüzünden ortaya çıkan zorluklar için birkaç potansiyel çözüm vardır;

Bunlardan en basit olanı, aynı alan adında barındırılan URL'i çağırmaktır, ancak bloke edilmiş faklı bir alan adından istenilen bilgiyi ayrıştırmak icin betik dillerden de faydalınılabilmektedir. Ancak bu tekniğin kullanımı uygulanan diger herhangi bir betikten daha az göze alınabilir derecede risksiz degildir. En iyisi çapraz alan adı betiklerinden kaçınmaktır.

Çabalayan gelistirmeciler için mekanizmalar sadece bunlarla sınırlı değildir. Flash kullanılarak da, XHR nesnelerinin sınırlarını aşabilmek ve çapraz alan adı problemi yaşamamak mümkündür. XHR nesnelerinin sağlamlıklarını aşan sadece gelistiriciler degildir, saldırganlar bu durumdan faydalanmak için bircok yol bulmuşlardır. Opera ve Mozillada olan bir cok hata kullanıcı geri dönüşleri sayesinde kapatılabilmiştir. Tarayıcı ve Ajax istekleri sunucu için benzerdir, bu yüzden sunucu bu istekleri sınıflandıramaz ve arka planda kimin istekte bulunduğunu anlayamaz. Bir JavaScript uygulaması,

kullanıcının bilgisi dahilinde olmadan arka planda çalışarak bir kaynağa istekte bulunabilir.

### 5.8.5 Hizmetin Reddi Saldırıları (Denial Of Service Attack- DoS)

Bilinen en eski saldırı tekniklerindendir. Saldırgan, servis veren web uygulamasına anlık olarak yüklü miktarda istekte bulunarak hizmetin sunucu tarafından karşılanmamasına ve kapanmasına neden olabilir. DoS saldırılarından korunabilmek için uygulama geliştiricilerin, geliştirdikleri uygulamanın limitini biliyor olması gerekir. Web sunucu üzerindeki ayarlamalar bilinen bu limitler doğrultusunda yapıldığında, saldırgana verilen hizmet istenilen anda sonlandırılabilir ve diğer kullanıcıların bu saldırıdan etkilenmemesi sağlanabilir. Ayrıca, yazılım ve donanım olarakda, ağ seviyesinde çalışan sunucu taraflı DoS saldırısını önleyen birçok çözüm bulunmaktadır.

### 5.8.6 Ajax Köprüleme

Güvenlik nedenleriyle, Ajax uygulamaları sadece sunuldukları internet sitesi ile bağlantı kurabilirler. Örneğin, yahoo.com adresinden indirilen Ajax içerisindeki JavaScript fonksiyonları, google.com adresine bağlantı kuramazlar. Ajax'ın bu şekilde 3. parti sitelere bağlantı kurabilmesi için, Ajax servis köprüsü inşa edilmiştir. Bir köprü içerisinde, bir sunucu, istemci tarafında çalışan JavaScript ile 3. parti site arasındaki trafiği yönlendiren bir proxy şeklinde çalışan bir web servisi bulundurur. Köprü bu bağlamda, "web servis" – "web servis" bağlantısı şeklinde düşünülebilir. Bir saldırgan, izni olmayan bilgilere ulaşmak için köprü özelliği kullanabilir.

### 5.8.7 Tarayıcı Tabanlı Saldırılar

Birçok tarayıcı tabanlı saldırı tipi bulunmaktadır, bu saldırı tipleri tarayıcılarda ki güvenlik açıklarından yararlanırlar. Örnek olarak, tarayıcı üzerinden gerçekleştirilebilecek iç ağ saldırıları verilebilir. İç ağda çalıştırılan JavaScript nesneleri ve komutları ağ üzerinde tarama yaparak web sunucularına ya da yazıcı, IP telefonu gibi ağ üzerinde çalışan başka cihaz ya da web arayüzü bulunan bir uygulamaya da erişebilirler. JavaScript tarayıcı uygulaması imaj nesnesini kullanarak, IP adreslerine ping atabilir ve dönen hata mesajlarını analiz edebilir niteliktedir.

Düzensiz hazırlanan Web uygulamaları ve Web tarayıcıları, saldırganlara bir çok açık kapı sağlayabilir. Örneğin saldırgan tarayıcı uygulamadaki açıklardan faydalanarak,

işletim sistemine ve dolayısiyle istemci bilgisayarlardaki dosyalara erişebilir ve bunları yönetebilir.

### 5.8.8 Sonuç

Ajax'ın genel anlamda geleneksel Web uygulamalarında yaşanabilecek güvenlik problemlerinden daha farklı bir açığı yoktur. Ancak, özellikle Ajax'la birlikte kullanılarak yapılan çapraz-site betiklerine dikkat edilmesi gerekmektedir. Çapraz-site betik kullanımı Ajax ile birlikte artmıştır ve bu durum ciddi güvenlik açıkları yaratmaktadır.

Bir Web uygulaması geliştirilirken, ister geleneksel yaklaşımda ister Ajax tekniği kullanılmış olsun dikkat edilmesi gerekenler aşağıda sıralanmıştır:

- Uygulama için de izin verilecek girdilerin dökümante edilmesi;
- Girdilerin geçerliliğinin işlenmeden önce kontrol edilmesi; (özellikle XRR saldırıları için önemli),
- Web uygulamalarının ve web uygulamalarına erişimin mümkün olduğu kadar Secure Socket Layer (SSL) protokolü kullanılarak sağlanmasıdır.

Güvenlik, uygulama geliştiriciler tarafından genelde ikinci plana atılan bir durumdur. Bu nedenle, birçok uygulama geliştirme platformu ya da güvenlik yazılımları, yukarıda bahsedilen birçok saldırıyı engeller nitelikte tasarlanmış ve geliştiricileri yönlendirir özelliktedir.

## 5.9 AJAX VE WEB 2.0

2004 yılında O' Reilly Media firması tarafından yapılan beyin fırtınası konferansı [18], Web 2.0 kavramının ortaya çıkmasına neden olmuş ve internet dünyasında yeni bir akım doğmuştur. İkinci nesil internet olarak adlandırılan Web 2.0 internet anlayışını geliştirerek, internetten beklentileri değişik bir platforma yönlendirmiştir. Kelimenin tam anlamı tartışmaya açıktır. Tim O'Reilly'e göre Web 2.0'ın kısmen tanımı şöyledir: "Web 2.0 bilgisayar endüstrisinde internetin bir platform olarak ilerlemesiyle oluşan bir

işletme devrimi ve bu yeni platformun kurallarını başarı için anlamaya çalışmaktır. Bu kurallar arasında başlıcası şudur: Ağ etkilerini daha çok insanın kullanabilmesi için uygulamalar yaratmak" [31]. Web 2.0 bir yazılım ya da programlama dili değil bir konsepttir. İnternete yeni bir bakış açısı kazandırmak için üretilmiş bir sürü teknik ve teknolojinin kullanılmasını amaçlar. Web 2.0 teknikleriyle yapılan siteler, kullanıcı odaklı yaklaşımın ürünüdür. Kullanıcının katılımıyla zengin bilgi kaynakları oluşturmak, Web 2.0 ın ana fikrini oluşturmaktadır.

Web 2.0 web siteleri tipik olarak aşağıda belirtilen özellik ve teknikleri içerebilir:

- Cascading Style Sheets (Biçim Şablonları) kullanımı.
- Kullanıcılar tarafından konulan etiketler ile oluşan bir sınıflandırma sistemi olan Folksonomilerin (sosyal sınıflandırma, indeksleme ve etiketleme olarak da bilinir) detaylı kullanımı.
- XHTML kullanımı.
- ZİU teknikleri (Ajax, Flex, Flash vb.) kullanımı.
- Bir sitenin içeriğiyle özdeş bir veriyi, uygulama geliştirme arayüzleri vasıtasıyla başka bir deneyim ile birleştirerek daha değerli bir içerik elde edilmesini sağlayan melez bir Web uygulaması olan; Mashup kullanımı.
- Bloglar.
- Wiki yada farklı forum uygulamaları.
- İçerik beslemeleri (RSS, Atom) kullanımı.

Web 2.0 kavramı ortaya atıldığı ilk zamanlarda birçok yanlış anlaşılmalara neden olmuştur. Örneğin, bazı görsel tasarım elemanları Web 2.0 konseptiyle özdeşleşmiş (köşeleri yuvarlatılmış kutular, altalanı gölgelendirilmiş imajlar vb.) ve yaygın olarak kullanılmaya başlanmıştır. Birçok kullanıcı bu tasarım elemanlarını kullanan siteleri Web 2.0 sitesi olarak nitelemiştir. Bir diğer ortak yanlış anlama ise; Web 2.0 sadece Ajax'dır şeklinde olanıdır. Bu hata muhtemelen yeni nesil birçok Web sitesinin Ajax temelli olmasından ve DHTML efektleri kullanmasından kaynaklanmaktadır.

ZİU geliştirme teknikleri arasında yeralan ve bir programlama dili olmayıp, yaklaşım olan Ajax'ın Web 2.0 için önemi büyüktür. Ajax için, Web 2.0'ın en çok heyecan uyandıran ve aslında temel bileşini olan yaklaşımdır denilebilir. Kullanıcıların, tarayıcı vasıtasıyla sunucuya gönderdikleri isteklerin daha hızlı şekilde yanıtlanması, sunucunun daha az meşgül edilmesi ve böylelikle daha fazla kullanıcıya hizmet eder hale gelmesi,

Web 2.0 konsepti içerisinde Ajax'ın en çok kullanılan teknik olmasını sağlamıştır. Google, Microsoft, Apple gibi önemli şirketler kısa zamanda Web sitelerini Ajax tabanlı hale getirmişlerdir [9].

Web 2.0 toplumsal iletişim sistemleri kurmayı amaçlamaktadır. Dolayısı ile geniş kitlelerin internet uygulamalarını kullanması Web 2.0'ın önemli amaçları arasındadır. Bunun için klasik Web geliştirme yaklaşımları yetersiz kalmıştır ve kullanıcının katılımını amaçlayan yenilikçi, özgürlükçü, kullanışlı sayfalar oluşturabilmek Ajax gibi tekniklerin kullanımını da beraberinde getirmiştir.

# 6. ÖRNEK BİR UYGULAMA:ÖĞRENC-İM

## 6.1 AMAÇ

Bu çalışma kapsamında geliştirilen Öğrenc-im uygulamasının temel amacı akademisyenler ve öğrenciler arasında anlık olarak iletişimin sağlanabilmesidir. Uygulamanın adı içerisinde yeralan "im" harfleri de ingilizce "instant messaging" (anlık mesajlaşma) kelimelerinin barş harfleridir. Öğrenc-im başlangıç olarak prototip şeklinde geliştirilmiştir. Uygulama, yeni eklentiler yapılabilecek, farklı sistemlere entegre olabilecek şekilde tasarlanmıştır. Yeni geliştirme talepleri kullanıcı deneyimleri sonucunda ortaya çıkacaktır.

## 6.2 TEORİ VE METODOLOJİ

Bu uygulama sayesinde, bir üniversitenin/fakültenin veya bölümün web sitesine giren öğrenciler o anda çevrimiçi olan akademisyen ya da öğrenci işleri yetkilileriyle anlık olarak görüşebilecek ve istedikleri konu hakkında bilgi alabileceklerdir.

Böyle olmakla birlikte, Öğrenc-im sadece bir anlık mesajlaşma uygulaması değildir; teorik olarak, mevcut üniversite veritabanlarına kolaylıkla entegre edilerek çalışabilecektir. Bu sayede, herhangi bir üyelik sistemine ihtiyaç duyulmayacak; öğrenciler sisteme öğrenci numaralarıyla giriş yapabilecek ve kendi bölümlerine bağlı akademisyenlerle anlık olarak iletişime geçebileceklerdir.

Öğrenc-im, iki bölümde incelenebilir: operatör (yetkili) modülü ve öğrenci modülü.

### 6.2.1 Operator (yetkili) modülü

Operatör modülü, akademisyen ve öğrenci işleri yetkililerine, kendilerine gelen anlık mesajlaşma taleplerini giriş yaptıkları ekranlardan görebilmeyi ve mesaj iletişimini başlatabilmeyi sağlamaktadır. Bu modül, kendilerine talepte bulunan tüm öğrencilerin aşağıda yer alan genel bilgilerine ulaşabilme imkanı da vermektedir:

1. Adı ve Soyadı
2. Öğrenci Numarası
3. E-posta adresi
4. Üniversite
5. Fakülte
6. Bölüm

Listelenen bu bilgiler veritabanından alınabilmekte ya da uygulamanın kullanımı sırasında oluşabilecek taleplere göre genişletilebilecek formattadır.

Operatör ekranları akışı Şekil 6.1'de gösterilmiştir.

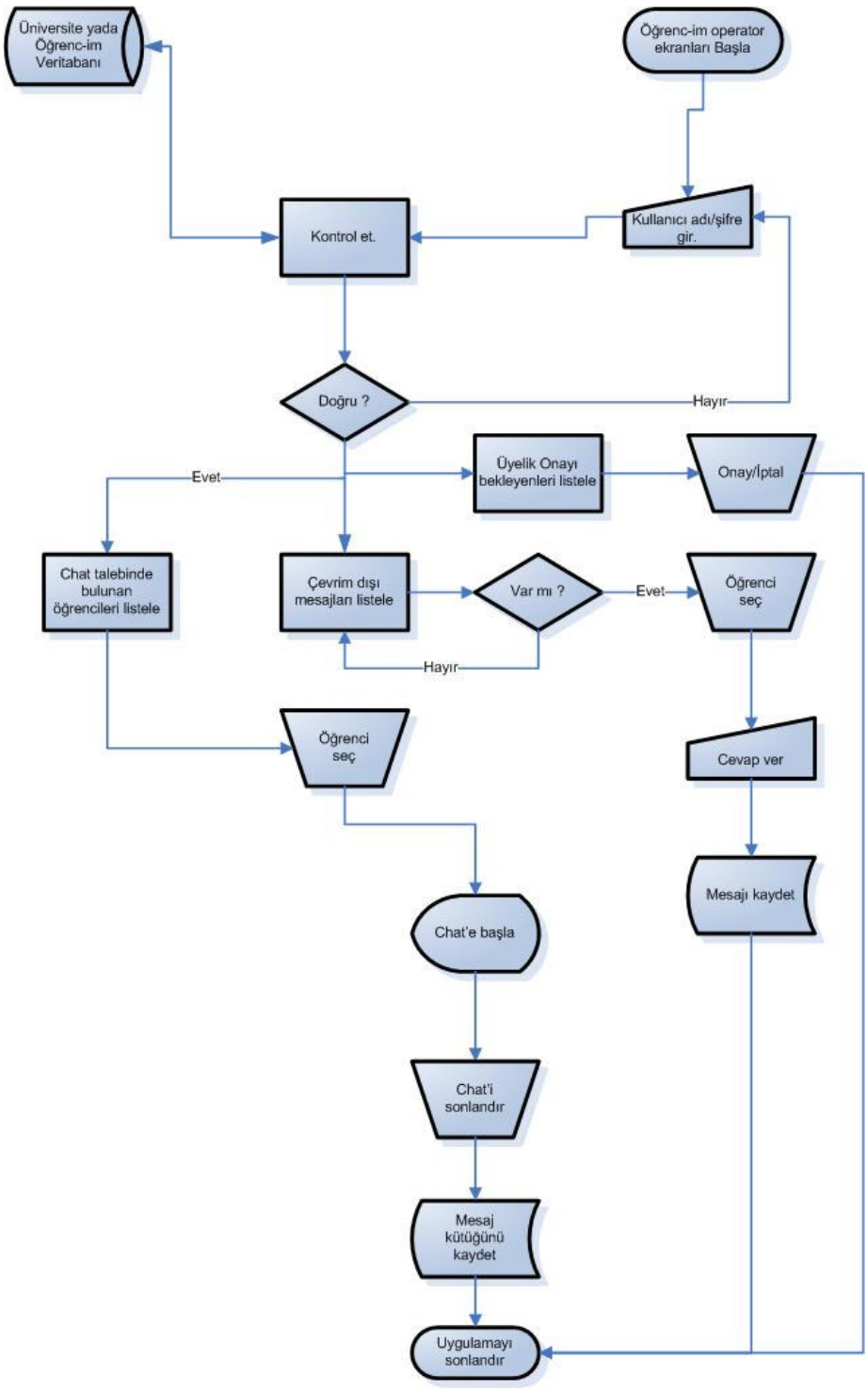
Akademisyenler ve öğrenci işleri yetkilileri üniversite veritabanlarında kayıtlı olarak bulunmaktadır. Önceki bölümde de vurgulandığı gibi sistem üniversite veritabanına entegre bir şekilde çalışabileceği gibi farklı bir üyelik sistemi mekanizmasıyla da kullanıcı ve operatörlerin yaratılması sağlanabilmektedir.

### 6.2.2 Öğrenci Girişi

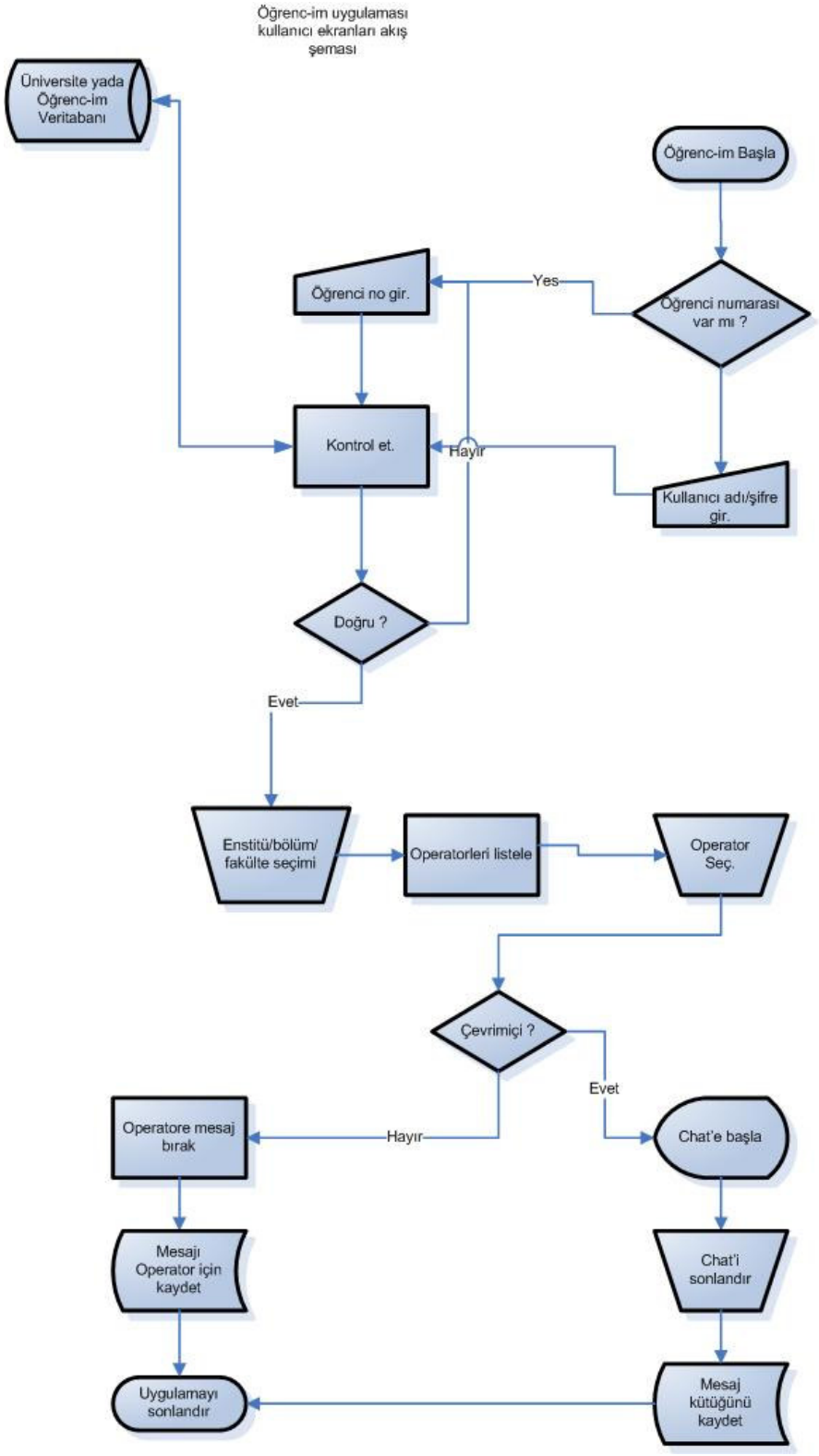
Öğrenciler sisteme giriş yaptıklarında karşılarına çıkan filtreleme ekranı vasıtasıyla, istedikleri fakülte, bölüm ya da enstitüden çevrimiçi operatörlerin listesini alabilmekte ve anlık mesajlaşma talebinde bulunabilmektedirler. Çevrimdışı operatörlere ise mesaj bırakmak için hazırlanmış olan ekranı kullanabilmektedirler.

Bıraktıkları çevrimdışı mesajlar ilgili operatörün ekranına düşmekte ve operatör sisteme giriş yaptığında mesajı alıp öğrenciye geri dönüş yapabilmektedir.

Öğrenci ekranları akışı Şekil 6.2'de gösterilmiştir.




Şekil 6.1: Öğrenc-im uygulaması operator ekranları akışı.




Şekil 6.2: Öğrenc-im uygulaması öğrenci ekranları akışı.

## 6.3 TEKNOLOJİ

Ajax yaklaşımı kullanılarak hazırlanan uygulamalar özel bir dil kullanmayı gerektirmemektedir. Öğrenc-im uygulamasının yazılımı gerçekleştirilirken aşağıdaki araç/diller kullanılmıştır:

**ASP.NET 1.0**

ASP.NET, Microsoft şirketinin özellikle Web programcılarına yönelik geliştirdiği Web uygulamaları geliştirme dilinin genel adıdır [30]. Bu dil dinamik Web sayfaları, Web uygulamaları ve XML tabanlı Web hizmetleri yaratılmasına olanak sağlamaktadır. Aynı şirket tarafından geliştirilen .NET altyapısının parçası, ve artık işletmece desteklenmeyen ASP teknolojisinin devamını teşkil eder. Dosya uzantıları genelde ".aspx" ile biter. Dosya yapısı ASP'ye göre farklılık gösterir ve iki bölümden oluşur. İlk bölümde HTML tabanlı kodlama yapılır [21]. İkinci bölümde ise sunucuda çalışması gereken mantıksal işlemler ve veritabanı işlemleri gerçekleşir. İkinci kısım "kod arkası" olarak da bilinir. Bu kısımın dosya uzantısı kodlandığı dilin kısaltmasını alır. Örneğin ilk kısmın sunucuda çalışan kısmı index.aspx ise ve kod arkası da Visual Basic dilinde yazılmış ise kod arkası dosya index.aspx.vb şeklinde kayıt edilir. Bu projede kullanılan kod arkası dili C#(sharp)'tır.

**C#**

C# Programlama Dili, Microsoft'un geliştirmiş olduğu yeni nesil bir dildir. .NET altyapısının sunduğu dillerden biridir. Java programlama diline benzer ve nesne yönelimli bir dildir.

**SQL Server 2005**

SQL Server 2005 Microsoft şirketi tarafından geliştirilmiş bir ilişkisel veritabanı yönetim sistemidir. Birincil sorgu dili olarak Transact-SQL'i kullanır. Analiz, raporlama ve uyarı servisleri gelişmiş ve yaygın kullanıma sahip bir veritabanı üründür.

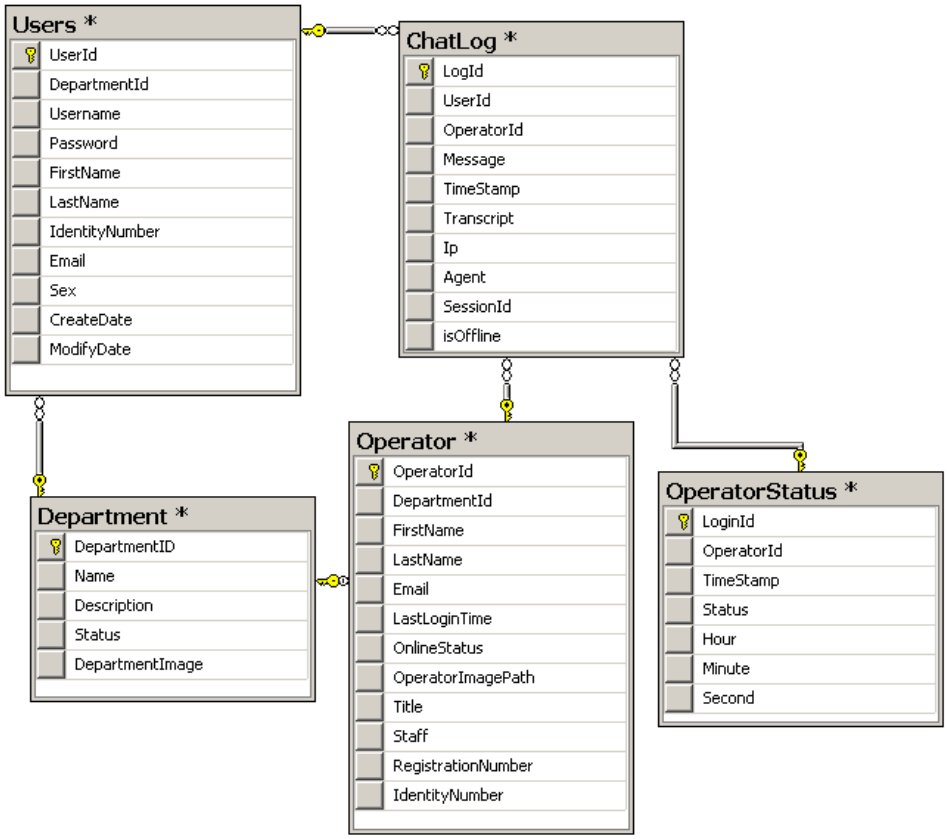
**Visual Studio 2003**

Microsoft'un Web ve masaüstü projeleri yaratmak isteyenler için geliştirdiği bir yazılım geliştirme platformudur. Bu tip ürünler IDE (Integrated Development Environment)

olarak da bilinirler. Bir Web sayfasını proje olarak kurgulayıp derleyebilme yeteneğine sahiptir. IDE olarak adlandırılan yardımcı yazılım geliştirme programları programcılara zaman kazandırmaktadırlar.

## 6.3 VERİTABANI YAPISI

Uygulamanın kullandığı veritabanı yapısı Şekil 6.3'de gösterilmiştir.




Şekil 6.3: Öğrenc-im uygulaması veritabanı yapısı.

# 7. SONUÇ

Bilgiye ulaşım için kullanılan internet, erişilecek bilgiyi kullanıcının kendisinin oluşturabilmesi imkanının yaratılması ile birlikte yeni bir döneme girmiştir. Web 2.0 olarak adlandırılan bu dönem, son kullanıcılar için Zengin İnternet Uygulaması (ZİU) geliştirme tekniklerinin ortaya çıkmasını da beraberinde getirmiştir. Kullanıcıların, tarayıcı vasıtasıyla sunucuya gönderdikleri isteklerin daha hızlı şekilde yanıtlanması, sunucunun daha az meşgul edilmesi ve böylelikle daha fazla kullanıcıya hizmet eder hale gelmesi, Web 2.0 kavramı içerisinde Ajax'ın en çok kullanılan teknik olmasını sağlamıştır. Web 2.0 toplumsal iletişim sistemleri kurmayı amaçlamaktadır. Dolayısı ile geniş kitlelerin internet uygulamalarını kullanması Web 2.0'ın önemli amaçları arasındadır. Bunun için klasik Web geliştirme yaklaşımları yetersiz kalmıştır ve kullanıcının katılımını amaçlayan yenilikçi, özgürlükçü, kullanışlı sayfalar oluşturabilmek Ajax gibi tekniklerin kullanımını da beraberinde getirmiştir.

ZİU yaklaşımı, masaüstü uygulamalarının zengin grafiksel arayüzünü Web'in evrensel erişimiyle bir araya getirerek, kullanıcıların Web sayfaları ile olan etkileşimini mümkün kılmakta, bant genişliği kullanımını ve sunucu yükünü azaltmaktadır; dolayısıyla, son kullanıcı deneyimini iyileştirmektedir.

Bu tez çalışmasında, geleneksel sunucu-istemci modelli Web tabanlı yazılım geliştirme yaklaşımının evrimi ile ilgili bilgiler verildikten sonra, ZİU yaklaşımı ve araçları tüm ayrıntıları ile anlatılmıştır. Daha sonra Ajax ile geliştirilmiş ve son kullanıcılar tarafından büyük ilgi görmüş olan; Google Maps (harita uygulaması), Flickr (fotoğraf paylaşım ve telifleme sitesi), Google Suggest (alternatif arama sonuçlarının önceden listelenmesini sağlayan uygulama) gibi uygulamaları hakkında bilgi verilmiştir.

Tez çalışması kapsamında geliştirilen Öğrenc-im uygulaması, yükseköğretim kurumlarında çalışan akademisyenlerin ve öğrenci işleri yetkililerinin, öğrencilerle anlık olarak Web üzerinden mesajlaşabilmesi olanağını vermektedir ve Enformatik Bölümü öğretim üyeleri ve Yüksek Lisans öğrencilerinin verileri ile çalışmak üzere hazırlanmıştır. Bu amaçla öğretim üyelerinin ve yardımcılarının verileri ile öğrencilere

ilişkin verilerin içinde bulunduğu bir MS-SQL veritabanı oluşturulmuştur. Öğrenc-im uygulaması, Enformatik Bölümü'nde istenen performansı vermesi ve diğer bölümlerden talep gelmesi durumunda, İstanbul Üniversitesi'nin veritabanlarına kolaylıkla entegre edilip o bölümlerde de kullanılabilecektir. Bu uygulama sayesinde, herhangi bir üyelik sistemine ihtiyaç duyulmayacak, öğrenciler sisteme öğrenci numaralarıyla veya tanımlanabilecek başka bir numara (şifre) ile giriş yapabilecek ve kendi bölümlerine bağlı akademisyenlerle anlık olarak iletişime geçebileceklerdir. Bunun dışında, akademisyenlerin çevrimdışı olması durumunda, öğrenciler tarafından mesaj bırakılabilecektir. Öğrenc-im, başlangıç olarak prototip şeklinde geliştirilmiştir. Uygulama, yeni eklentiler yapılabilecek, farklı sistemlere entegre olabilecek şekilde tasarlanmıştır.

Bu tez çalışmasında yeralan Öğrenc-im uygulaması ile ilgili yeni geliştirme talepleri kullanıcı deneyimleri sonucunda ortaya çıkacaktır. Sistemin üniversitelerin veritabanlarıyla entegrasyonu durumunda farklı çözümler geliştirilebilecektir. Örneğin, anlık mesajlaşma yapılan öğrenci ile ilgili sınav sonuçları, transkript, öğrencinin devam durumu gibi bilgiler anında akademisyenler tarafından ulaşılabilir olacaktır.

# KAYNAKLAR


1. ASLESON, R., SCHUTTA, N., 2006, *Foundations of Ajax*, Apress, Newyork, 1-59059-582-3

2. COLONY G., 2000, *My View : X Internet* [online], http://www.forrester.com/ER/Marketing/1,1503,214,FF.html

3. ALLAIRE, J., 2002, *Macromedia Flash MX—A next-generation rich client* [online], Macromedia, http://www.adobe.com/devnet/flash/whitepapers/richclient.pdf

4. MOLLY E., Holzschlag and Shirley E. Kaiser, 2002, *Working together for standards: The Web Standards Project* [online], http://www.webstandards.org/learn/faq/

5. JOHANSSON R., 2006, *Developing With Web Standards Recommendations and best practices* [online], 456 Berea Street, http://www.456bereastreet.com/lab/developing_with_web_standards/webstandards/

6. FLEMING, J., 1998, *Web Navigation: Designing the User Experience*, O'Reilly Media, Inc., 1-56592-351-0

7. From Wikipedia, the free encyclopedia, *Rich Internet Application* [online], http://en.wikipedia.org/wiki/Rich_Internet_application

8. KLEIN, N., CARLSON M., 2008, *Lazslo in Action: Rich Web Applications with OpenLazslo,* Manning Publications, USA, 1-932394-83-4

9. GARRETT J. J., 2005, *AJAX: A New Approach to Web Applications* [online], Adaptive Path, http://www.adaptivepath.com/publications/essays/archives/000385.php

10. DAVE, C., PASCARELLO, E., James D., 2006 *AJAX in Action*, Manning Publications, Greenwich, 1-932394-61-3

11. JACOBI, J., FALLOWS, J., 2006, *Pro JSF and Ajax Building Rich Internet Components*, Apress, Newyork, 1-59059-580-7

12. KEITH, J., 2007, *Bulletproof Ajax* , New Riders Press, Berkeley, CA, 0-321-47266-7

13. MUSSER, J., O'REILLY, T., 2007, *Web 2.0 Principles and Best Practices* , O'Reilly Media, Inc., Gravenstein Highway North, Sebastopol, 0-596-52769-1

14. *LOOSLEY, C., 2006, Rich Internet Applications: Design, Measurement and Management Challenges* [online], Keynote Systems, http://www.keynote.com/docs/whitepapers/RichInternet_5.pdf

15. BARKOL, D., GALLO, A., VAVILALA, R.K., 2008, *ASP.NET Ajax in Action*, Manning Publications, Greenwich, 1-933988-14-2

16. MOORE, D., BUDD, R., BENSON, E., 2007, *Professional Rich Internet Applications: Ajax and Beyond*, Wiley Publishing, Canada, 978-0-470-08280-5

17. PERRY, B., 2006, *Google Web Toolkit for Ajax,* O'Reilly Media, Inc., Sebastapol, California, 0-596-51022-5

18. O'REILLY T., 2005, *What is Web 2.0 ? Design Patterns and Business Models for the Next Generation of Software* [online], http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

19. GARRETT, J. J., 2002, *The Elements of User Experience: User-Centered Design for the Web*, New Riders Publishing, 0-7357-1202-6

20. MORVILLE, P., 2005, *Ambient Findability: What We Find Changes Who We Become,* O'Reilly Media, Inc., Sebastapol, California, 0-596-00765-5

*21.* WOOLSTON, D., 2006, *Pro Ajax And The .Net 2.0 Platform*, Apress, Newyork, 1-59059-670-6

22. GROSS, C., 2006, *Ajax Patterns and Best Practices*, Apress, Newyork, 1-59059-616-1

23. BAUSCH, P., BUMGARDNER J., 2006, *Flickr Hacks: Tips and Tools for Sharing Photos Online*, O'Reilly Media, Sebastapol, 0-596-10245-3

24. HOFFMAN, B., SULLIVAN, B. 2007, *Ajax Security*, Addison Wesley Pub., Boston, 0-321-49193-9

*25.* HOFFMAN, B., 2006, *Ajax Security Dangers* [online], SPI Labs HP Software, http://www.spidynamics.com/assets/documents/AJAXdangers.pdf

*26. An Example XML Document* [online], Refsnes Data, http://www.w3schools.com/xml/xml_tree.asp

*27. An Example XHTML Document* [online], Refsnes Data, http://www.w3schools.com/xhtml/xhtml_syntax.asp

28. From Wikipedia, the free encyclopedia, *Facebook* [online], http://en.wikipedia.org/wiki/Facebook

*29.* DUMLUPINAR, E., 2006, *Pro Ajax And The .Net 2.0 Platform*, Apress, Newyork, 1-59059-670-6

30. From Wikipedia, the free encyclopedia, *ASP.NET* [online], http://en.wikipedia.org/wiki/ASP.NET

31. O'REILLY T., 2006, *Web 2.0 Compact Definition: Trying Again* [online], http://radar.oreilly.com/2006/12/web-20-compact-definition-tryi.html

# ÖZGEÇMİŞ

1978 yılında İstanbul'da doğdum. İlk, orta ve lise öğrenimimi İstanbul'da tamamladım. Mezun olduğum 1995 yılında İstanbul Üniversitesi Bilgisayar Mühendisliği Bölümünü kazandım ve 2000 yılında da buradan mezun oldum. 2001 yılından bu yana Doğan Holding bünyesindeki şirketlerde, yazılım uzmanı pozisyonu da dahil olmak üzere çeşitli görevler aldım. 2003 yılında evlendim. 2 yıldır Medyanet Reklam Pazarlama A.Ş.'de yazılım departmanında Yazılım Müdürü olarak çalışmaktayım.