

1. İNTERNET VE WEB

Birlikte Düşünelim

1. İnternet nedir?
2. Web ile İnternet arasındaki ilişki nedir?
3. Arpanet ile günümüz web kullanımı arasındaki ilişki nedir?

Başlamadan Önce

E-Ticaret konusunu ele alabilmek için şüphesiz ki öncelikle yapılması gereken İnternet ve Web kavramlarının anlaşılmasılığını sağlamak olacaktır. Bununla birlikte İnternet'in tarihçesi de ele alınacak ve başlangıç noktası olarak kabul edilebilecek ARPANET üzerinde durulacaktır. Daha sonra İnternet'in aslında büyük önem kazanmasına yardımcı olan Web konusu ele alınacaktır. İnternet tek başına veri ve bilgi iletimi biçiminde düşünülebilir. Oysa Web' in gelişimiyle birlikte görsel ve işitsel zenginlik artmıştır. Zira, günümüzde pek çok eğlence platformu Web tabanlı hale gelmiştir.

Giriş

İnternet her ne kadar artık Türkçeleşmiş bir terim olsa da orijinal adı ağlar arası çalışma anlamına gelen *internetworking* teriminden gelmektedir. Kısaca uluslararası ağlar gibi algılanabilir. Esas karşılığı pek çok ağı birbirine bağlayan bir ağdır. Böylece ağa doğrudan bağlı bazı bilgisayarların sürekli aktif olmasından dolayı sıradan bir kullanıcı her girişinde istediği veriye erişebilmektedir. İnternet' in temeli ABD Savunma Bakanlığı' nın yaptırmış olduğu bir araştırmaya dayanmaktadır. Buna göre bilgisayarların zarar görmesi durumunda haberleşmeyi mümkün kılmak için bir sistem tasarlanmış ve ARPANET olarak isimlendirdikleri küçük bir ağda denemeler yapmışlardır. Ardından ABD Ulusal Bilim Kurulu (National Science Foundation veya NSF) ülkenin değişik yerlerine yerleştirdiği çok pahalı 6 adet süper bilgisayarı araştırmacıların hizmetine sokmuştur [Gökçen-2005]. Özette 1970' li yılların başlarında ABD Savunma Bakanlığı' nın dünyadaki pek çok bilim adamını birbirine bağlamak için kurduğu ağ bugünkü İnternet' in temelini oluşturmuştur [Laudon ve Laudon-2006].

Şu anda dünyada yüz binleri aşan sayıarda ağ, İnternet aracılığıyla birbirine bağlıdır. Ayrıca yüz milyonlarla ifade edilebilecek sayıda bireysel İnternet kullanıcısı vardır. Bireysel kullanıcılar İnternet hizmet sağlayıcıları aracılığıyla İnternet' e bağlanmaktadır.

Amerika Federal Ağ Oluşturma Konseyi 1995 yılında İnternet' i aşağıdaki üç karakteristiğe göre açıklanabilen bir küresel ES olarak tanımlamıştır [bkz. Laudon ve Laudon-2006]. Buna göre İnternet:

- İnternet Protokolünü temel alarak, eşsiz birer adresle birbirine bağlı Bilgisayarlar ve diğer cihazlardan oluşan bir ağdır.
- Ağ cihazlarının (bilgisayarlar, yönlendiriciler, göbekler vb.) TCP/IP veya buna uyumlu protokollerini kullanarak haberleşebildiği bir ağdır.
- Haberleşme ve ağ altyapısında üst seviyede yer alan hizmetleri sağlayan bir ağdır.

İnternet teknolojilerinin işletme performansına olan etkisi kolayca ölçülememektedir. Oysa yöneticiler genellikle bazı ölçütlerle göre değerlendirmeler yapmak ve karar vermek isterler. Çoğu zaman İnternet teknolojilerinin maliyetleri organizasyonlar için yüksek olabilmektedir. Bunun için *Soto-Acosta ve Merono-Cerdan* [2008] işletme performansı üzerinde İnternet teknolojilerinin nasıl bir etkisi olduğunu araştırmışlardır. Bunun sonucunda İnternet teknolojileriyle işletme performansı arasında doğrudan pozitif bir ilişki olmadığını tespit etmişlerdir. Ancak bununla beraber işletme içi işlemlere ve tüketiciye yönelik e-ticaret ile İnternet teknolojileri arasında pozitif bir ilişkiyi de tespit etmişlerdir. Araştırmanın en ilgi çekici bulgusu ise işletme içi işlemlere ve tüketiciye yönelik e-ticaret ile işletme performansı arasında pozitif bir

ilişki olmasıdır. Böylece dolaylı yoldan İnternet teknolojilerinin e-ticaret kullanımını sayesinde işletme performansına pozitif yönde etki ettiği ortaya çıkmaktadır.

Birçok standart Web' te zengin enformasyon sunumunu sağlamak için geliştirilmiştir. Bunlardan bazıları yardımçı metin aktarım protokolü, yardımçı metin biçimlendirme dili, uzatılabilir biçimlendirme dili, dosya aktarımı, gerçekten basit dağıtım, Web günlüğü, ses yayıcılığı, anında mesajlaşma ve cerezlerdir.

Web, İnternet' in en yaygın kullanılan hizmetidir. Bilindiği gibi İnternet birbirine bağlı ağlar anlamına gelmektedir. Bu ağlar üzerinden enformasyon alışverişi olmaktadır. Buna göre istemci-sunucu ilişkisi söz konusu olmaktadır. Ancak enformasyonu bulunduran bir sunucu ile onu almak isteyen istemci arasında biçim uyuşmazlıkları olabilir. Bu durumda evrensel standartlar sunan bir sisteme ihtiyaç duyulmuş ve Web ortaya çıkmıştır. Bu standartlar ilk olarak 1989 yılında belirlenmiştir. Böylece Web üzerinde paylaşılan enformasyon metin temelli olmuştur. Bunu takiben 1993 yılından sonra ilk kez grafik ara yüzlü Web tarayıcı geliştirilmiş ve adına Mosaic denmiştir. Böylece dokümanların Web' te grafiksel olarak görüntülenebilmesi sağlanmıştır. Ardından 1994 yılında Netscape Navigator isimli ilk ticari tarayıcı geliştirilmiştir. Daha sonra 1995 yılında Microsoft Internet Explorer ismindeki tarayıcıyı geliştirmiştir. Son olarak Google 2008 yılında günümüzde en yaygın olarak kullanılmakta olan Chrome ismindeki tarayıcıyı çıkartmıştır. Halen yeni tarayıcıların üretimi sürmekte olup, bunlar adından bahsetmeyi gerektirecek kadar yaygın kullanılmamaktadırlar.

Özetle Web sayfaları yardımçı metin ile biçimlendirilmiştir ve içlerine gömülü olan bağlar (link) aracılığıyla birbirlerindeki dokümanlara erişebilirler. Bu bağlar ses, resim, görüntü ve animasyon dosyaları gibi diğer nesnelere bağlanmayı da sağlayabilirler [Laudon ve Laudon-2006].

1.1. ARPANET

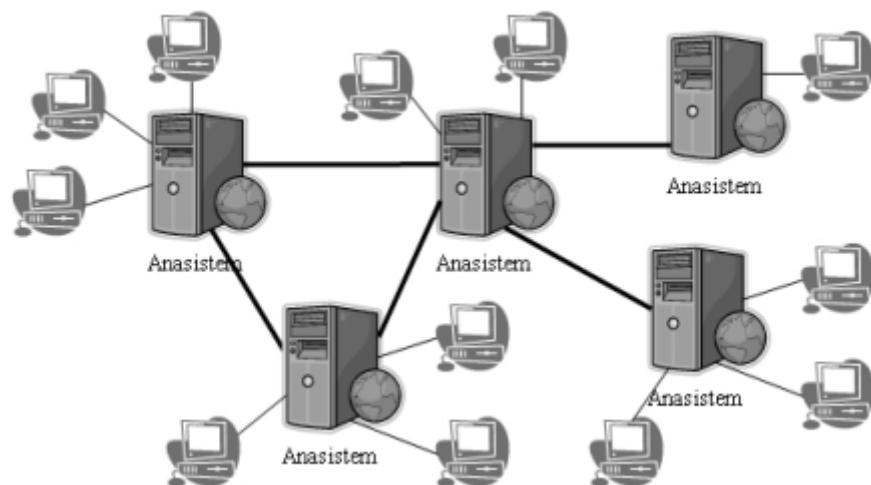
Gelişmiş Araştırma Projeleri Ajans Ağı (ARPANET), dünyanın ilk operasyonel paket anahtarlama ağlarından olan TCP/IP' yi uygulayan ilk ağdır ve küresel İnternet'in ne olacağının öncüsü olmuştur. Ağ başlangıçta ABD Savunma Bakanlığı'ndaki projelerde ve ABD'deki araştırma laboratuvarlarında kullanılmak üzere, ABD Savunma Bakanlığı'ndaki Gelişmiş Araştırma Projeleri Ajansı (ARPA, daha sonra DARPA) tarafından finanse edildi. ARPANET'in paket anahtarlaması, TCP/IP ile birlikte, İnternet'in çalışma şeklinin bel kemiğini oluşturur. Paket geçisi, Amerikalı mühendis Paul Baran, İngiliz bilim adamı Donald Davies [The History of Computing Project-2019 ve Internet Society-2019] ve Lincoln Laboratuvarı'ndan Lawrence Roberts'in konseptlerine ve tasarımlarına dayanıyordu. TCP/IP iletişim protokollerı ARPANET için bilgisayar bilimcileri Robert Kahn ve Vinton Cerf tarafından geliştirilmiştir.

1.1.1. TCP/IP

İletim kontrol protokolü (transmission control protocol veya TCP) ve İnternet protokolü (Internet protocol veya IP) günümüzde İnternet iletişiminde kullanılan protokollerdir. Aslında protokoller üst üste geçmiş katmanlar olarak görülmelidir. Buna göre en üst katmanda uygulama protokolleri vardır. Bunlar *yardımcı metin aktarım protokolü* (Hypertext transfer protocol veya HTTP) ve *dosya aktarım protokolü* (File transfer protocol veya FTP) gibi kullanıcı düzeyindeki protokollerdir. Örneğin HTTP, Web sayfalarının aktarımını bir tarayıcı aracılığıyla sağlanırken FTP, çeşitli yazılımlar vasıtasiyla dosya aktarımını sağlar. Bu katmandaki protokollere uygun olarak düzenlenmiş olan iletilecek veri bir alt katmana, yani TCP veya *kullanıcı veri bloğu protokoli*ne (user datagram protocol veya UDP) iletilir. Daha çok TCP kullanılmaktadır. Buna göre TCP gelen veriyi parçalara ayırır. Her bir parçanın başına alıcıda doğru olarak bir araya getirilebilmesi için sıra numarası koyar. Aynı amaçla kullanılan UDP' nden ise kısa verilerin gönderilmesinde yararlanılır. Çünkü UDP veriyi parçalara ayırmaz.

Internet' e doğrudan bağlı olan bir bilgisayar *anasistem* (host) olarak adlandırılır. Ancak unutmamak gereklidir ki bir yerel ağdaki bilgisayar Internet' e doğrudan bağlı değildir. Söz konusu bilgisayar çıkış noktasındaki bir sunucu vasıtasiyla Internet' e bağlanır. Görüldüğü gibi Internet' e doğrudan bağlı olan anasistem, yani bahsi geçen çıkış bilgisayarı olmaktadır. Bu sebeple anasistemler bazen sunucu olarak tanımlanır. Her anasistemin adresini gösteren bir numara vardır ve bu numaraya IP numarası denilir. Çünkü veri iletimini bir anasistemden diğerine gidene kadar IP kontrol eder. Ancak hemen hatırlatmak gereklidir ki numaraların akılda tutulması zorluğundan dolayı, bunlara karşılık gelen isimler kullanılır. Bu isimler *alan adı* (domain name)

olarak adlandırılırlar. Örneğin “istanbul.edu.tr” alan adıdır ve karşılık geldiği bir IP numarası vardır. Alan adı anasistemin adresini göstermemektedir. Dolayısıyla alan adlarının karşılık geldiği IP numaralarının listesi *alan adı sistemlerinde* (domain name system veya DNS) saklanır. İnternet hizmet sağlayıcılarının DNS’leri olduğu gibi bazı organizasyonlar kendi yerel ağları içerisinde DNS barındırabilirler.



Şekil 1 Anasistemlerin TCP/IP' üne göre iletişimdeki rolleri

Böylece herhangi bir anasisteme ulaşan veri buradan da Şekil 1’ de görüldüğü gibi o anasisteme bağlı istemci aygıta iletilir. Başka bir anlatım biçiminde, TCP katmanında düzenlenen verinin IP katmanında gideceği adres (veya rota) belirlenmektedir.

Eğer bir yerel ağ yönlendirici üzerinden Internet' e bağlanıyorsa o zaman bu ağdaki aygıtların yerel ağdaki IP numaraları önemsizdir. Zira tüm ağın tek bir IP numarası ile Internet' e bağlanması söz konusudur. Ancak yönlendiricide bilgisayarların yerel ağ IP numaraları saklanır. Böylece yönlendirici Internet' te yaptığı veri iletiminin sonuçlarını doğru bilgisayarlara aktarabilmektedir.

Bir sunucu veya anasistem kalıcı IP numarasına sahip olmak zorundadır. Sürekli değişen bir IP numarası bunların adreslerinin bilinmemesi anlamına gelecektir. Dolayısıyla bunlara *statik IP* numarası verilir. Ancak anasistemlere bağlı bilgisayarların IP numaraları genellikle değişkendir. Anasisteme esneklik sağlamak için kendisine tahsis edilmiş mevcut IP numaraları içinden birisini ataması mümkündür. Bu durum çoğu ev kullanıcıları için geçerlidir. İnternet hizmet sağlayıcı her bağlantıda farklı bir IP numarası verebilmektedir.

1.1.2. Tarihçe

Bugün dünya çapında veri iletişimiminin baskın temeli olan paket anahtarlama, ARPANET anlayışı sırasında ortaya çıkan bir konsepttir. Paket anahtarlama işleminden önce hem ses hem de veri iletişimini, geleneksel telefon devresinde olduğu gibi, her bir telefon görüşmesine iki iletişim istasyonu arasında özel, uçtan uca bir elektronik bağlantı tahsis edilen devre anahtarlama fikrine dayanıyordu. Kısaca veri uçtan uca parçalara ayrılmadan bütün olarak aktarılıyordu. Bu istasyonlar telefon veya bilgisayar olabilir. (Geçici olarak) tahsis edilmiş hat tipik olarak, başlangıç istasyonundan varış istasyonuna kadar uzanan bir zincire monte edilmiş birçok ara hattan oluşur. Paket anahtarlama ile bir veri sistemi, verileri hat boşaldıkça paket-paket hatta aktararak birden fazla makine ile iletişim kurmak için tek bir iletişim linkini kullanabilir. Böylece, bağlantı yalnızca paylaşılabilir değildir, farklı hedeflere mektup göndermek için tek bir posta kutusu kullanılabildiği gibi her paket diğer paketlerden bağımsız olarak yönlendirilebilir [Packet Switching History-2019].

Bilgisayar kullanıcıları arasında genel iletişimini sağlamak için tasarlanan bir bilgisayar ağı için ilk fikirler bilgisayar uzmanı Licklider, Bolt, Beranek ve Newman tarafından Nisan 1963'te bir "Galaksiler Arası Bilgisayar Ağı" konsepti tartışılarken yazıldı. Bu fikirler çağdaş interneti oluşturan hemen hemen her şeyi içeriyyordu. Licklider daha sonra Ivan Sutherland ve Bob Taylor'u bu bilgisayar ağı konseptinin çok önemli bir gelişme olduğuna ikna etti, ancak Licklider bu konsept üzerinde çalışmasına izin verilmeden önce ARPA'dan ayrıldı. Ivan Sutherland ve Bob Taylor, ARPA sponsorlu araştırmacıların, çeşitli kurumsal ve akademik yerlerde bulunan ARPA'nın sağladığı bilgisayarları kullanmasına izin vermek ve bu sayede, yeni yazılım ve bilimsel sonuçlara hızlıca erişim yapmak için bilgisayar iletişim ağı oluşturma konusundaki

ilgilerini sürdürdüler. Taylor, her biri ARPA'nın finanse ettiği ayrı bilgisayarlara bağlı üç bilgisayar terminaline sahipti: birincisi, Santa Monica'daki System Development Corporation (SDC) Q-32 için; ikincisi, Berkeley Kaliforniya Üniversitesi'ndeki Project Genie için; ve üçüncüsü, MIT'de Multics için. Taylor, durumu şöyle hatırlıyor: "Bu üç terminalin her biri için, üç farklı kullanıcı komut setim vardı. Bu yüzden, çevrimiçi olarak SDC'deki biriyle konuşuyorsam ve Berkeley'de veya MIT'de tanıdığım birisiyle konuşmak istiyorsam, bunun için SDC terminalinden kalkmali, gidip diğer terminale giriş yapmalı ve onlarla temas geçmem gerekiyordu. Ne yapılacağı açıktı: Bu üç terminale sahipseniz, gitmek istediğiniz her yere giden bir terminal olmalıydı. Bu fikir ARPANET'ti". Birbirine yakın tarihlerde, birkaç kişi daha (çoğunlukla bağımsız olarak) "paket anahtarlama" konusu üzerinde çalıştı ve Ulusal Paket Laboratuvarı (NPL) 5 Ağustos 1968'de Büyük Britanya'da ilk halka açık sunumunu yaptı.

1.2. İnternet Adresleri

İnternet, belirli bir destinasyondaki bir bilgisayara veri ve bilgi göndermek için posta hizmeti gibi bir adresleme sistemine dayanmaktadır. İnternet Protokol adresi için kısaltma olan bir IP adresi, İnternete bağlı her bir bilgisayarı veya cihazı benzersiz şekilde tanımlayan bir sayıdır. IP adresi genellikle her biri nokta ile ayrılmış dört sayı grubundan oluşur. İnternete bağlı bilgisayarlarındaki artış nedeniyle altı sayı grubunun da kullanılmaya başlandığına deðinmek gerekiyor. Her gruptaki sayı 0 ile 255 arasındadır. Örneðin, 72.14.207.99 sayıları bir IP adresidir. Genel olarak, her IP adresinin ilk kısmı aðı, son kısım ise belirli bilgisayarı tanımlar. Fakat, tamamen sayısal IP adreslerini hatırlamak ve kullanmak zordur. Bu nedenle, İnternet bir veya daha fazla IP adresini temsil eden metin bir adın kullanılmasını destekler. Bir alan adı, bir IP adresinin metin versiyonudur. Örneðin 72.14.207.99 IP adresine www.google.com alan adı karşılık gelmektedir. Bir IP adresinde olduğu gibi, bir alan adının bileşenleri noktalarla ayrılır. Alan adındaki ilk noktaya kadar olan metin, İnternet sunucusunun türünü tanımlar. Örneðin, www bir Web sunucusunu belirtir. Her alan adının son kısmı üst seviye olarak isimlendirilir. En tipik örneði olan com organizasyonun türünü göstermektedir ve ticari amaçlı olduğu bilgisini verir. Üst seviye alan adlarını atayan ve denetleyen kuruluş Internet Tahsisli Sayilar ve ïsimler Kurumudur (ICANN). Biz, com, info, name, net ve org gibi üst düzey alan adları için, alan adlarını satan ve yöneten bir organizasyon olan *registrar*' a bir alan adı için kaydolunur. Amerika Birleşik Devletleri dışındaki uluslararası Web siteleri için alan adı, Türkiye için tr gibi iki harften oluşan bir ülke kodunu da içerir.

Alan adı sistemi (DNS), İnternet'in alan adlarını ve ilgili IP adreslerini depolamak için kullandığı yöntemdir. Bir alan adı belirtildiðinde, DNS sunucusu, alan adını ilişkili IP adresine çevirir; böylece veri ve bilgiler doğru bilgisayara yönlendirilebilir. DNS sunucusu, genellikle bir İnternet erişim sağlayıcısı ile ilişkilendirilmiş bir İnternet sunucusudur. Daha önce deðinildiği gibi İnternetin büyümesi, IP adreslerinin yetersiz kalmasına neden olmuştur. Bu nedenle, IPv6 adı verilen yeni bir IP adresleme biçimini, mevcut IP adreslerinin sayısını artırmaktadır.

1.3. Dünya Çapında Ağ (World Wide Web)

Her ne kadar birçok kişi World Wide Web ve Internet terimlerini birbirlerinin yerine kullanıyor olsa da World Wide Web aslında İnternet'in bir hizmetidir. İnternet 1960'ların sonunda geliştirilmeye başlanmasaðken, World Wide Web 1990'ların başında ortaya çıkmıştı. O zamandan beri, daha yaygın olarak kullanılan İnternet hizmetlerinden biri olmak için olaðanüstü bir şekilde büydü. World Wide Web (WWW) veya Web, dünya çapında bir elektronik belge koleksiyonundan oluşur. Web üzerindeki her elektronik belgeye Web sayfası adı verilir ve bunlar metin, grafik, animasyon, ses ve video içerirler. Ayrıca, Web sayfalarının genellikle diğer belgelere yerlesik bağlantıları vardır. Bazı Web sayfaları statiktir (sabit), diğerleri ise dinamiktir (deðiþken). Statik Web sayfalarının ziyaretçileri hep aynı içeriði görürler. Bunun tam aksine, dinamik bir Web sayfasıyla, ziyaretçiler, istenen hisse senedi fiyatları, bölge için hava durumu veya uçuşlar için bilet kullanılabilirliği gibi görüntülenen içeriðin bir kısmını veya tamamını özelleştirebilir.

Bir Web sitesi, ilgili Web sayfalarının ve bir Web sunucusunda depolanan belgeler ve resimler gibi ilişkili öğelerin toplamıdır. Bir Web sunucusu, istenen Web sayfalarını bilgisayarınıza veren bir bilgisayardır. Aynı Web sunucusu birden fazla Web sitesini depolayabilir. Bazı endüstri uzmanları, Web 2.0 terimini, kullanıcıların kişisel bilgileri paylaşmaları için bir araç saglayan (sosyal ağlarda çalışan Web siteleri gibi), kullanıcıların Web sitesi içeriðini (örneðin wikiler gibi) değiþtirmelerine olanak tanlayan ve ziyaretçilerin

kullanması için siteye yerleştirilmiş uygulama yazılımlarını (örneğin e-posta ve kelime işlem programları gibi) barındıran Web siteleri için kullanmaktadır.

1.3.1. Web' de Gezinme

Bir Web tarayıcısı veya kısaca tarayıcı, kullanıcıların Web sayfalarına erişmelerini ve görüntülemelerini veya Web 2.0 programlarına erişmelerini sağlayan bir uygulama yazılımıdır. Web'de gezinmek için, İnternet'e bağlı ve bir Web tarayıcısı olan bir bilgisayara veya mobil bir cihaza ihtiyaç vardır. Kişisel bilgisayarlar için daha yaygın olarak kullanılan Web tarayıcıları, Internet Explorer, Firefox, Opera, Safari ve Google Chrome'dur. İnternet bağlantısı kurulduğunda, bir Web tarayıcısı başlatırsınız. Tarayıcı, bir başlangıç Web sayfasını alır ve görüntüler. Bu sayfa bazen tarayıcının ana sayfası olarak da adlandırılır. Görüntülenen ilk giriş sayfası Web tarayıcısı tarafından seçilir. Tarayıcının ana sayfası istenildiği zaman değiştirilebilir.

Giriş sayfası teriminin bir başka kullanımı, bir Web sitesinin gösterdiği ilk sayfayı ifade eder. Bir kitap kapağı veya bir Web sitesi için içindekiler tablosuna benzer şekilde, ana sayfa, Web sitesinin amacı ve içeriği hakkında bilgi sağlar. iGoogle gibi birçok Web sitesi, giriş sayfasını sizin ilginizi çeken alanları içerecek şekilde kişiselleştirmenize izin verir. Giriş sayfası genellikle diğer belgelere, Web sayfalarına veya Web sitelerine bağlantılar içerir.

Hiperlink' in kısaltması olan bir bağ, ilgili başka bir Web sayfasına veya bir Web sayfasının bir bölümünü yerlesik bir bağlantıdır. Akıllı telefonlar gibi İnternet özellikli mobil cihazlar, küçük ekranları ve sınırlı bilgi işlem gücü için tasarlanmış, mikro tarayıcı olarak adlandırılan özel bir tarayıcı türü kullanır.

Bir bilgisayarın veya mobil cihazın bir Web sayfasını görüntülemesi için sayfanın indirilmesi gereklidir. İndirme, bir bilgisayarın veya cihazın Web sayfası gibi bilgileri İnternet'teki bir sunucudan alma işlemidir. Bir tarayıcı bir Web sayfasını indirirken, genellikle tarayıcı penceresinde bir animasyonlu logo veya simge görüntüler. İndirme tamamlandığında animasyon durur. Bir Web sayfasını indirmek için gereken süre, İnternet bağlantınızın hızına ve ilgili grafiklerin miktarına bağlı olarak değişir.

1.3.2. Web Adresleri

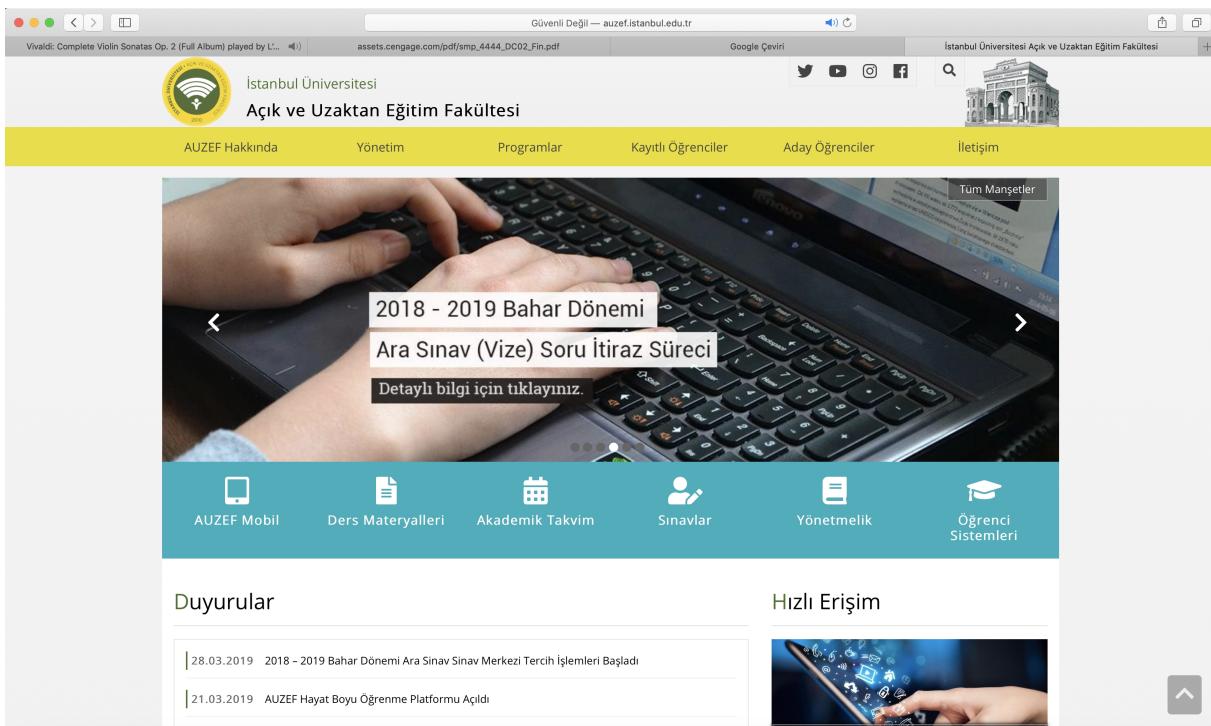
Bir Web sayfasının URL (Tek Tip Kaynak Bulucu) veya Web adresi adı verilen benzersiz bir adresi vardır. Örneğin, İstanbul Üniversitesi'nin Web sitesinin ana sayfasının Web adresi olarak <http://www.istanbul.edu.tr> vardır. Web tarayıcısı, Web adresini kullanarak bir Web sayfasını alır.

Bir Web sayfasının Web adresini biliyorsanız, tarayıcı penceresinin üstündeki Adres çubuğuuna yazabilirsiniz. Örneğin, http://auzef.istanbul.edu.tr/tr/_ Web adresini Adres çubuğuuna yazarsanız ve Enter tuşuna basarsanız, tarayıcı Şekil 2' de gösterilen Web sayfasını indirir ve görüntüler.

Bir Web adresi, bir protokol, alan adı ve bazen bir Web sitesindeki belirli bir Web sayfasına veya konuma giden yoldan oluşur. Birçok web sayfası adresi <http://> ile başlar. Hipemetin Aktarım Protokolü anlamına gelen http, sayfaların İnternette nasıl aktarıldığını tanımlayan bir kurallar kümesidir.

Hataları en aza indirmek için, birçok tarayıcı ve Web sitesi, Adres çubuğuuna Web adresinin <http://> ve www bölgümlerini girmenizi gerektirmez. Yanlış bir Web adresi girilmesi halinde, tarayıcı benzer adreslerin veya seçebileceğiniz ilgili Web sitelerinin bir listesini görüntüleyebilir.

Web tarayıcısına <http://www.istanbul.edu.tr/tr/content/yonetim/rektorluk> Web adresi girildiğinde, [istanbul.edu.tr](http://www.istanbul.edu.tr) Web sitesini içeren Web sunucusuna bir istek gönderilir. Sunucu daha sonra "content/yonetim" yolunda bulunan "rektorluk" adlı Web sayfasını alır ve tarayıcınıza iletir, ardından ekranın Web sayfasını görüntüler.



Şekil 2 Açık ve Uzaktan Eğitim Fakültesinin Web sitesinin görüntülenmesi

Zaman kazanmak için, birçok kullanıcı sık ziyaret ettiği Web sayfaları için yer işaretleri oluşturur. Bir yer imi veya favori, listedeki adını tıklatarak eriştiğiniz kayıtlı bir Web adresidir. Yani, bir Web sayfasını görüntülemek için bir Web adresi girmek yerine önceden kaydedilmiş bir yer imi tıklanabilir.

Tarayıcıya bir Web adresi girdiğinizde, bir Web sunucusundan bilgi ister veya alırsınız. Ayrıca, bazı Web sunucuları düzenli aralıklarla veya siteye güncelleme yapıldığında içeriği bilgisayarınıza gönderebilir. Örneğin, bazı Web sunucuları, spor müsabakalarının sonuçlarını veya hava durumu raporlarını bilgisayar ekranınızda görüntüleyebilmenizi sağlar.

Bölüm Özeti

‘Internet’ in önemi ve ilk olarak nasıl ortaya çıktıgı ele alınmıştır. Daha sonra dünya çapında ağa nasıl geçildiği incelenerek tarihçesi ve temel teknik alt yapısı gözden geçirilmiştir.

Kaynakça

Gökçen H., Yönetim Bilgi Sistemleri, EPİ Yayınları, 2005.

Internet Society, <http://www.internethalloffame.org/inductees/donald-davies>, 2019

Laudon K. C., Laudon J. P., Management Information Systems, Prentice Hall, Dokuzuncu baskı, 2006.

Packet Switching History, http://www.livinginternet.com/i/iw_packet_inv.htm, Living Internet, 2019

Soto-Acosta P., Merono-Cerdan A. L., “Evaluating Internet technologies business effectiveness”, Telematics and Informatics 50, 211 - 221, 2009.

The History of Computing Project, http://www.thocp.net/biographies/davies_donald.htm, 2019

2. HTTP, HTTPS, HTML VE XML I

Birlikte Düşünelim

1. http ne demektir? Ne amaçla kullanılır?
2. https ile http arasındaki fark nedir?
3. Şifreleme nedir? Güvenli haberleşmeyle şifrelemenin nasıl bir ilişkisi vardır?

Başlamadan Önce

Bilindiği gibi İnternet cihazlar arası bağlantı yapılmasını sağlayan bir ağdır. Fakat bu ağıda görsel unsurların daha etkili kullanılabilmesi için veri paylaşan cihazların arasında verinin nasıl gösterileceği hususunda bir biçim olması gerekmektedir. Söz konusu biçim için iki temel protokol kullanılmaktadır. Bunlar HTTP ve HTTPS protokolleridir. HTTP güvenli olmayan iletim için kullanılır. Bu kısımda http' nin temel özellikleri, mimarisi ve protokolün parametreleri ele alınacaktır. HTTPS ise güvenli iletim için kullanılır. Güvenli veri aktarımı ile şifreleme ve açık anahtar yapısı bu kısımda inceleneciktir.

Giriş

Kısaca Web enformasyonunu aktarmak için kullanılan protokole hiper metin aktarım protokolü (http) denir. Güvenlik açısından geliştirilmiş sürümü ise *güvenli hiper metin aktarım protokolü* (HTTPS) olarak adlandırılır. Bu protokole göre her bir Web sunucusunun *birörnek kaynak bulucu* (uniform resource locator veya URL) olarak adlandırılan bir adresi vardır. Bunlar Web sitesi adresi olarak bilinirler. Aslında söz konusu adres, IP numarası olarak adlandırılan bir numaradır. Fakat numaralar akılda kalmadığından *alan adı* (domain name) kullanılır. Alan adı harflerle ifade edilir ve her bir Web sayfasının eşsiz alan adı vardır.

Alan adlarının nasıl belirlendiğine deðinilmiştir. Buna göre en üst seviyede *kök alan* yer almaktadır. Bu alan temsili olarak “.” işaretile gösterilir. *Üst seviye alanda* ülke kısaltmaları yer alır. Örneðin *Türkiye*'nin kısaltması “tr” dir. Bunun istisnası ABD' dir. Zira ABD için hiçbir kısaltma kullanılmaz. Böylece diğer ülkeler için ikinci seviye, ABD için üst seviye olmaktadır. Hiyerarşı bu şekilde devam eder. Bu hiyerarşinin sonunda ise anasistem (Host) olan bilgisayar gösterilir. Unutulmamalıdır ki bu bir isimlendirme biçimidir. Aslında önemli olan IP numaralarıdır. Ayrıca gerekli olmasa da alan adlarının çoğunun başında “www” yer almaktadır.

Her ne kadar birçok girişimci alan adı alırken belirleyecekleri ismin çok önemli olduğunu düşünse de geçmiş deneyimler göstermiştir ki çoğu anlamı dahi bilinmeyen alan isimleri bugün herkes tarafından kullanılmaktadır. Bunların başında şüphesiz Amazon.com ve Google.com gelmektedir [bkz. Laudon ve Laudon-2006 ile Oz-2006].

2.1. Hipermetin Aktarım Protokolü (HTTP)

Hiper Metin Aktarım Protokolü (HTTP), dağıtılmış, işbirlikçi, hiper medya bilgi sistemleri için uygulama düzeyinde bir protokoldür. Bu, 1990'dan beri World Wide Web için veri iletişimini temelidir. HTTP, istek yöntemlerinin uzantılarını, hata kodlarını ve başlıklarını kullanarak başka amaçlarla kullanabilecek genel ve durum denetimsiz bir protokoldür.

Bu bölüm, HTTP/1.1 olarak adlandırılan protokolü tanımlayan RFC-2616 teknik özelliklerini temel almaktadır. HTTP/1.1, orijinal HTTP'nin (HTTP/1.0) bir revizyonudur. HTTP/1.0 ve HTTP/1.1 arasındaki temel fark, HTTP/1.0'in her istek-yanıt alışverişi için yeni bir bağlantı kullanmasıdır; HTTP/1.1 bağlantısı ise bir veya daha fazla istek - yanıt alışverişi için kullanılabilir.

Temel olarak HTTP, Web'de veri (HTML dosyaları, resim dosyaları, sorgu sonuçları vb.) sunmak için kullanılan TCP/IP tabanlı bir iletişim protokolüdür. Varsayılan bağlantı noktası TCP 80'dir, ancak başka bağlantı noktaları da kullanılabilir. Bilgisayarların birbirleriyle iletişim kurmaları için standart bir yol sağlar. HTTP belirtimi, istemcilerin istek verilerinin nasıl oluşturulacağını ve sunucuya gönderileceğini ve sunucuların bu isteklere nasıl yanıt vereceğini belirler.

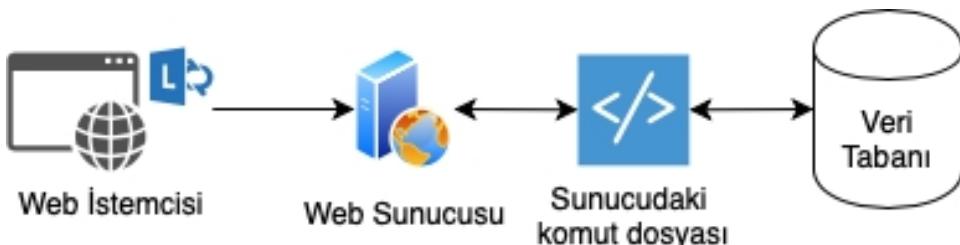
2.1.1. Temel Özellikler

HTTP'yi basit ama güçlü bir protokol yapan üç temel özellik vardır:

- HTTP bağlantısızdır: HTTP istemcisi, yani bir tarayıcı bir HTTP isteği başlatır ve bir istek yapıldıktan sonra istemci yanıt bekler. Sunucu isteği işler ve istemcinin bağlantıyı kesmesinden sonra bir yanıt gönderir. İstemci ve sunucu arasında birbiri için yeni olan bağlantılarla ilgili daha fazla talepler yapılır.
- HTTP medyadan bağımsızdır: Hem istemci hem de sunucu, veri içeriğinin nasıl işleneceğini bildiği sürece, herhangi bir veri türünün HTTP tarafından gönderilebileceği anlamına gelir. İstemcinin yanı sıra sunucunun da içerik türünü belirtmesi gereklidir.
- HTTP durum denetimsizdir: Yukarıda da belirtildiği gibi, HTTP bağlantısızdır ve bu HTTP'nin durum denetimsiz bir protokol olmasının doğrudan bir sonucudur. Sunucu ve istemci, yalnızca geçerli bir istek sırasında birbirlerinden haberdardır. Daha sonra, ikisi de birbirlerini unutur. Protokolün bu niteliği nedeniyle, ne istemci ne de tarayıcı web sayfalarındaki farklı istekler arasındaki bilgileri koruyamaz.

2.1.2. Temel Mimari

Şekil 3'teki şema bir web uygulamasının çok temel bir mimarisini göstermektedir ve HTTP'nin oturduğu yer açıklamaktadır.



Şekil 3 HTTP Protokolü

HTTP protokolü, web tarayıcılarının, robotların ve arama motorlarının vb. HTTP istemcileri gibi davranışları ve Web sunucusunun bir sunucu gibi davranışları istemci/sunucu tabanlı mimariye dayalı bir istek/yanıt protokolüdür.

HTTP istemcisi, sunucuya bir istek yöntemi, URI ve protokol sürümü biçiminde bir istek gönderir, ardından istek düzenleyicileri, istemci bilgileri ve TCP / IP bağlantısı üzerinden gövde içeriğini içeren bir mesaj izler.

HTTP sunucusu, mesajın protokol versiyonunu ve bir başarı veya hata kodunu içeren bir durum hattına cevap gönderir ve ardından sunucu bilgilerini, varlık meta bilgilerini ve gövde içeriğini içeren bir mesaj izler.

2.1.3. Protokol Parametreleri

Bu kısım, önemli HTTP Protokol Parametrelerinin birkaçını ve bunların iletişimde kullanılma şeklinin (örneğin, tarih formatı, URL formatı vb.) sözdizimlerini listeleyecektir. Bu, HTTP istemcisi veya sunucu programları yazarken istek ve yanıt mesajlarını oluşturmaya yardımcı olacaktır.

2.1.3.1. HTTP Versiyonu

HTTP, protokolün sürümlerini belirtmek için bir “`<major>. <minor>`” numaralandırma düzeni kullanır. Protokol sürüm politikası, gönderenin bir mesajın biçimini ve bu iletişim yoluyla elde edilen özelliklerden ziyade gönderenin mesajını biçimini ve daha fazla HTTP iletişimini anlayabilme kapasitesini belirtmeye izin vermek için tasarlanmıştır. İletişim davranışını etkilemeyen veya yalnızca genişletilebilir alan değerlerine ilave edilen ileti bileşenlerinin eklenmesi için sürüm numarasında herhangi bir değişiklik yapılmaz. Protokolde yapılan değişiklikler, genel ileti ayrıştırma algoritmasını değiştirmeyen, ancak ileti semantигine ekleyebilen ve gönderenin ek yeteneklerini ima eden ufak özellikler eklediğinde, `<minor>` sayı artar. Protokol içindeki bir mesajın formatı değiştirildiğinde ise `<major>` sayısı artırılır.

Major ve minor sayılarının ayrı tamsayılar olarak değerlendirilmesi gereği ve her birinin tek bir rakamdan daha yüksek olabileceği unutulmamalıdır. Dolayısıyla, HTTP / 2.4, HTTP / 2.13'ten daha düşük bir

sürümdür. HTTP / 2.13 de HTTP / 12.3'ten düşük bir sürümdür. Baştaki sıfırlar alıcılar tarafından göz ardı edilmeli ve gönderilmemelidir.

2.1.3.2. Tek Tip Kaynak Tanımlayıcıları

URI'lar birçok adla bilinir: WWW adresleri, Evrensel Belge Tanımlayıcıları, Tek Tip Kaynak Tanımlayıcıları [Berners-Lee, T. - 1994] ve son olarak Tek Tip Kaynak Belirleyicileri (URL) [Berners-Lee, T. ve ark. - 1994] ile Tek Tip Kaynak Adlarının (URN) [Sollins, K. ve Masinter .L. - 1994] birleşimidir. HTTP söz konusu olduğunda, Tek Tip Kaynak Tanımlayıcıları basitçe, sunucu adı, yeri ya da herhangi bir başka özelliği veya bir kaynağı tanımlayan, biçimlendirilmiş dizelerdir.

URI, bir kaynağı belirli bir isim alanında ada göre tanımlar ancak kaynağın nasıl elde edilebileceğini belirtmez. Bu URI türüne URN denir. Genellikle bazı yazım biçimleri kullanılarak, bir isim alanını tanımlamak için XML Şeması belgelerinde kullanılan URN'lere rastlanabilir.

URL, belirli bir kaynağın ağ konumunu tanımlayan bir URI türüdür. Bir URN'den farklı olarak, URL, kaynağın nasıl elde edilebileceğini belirtir. URL'leri her gün <http://ornek.com> biçiminde kullanıyoruz. Ancak bir URL'nin bir HTTP URL olması gerekmek, <ftp://ornek.com>, <smb://ornek.com> vb. olabilir.

2.1.3.3. Tarih / Saat Biçimleri

HTTP uygulamaları, tarih / saat gösterimi için tarihsel olarak üç farklı formata izin vermiştir:

Sun, 06 Nov 1994 08:49:37 GMT

Sunday, 06-Nov-94 08:49:37 GMT

Sun Nov 6 08:49:37 1994

İlk format bir İnternet standardı olarak tercih edilir ve RFC 1123 tarafından tanımlanmış olan sabit uzunluklu bir alt kümeyi temsil eder [Braden - 1989]. İkinci format sık olarak kullanımdadır, eski RFC 850 [Horton - 1987] tarih formatına dayanır fakat dört basamaklı bir yıl içermez. Tarih değerini çözümleyen HTTP / 1.1 istemcileri ve sunucuları, her üç biçimde kabul etmelidir (HTTP / 1.0 ile uyumluluk için), ancak başlık alanlarındaki HTTP-tarih değerlerini temsil etmek için yalnızca RFC 1123 biçimini oluşturmaları gereklidir.

Tüm HTTP tarih / saat biçimleri istisnasız Greenwich Mean Time (GMT) ile temsil edilmelidir. HTTP' nin amaçları doğrultusunda, GMT UTC' ye (Eşgüdümlü Evrensel Zaman) tam olarak eşittir. Bu, ilk iki formatta "GMT" nin zaman dilimi için üç harfli kısaltma olarak dahil edilmesiyle gösterilmektedir. HTTP tarihi büyük / küçük harfe duyarlıdır ve özel karakter içermemelidir. Bazı HTTP başlık alanları, saniye cinsinden bir zaman değerinin, mesajın alındığı zamandan sonra geçen süreyi bir tam sayıla belirtmesine izin verir.

2.1.3.4. Karakter Kümesi

HTTP, "karakter kümesi" terimi için MIME (Çok Amaçlı İnternet Posta Uzantıları) için açıklanan aynı tanımı kullanır. Burada "karakter seti" terimi, bir byte' lik veriyi bir karakter dizisine dönüştürmek için bir veya daha fazla tabloyla kullanılan bir yöntemi belirtmek için kullanılır. Diğer yöndeği (karakter dizisinden byte' a) dönüşüm gerekli değildir, çünkü belirli bir karakter kümesinde tüm karakterler bulunamayabilir ve bir karakter kümesi belirli bir karakteri temsil etmek için birden fazla byte dizisi sağlayabilir. Bu tanım, ASCII gibi basit tek tablo eşlemesinden ISO-2022 tekniklerinin kullanıldığı karmaşık tablo değiştirme yöntemlerine kadar çeşitli karakter kodlaması türlerine izin vermek için tasarlanmıştır.

Bazı HTTP / 1.0 yazılımları, karakter üstdilgisi olmayan Content-Type üstdilgisini "karakter kümesini alıcı tahmin etmeli" biçiminde yanlış yorumladı. Bu davranıştı yenmek isteyen göndericiler, karakter kümesi ISO-8859-1 (en yaygın olan) olsa ve karakterin alıcıyı karıştırımayacağı biliniyorsa bile karakter kümesini belirtmelidir.

2.1.3.5. İçerik Kodlama

İçerik kodlama değerleri, bir varlığa uygulanan veya uygulanabilen bir kodlama dönüşümünü belirtir. İçerik kodlamaları öncelikle, bir belgenin, dayandığı ortam türünün kimliğini kaybetmeden ve bilgi kaybı olmadan sıkıştırılmasını veya başka şekilde dönüştürülmesini sağlamak için kullanılır. Sıklıkla, varlık kodlanmış biçimde depolanır, doğrudan iletilir ve alıcı tarafından çözülür.

2.1.3.6. Transfer Kodlama

Transfer kodlama, ağa üzerinden "güvenli taşımayı" sağlamak için bir varlık ve/veya gövde (body) üzerine uygulanan bir kodlama dönüşümünü belirtmek için kullanılır. Transfer kodlama orijinal varlık yerine mesajın bir özelliğidir ve bu nedenle içerik kodlamasından farklıdır. Tüm transfer kodlama değerleri büyük / küçük harf duyarsızdır.

2.1.3.7. Medya Türleri

HTTP, açık ve genişletilebilir veri yazmayı sağlamak ve haberleşme türünü belirtmek için Content-Type ve Accept başlık alanlarında tanımlı Internet Medya Türlerini [Postel - 1996] kullanır.

Tür, alt tip ve parametre öznitelik adları büyük / küçük harf duyarsızdır. Başka bir ifadeyle doğru yazıldığı sürece büyük ya da küçük harf yazmak önemsizdir. Medya tipi değerleri İnternet Atanmış Numara Otoritesine kayıtlıdır. Medya tipi kayıt işlemi RFC 1590'da [Postel - 1996] ana hatlarıyla belirtilmiştir. Kayıtlı olmayan medya türlerinin kullanımı önerilmez. Kayıtlı bazı medya türü örnekleri application, audio, image, video biçimindedir.

2.1.3.8. Ürün Belirteçleri

Ürün belirteçleri, iletişim kurma uygulamalarının kendilerini yazılım adı ve sürümleriyle tanımlamalarını sağlamak için kullanılır. Ürün belirteçlerini kullanan alanların çoğu, uygulamanın önemli bir bölümünü oluşturan alt ürünlerin boşlukla ayrılmış olarak listelenmesine izin verir. Geleneksel olarak ürünler, başvuruyu tanımlamak için önem sırasına göre listelenmiştir. Örneğin Server: Apache / 0.8.4 gibi bir belirteç sunucusu olarak (ürün) Apache sürüm 0.8.4' ün kullanıldığını belirtmektedir. Görüldüğü gibi ürün belirteçleri, ürün ve sürümü şeklinde iki parçadan oluşmakta ve gerektiği kadar ürün-sürüm belirteci kullanılmaktadır.

Ürün belirteçleri kısa ve net olmalıdır. Reklam veya diğer önemli olmayan bilgiler için kullanılmamalıdır. Herhangi bir belirteç ürün sürümünde görünebilse de bu belirteç yalnızca bir sürüm tanımlayıcısı için kullanılmalıdır (yani, aynı ürünün art arda gelen sürümleri, yalnızca ürün değerinin ürün sürümü bölümünde farklı olmalıdır). Sürüm belirteci olarak farklı bilgilere yer verilmemelidir.

2.1.3.9. Kalite değerleri

HTTP içerik iletişim, çeşitli gözardı edilebilir parametrelerin göreceli önemini ("ağırlık") göstermek için kısa sayılar kullanır. Ağırlık 0 ile 1 arasında gerçek bir sayıya nispi olarak dönüştürülür, burada 0 en düşük ve 1 en yüksek değerdir. Bir parametrenin kalite değeri 0 ise, bu parametreye sahip içerik alıcı için "kabul edilemez" dir. HTTP / 1.1 uygulamaları, ondalık noktadan sonra üç basamaktan fazlasını kabul etmemektedir. Bu değerlerin kullanıcı yapılandırması da bu şekilde sınırlanmalıdır. "Kalite değerleri" aslında yanlış bir isimlendirmidir, çünkü bu değerler sadece istenen kalitede nispi bozulmayı temsil eder.

2.1.3.10. Dil Etiketleri

Dil etiketi, bilginin diğer insanlara iletilmesi için insanlar tarafından konuşulan, yazılmış veya başka şekilde aktarılan doğal bir dili tanımlar. Burada bilgisayar dilleri açıkça hariç tutulmuştur. HTTP, Accept-Language ve Content-Language alanındaki dil etiketlerini kullanır.

HTTP dili etiketlerinin sözdizimi ve kaydı, RFC 1766 [Alvestrand - 1995] tarafından tanımlanalarla aynıdır. Özettir, bir dil etiketi bir veya daha fazla bölümden oluşur: Birincil dil etiketi ve muhtemelen boş bir alt etiket dizisi.

Etikette boşluk bulunmaz ve tüm etiketler büyük / küçük harf duyarsızdır. Örnek etiketler şunları içerir:

en, en-US, en-cockney, i-cherokee, x-pig-latin

Burada herhangi bir iki harfli birincil etiketin (örneğin en) bir ISO-639 dil kısaltması olduğu ve herhangi bir iki harfli ilk alt etiketin (örneğin US) bir ISO-3166 ülke kodu olduğu unutulmamalıdır. Yukarıdaki son üç etiket kayıtlı etiketler değildir; sonucusu hariç tümü, gelecekte kaydedilecek etiket örnekleridir.

2.1.3.11. Varlık Etiketleri

Varlık etiketleri, aynı istenen kaynaktan iki veya daha fazla varlığın karşılaştırılması için kullanılır. HTTP / 1.1, ETag, If-Match, If-None-Match ve If-Range başlık alanlarındaki varlık etiketlerini kullanır. Bir varlık etiketi, muhtemelen bir zayıflık göstergesi ile önceden eklenmiş bir dizeden oluşur. Bir "güçlü varlık etiketi", yalnızca byte-byte karşılıklı eşitse, kaynağın iki varlığı tarafından paylaşılabilir. "W /" öneki ile belirtilen "zayıf varlık etiketi", yalnızca varlıklar eşdeğer olduğunda (tam eşitlik yok) ve anlambilimde önemli bir değişiklik olmadan birbirleriyle değiştirilebiliyorsa, kaynağın iki varlığı tarafından paylaşılabilir. Zayıf bir varlık etiketi yalnızca zayıf karşılaştırma için kullanılabilir.

2.1.3.12. Menzil Birimleri

HTTP / 1.1, istemcilerin, yanıt öğesinin yalnızca bir kısmının (bir dizi aralığı) yanıtın içine dahil edilmesini istemesini sağlar. HTTP / 1.1, Range ve Content-Range başlık alanlarındaki aralık birimlerini kullanır. Bir varlık, çeşitli yapısal birimlere göre alt gruplara ayrılabilir.

HTTP / 1.1 tarafından tanımlanan tek aralık birimi "byte" dır. HTTP / 1.1 uygulamaları, diğer birimleri kullanarak belirtilen aralıkları göz ardı edebilir. HTTP / 1.1, aralık bilgisi belirtmemiş uygulamaların yürütülmesine izin verecek şekilde tasarlanmıştır.

Aşağıdaki örnekte dosyanın ilk 500 ve son 500 baytinın istenmektedir. Aralıklar çakışırsa istek sunucu tarafından reddedilebilir.

Range: bytes=0-500, -500

2.2. Güvenli Hipernetin Aktarım Protokolü (HTTPS)

Güvenli HTTP (HTTPS), HTTP ile birlikte kullanılmak üzere tasarlanmış güvenli bir mesaj odaklı iletişim protokolüdür. HTTPS, HTTP' nin mesajlaşma modeliyle bir arada bulunacak ve HTTP uygulamalarıyla kolayca bütünlendirilecek şekilde tasarlanmıştır.

Güvenli HTTP, HTTP istemcileri ve sunucularına çeşitli güvenlik mekanizmaları sağlayarak, Web için çok çeşitli potansiyel kullanımlara uygun güvenlik hizmeti seçeneklerini sunar. HTTPS, işlem modelini ve HTTP' nin uygulama özelliklerini korurken hem istemciye hem de sunucuya (hem isteklere hem de cevaplara eşit muamele edildiği gibi) hem de her iki tarafın tercihleri için simetrik yetenekler sağlar.

Çeşitli şifreleme mesaj format standardı HTTPS istemcileri ve sunucularına dahil edilebilir. HTTPS, çeşitli uygulamalar arasında birlikte çalışmayı destekler ve HTTP ile uyumludur. HTTPS uyumlu istemcilerle HTTPS kullanmayan sunucular arası iletişim kurabilir ve bunun tersi de mümkündür, ancak bu tür işlemler HTTPS güvenlik özelliklerini kullanmamaktadır.

HTTPS, yalnızca simetrik anahtar işlem modlarını desteklediğinden, istemci tarafında açık anahtar sertifikaları (veya kısaca açık anahtarlar) gerektirmez. Bu önemlidir, çünkü kendiliğinden özel işlemlerin, bireysel kullanıcıların yerlesik bir açık anahtara sahip olmalarını gerektirmeden gerçekleşebilecegi anlamına gelir. HTTPS her yerde sertifikalandırma altyapısından yararlanabilir.

HTTPS, uçtan uca güvenli işlemleri destekler. Müşteriler, güvenli bir işlem başlatmak için (genellikle mesaj başlıklarında verilen bilgileri kullanarak) "kullanıma hazır hale getirilebilir"; bu, örneğin kullanıcı adı ve

şifre gibi doldurma formlarının şifrelenmesini desteklemek için kullanılabilir. HTTPS ile ağ üzerinden hiçbir hassas verinin gönderilmesine gerek kalmaz.

HTTPS şifreleme algoritmaları, modları ve parametrelerinin tam esnekliğini; seçeneklerin görüşülmesi, istemcilerin ve sunucuların işlem modları, kriptografik algoritmaları (imzalama için RSA, DSA, şifreleme için DES ve RC2, vb.) ve sertifika seçimiyle sağlar.

2.2.1. Şifreleme ve Açık Anahtar Altyapısı

Pek çok işletme sakladığı dijital enformasyonları, fiziksel transferi veya Internet üzerinden iletimi korumak için şifrelemeyi kullanır. Şifreleme kısaca açık metin veya verinin, gönderici veya istenilen alıcıdan başka kimse tarafından okunamayacak biçimde şifreli metne (cipher text) dönüştürülmesi sürecidir. Veri, şifreleme anahtarı olarak isimlendirilen gizli bir dijital kod kullanılarak şifrelenir. Anahtar açık veriyi şifreli metne dönüştürür. Mesaj alıcı tarafından deşifre edilmelidir.

Web üzerindeki ağ trafiğini şifrelemek için kullanılan iki metottan birisi SSL ve diğeri de HTTPS' dir. Güvenli Yuva Katmanı (SSL: Secure Sockets Layer) ve halefi Ulaşım Katmam Güvenliği (TLS: Transport Layer Security) istemci ve sunucu bilgisayarların güvenli bir Web oturumu esnasında birbirleriyle haberleşirken şifreleme ve deşifre etme faaliyetlerini yönetebilmelerini sağlar. Güvenli Hiper Metin Aktarım Protokolü (HTTPS) internet üzerinde akan verinin şifrelenmesi için kullanılan bir diğer protokoldür. Fakat SSL ve TSL' in iki bilgisayar arasında güvenli bir bağlantı oluşturmak için tasarlanmış olmasına karşın, bu bireysel mesajlarla sınırlıdır.

Güvenli oturum oluşturma becerisi istemci internet tarayıcı yazılımı ve sunucularda bulunmaktadır. İstemci ve sunucu hangi anahtar ve güvenlik seviyesinin kullanılacağı konusunda görüşür. Bir kez istemci ve sunucu arasında güvenli bir oturum oluşturulduğunda, bu oturumdaki tüm mesajlar şifrelenir.

Şifreleme için iki alternatif yöntem bulunmaktadır. Bunlar simetrik anahtar şifrelemesi ve açık anahtar şifrelemesidir. Simetrik anahtar şifrelemesinde gönderici ve alıcı tek bir şifreleme anahtarı yaratıp aynı anahtarı paylaşarak güvenli bir internet oturumu oluştururlar. Şifrelemenin uzunluğu sahip olduğu bit uzunluğu cinsinden Ölçülür. Günümüzde tipik bir anahtar 256 bit (ikilik düzendeki 256 haneden oluşan bir dizi) uzunlığında olacaktır.

Tüm simetrik şifreleme düzenlemelerindeki sorun, anahtarın bir şekilde gönderici ve alıcılar arasında paylaşılması zorunluluğudur. Zira bu durum anahtarı ele geçirebilecek olan dışarıda kılere onu açık etmek anlamına gelmektedir. Daha güvenli bir şifreleme biçimi açık anahtar şifrelemesidir ve iki anahtar kullanılır. Bunlar paylaşılan (veya açık) ve tamamen saklı anahtarlardır. Anahtarlar matematiksel olarak ilişkilidir. Böylece bir anahtarla şifrelenmiş olan veri yalnızca diğer anahtar kullanılarak deşifre edilebilir. Mesaj göndermek ve almak için haberleşme tarafları önce birbirinden ayrı, saklı ve açık anahtar çiftleri oluştururlar. Gönderici bir mesajı alıcının açık anahtarıyla şifreler. Mesaj alınırken, alıcı kendi saklı anahtarını onu deşifre etmek için kullanır.

Dijital sertifikalar çevrimiçi işlemlerin korunması iyi kullanıcılardan ve elektronik varlıkların kimliklerinin oluşturulmasında kullanılan veri dosyalarıdır. Bir dijital sertifika sistemi, onay makamı (CA: Certification Authority) olarak bilinen güvenilir üçüncü bir şahistan kullanıcının kimliğinin geçerliliğini tespit etmek için yardım alır. Birleşik Devletler ve dünya genelinde VeriSign, IdenTrust ve Avustralya'nın KeyPost'u gibi pek yok CA vardır.

CA bir dijital sertifikayı kullanıcının kimliği olarak çevrimdışı tasdik eder. Bu enformasyon bir CA sunucusuna konulur. Sunucu içinde sahibinin kimlik bilgileri ve açık anahtarının bir kopyasının bulunduğu şifrelenmiş bir dijital sertifikayı üretir. Sertifika açık anahtarın belirtilen kullanıcıya ait olduğunu tasdik etmektedir. CA kendi açık anahtarını yazılı olarak ya da internet üzerinde yayırlar. Şifrelenmiş bir mesajın alıcısı CA'nın açık anahtarını mesaja eklenmiş dijital sertifikayı deşifre etmek için kullanır. Böylece bunun CA'ye ait olduğu tasdik edilmiş olur. Daha soma alıcı göndericinin açık anahtarını ve sertifikada bulunan kimlik bilgilerini alır. Bu bilgileri kullanarak alıcı şifreli bir cevap gönderebilir. Örneğin; dijital sertifika sistemi, veri alış verisi başlamadan önce bir kredi kartı kullanıcısının ve bir tüccarın dijital sertifikalarının yetkili ve güvenilir bir üçüncü şahıs tarafından verildiğinin onaylanması sağlar. Açık Anahtar

Altyapısın'dan (PKI: Public Key Infrastructure), yani bir CA ile çalışan açık anahtar şifreleme kullanımından, e-ticaret içerisinde çok fazla yararlanılmaktadır. [Laudon ve Laudon – 2018]

Bölüm Özeti

Temel olarak İnternet üzerindeki haberleşmenin en önemli bileşenlerinden biri olan http ele alınmıştır. Bu protokolle ilgili detaylı inceleme yapılmış alt yapı ve mimarisile ilgili bilgiler verilmiştir. Ayrıca güvenli iletişim protokolü olan https de incelenmiş ve şifreleme yöntemleri ele alınmıştır.

Kaynakça

Alvestrand, H., “Tags for the Identification of Languages” RFC 1766, March 1995.

Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H.F., Secret, A. The World-Wide Web, Communications of the ACM, 37 (8), 76-82, 1994.

Braden, R., “Requirements for Internet Hosts -- Communication Layers,” STD 3, RFC 1123, Ekim 1989.

Horton, M., and R. Adams. “Standard for Interchange of USENET Messages,” RFC 1036, December 1987.

Laudon K. C., Laudon J. P., Management Information Systems, Prentice Hall, Dokuzuncu baskı, 2006.

Laudon K.C., Laudon J.P., Yönetim Bilişim Sistemleri, çeviri editörü: Prof: Dr. Uğur Yozgat, 12. Baskı, 2018.

Masinter, L., Sollins, K. Functional Requirements for Uniform Resource Names. RFC 1737, 1994.

Oz E., Management Information Systems, Thomson, Beşinci baskı, 2006.

Postel, J., “Media Type Registration Procedure,” RFC 1590, November 1996.

3. HTTP, HTTPS, HTML VE XML II

Birlikte Düşünelim

1. HTML nedir ve ne amaçla kullanılır?
2. Web uygulamaları geliştirilirken html nasıl kullanılır?
3. XML nedir ve ne amaçla kullanılır?

Başlamadan Önce

Web sayfalarında uzun süre yalnızca görünüm ön planda tutulmuştur. Başka bir ifadeyle verinin ne olduğundan çok kullanıcıya nasıl gösterildiği daha önemli bir konuydu. Fakat Internet gelişikçe ve kullanıcı sayısı dramatik olarak arttıkça verinin ne olduğunu da belirtmesine ihtiyaç duyulmaya başlandı. Bu nedenle html ve xml konuları çoğu zaman birlikte ele alınır. Bu bölümde de verinin html ile nasıl görüntüleneceği ve xml ile nasıl temsil edileceği ele alınacaktır. Bu nedenle önce Html' in temel özellikleri, mimarisi ve ana komutları incelenecuk, ardından da genişletilebilir işaretleme dili ile veri dosyalarının oluşturulması tartışılacaktır.

Giriş

Hiper Metin Biçimlendirme Dili (HTML), belgelerde metnin nasıl sunulması gerektiğini ve belgelerin birbirine nasıl bağlandığını gösteren etiketlerle işaretlemek için bir sistemdir. Tam olarak bir programlama dili değildir. Hiper metin bağlantıları oldukça güçlüdür. HTML işaretleme şemasında etkileşimli, platformlar arası, multimedya, istemci-sunucu uygulamaları oluşturma gücü yatkınlıkta. Bu tür bir sistem daha önce de ele alınmış olan World Wide Web'dir (WWW olarak da bilinir veya sadece Web'dir).

Web, biçimlendirilmiş metin, resim ve ses içeren birbirine bağlı canlı belgeler koleksiyonudur. Bu dokümanlar web alanında düzenlenir. Bir web alanı tipik olarak kendi alanının içinde ve dışındaki diğer sayfalara veya belgelere bağlantı içeren bir ana sayfanın etrafına yapılandırılır. Bir ana sayfa, bilgi alışverişi için siber alanda adeta sanal bir buluşma yeri olarak işlev görür. Ana sayfa HTML ile yazılır.

HTML ve World Wide Web hakkında bilgi içeren birçok giriş sayfası bulunmaktadır, HTML yapım halen gelişmekte olan bir dildir. HTML' nin devam eden gelişimi Web' de açık bir süreçte gerçekleştirilmektedir. Yeni araçlar ve teknikler sık sık ortaya çıkmakta ve Web yazarları topluluğu arasında hızla yayılmaktadır.

HTML bir programlama dili değildir ve bir HTML belgesi bir bilgisayar programı değildir. Bir bilgisayar programı, tipik olarak harici verilere uygulanan bir dizi prosedür ve talimattır. Bununla birlikte, bir HTML belgesi veridir. HTML verilere eklendiğinde, tarayıcılara (HTML belgelerini okuyan bilgisayar programlarına) belgeyi nasıl sunacağına ilişkin işaretleme etiketlerinin dilbilgisi ve sözdizimini belirtir.

XML, Genişletilebilir İşaretleme Dili'nin kısaltmasıdır. XML, World Wide Web Consortium (W3C) tarafından geliştirilen ve korunan bir genel standarttır. W3C bilgi, ticaret, iletişim ve ortak anlayış için bir forum olarak Web'in potansiyelini artırmak için birlikte çalışabilir teknolojiler (şartnameler, kılavuzlar, yazılım ve araçlar) geliştirir. XML, hızla Web üzerinde veri değişimi için standart hale gelen, metin tabanlı bir biçimlendirme dilidir. Verileri tanımlamak için tasarlanmıştır. XML, HTML'ye çok benzeyen bir biçimlendirme dilidir. HTML'de olduğu gibi, verileri etiketlerle (dik açılı parantez içine alınmış tanımlayıcılar) tanımlarız. XML etiketleri, XML'de önceden tanımlanmamıştır. Kendi etiketlerimizi tanımlamamız gereklidir. XML, verileri resmi olarak tanımlamak için bir DTD (Belge Türü Tanımı) kullanır. XML, verileri tanımlamak ve verinin ne olduğuna odaklanmak için tasarlanmıştır. HTML ise verileri görüntülemek ve verilerin görünümüne odaklanmak için tasarlanmıştır. HTML bilgi göstermekle ilgilidir, XML ise bilgi tarif etmeye ilgilidir.

3.1. HTML Etiketleri

HTML, dik açılı parantez (`<>`) ile çevrili bir etiket sistemi, bir sözcük veya kodlanmış komutlar olarak çalışır. Çoğu etiket, metni çevreleyen ve tarayıcı yazılımına veya bilgisayar sunucusuna bu metnin nasıl görünmesi gereği ve hangi işlevleri etkinleştirebileceği konusunda talimat veren bir ön ve arka biçim sahiptir. Tüm web sayfaları `<HTML>` etiketleriyle başlar ve biter. İlk `<HTML>` etiketinden sonra, `<HEAD>` `</HEAD>` etiketleri bulunur ve bunlar arasında tarayıcı ekranının üst çubuğunda görüntülenen sayfanın başlığını taşıyan `<TITLE>` `</TITLE>` etiketlerini içerir. Daha sonra `<BODY>` `</BODY>` etiketleri, sayfadaki tüm metnin önüne ve arkasına yerleştirilir ve gövde etiketi içerisinde metin ve arka plan renklerinin özellikleri belirtilir. Normal bir metin, metin olarak görüntülenir. Etiketler (aynı zamanda elemanlar da denir) özel talimatlardır. Etiketler, dik açılı parantez içindeki adları ile tanımlanır. Özel semboller (varlıklar da denir), & işaretleri ve tırnak işaretleri gibi kod noktalama işaretleridir. Böylece HTML işaretleme etiketleri açılı ayrıclarla sınırlanır,

`<.....>`

Metinde paragraf sonunu belirtmek için `<P>` etiketi gibi tek olarak veya içerdikleri içeriği değiştiren bir başlangıç ve bitiş etiketi çifti olarak görünürler. Örneğin `` Dikkat! `` Dikkat! metin dizesini gösterme talimatıdır ve kalın yazı tipinde olacağını belirtmektedir. Diğer örnekler, HTML dosyasına bir yorum olan

`<COMMENT>` Bu bir yorumdur `</COMMENT>`

ve metni veya resmi ortalayan

`<CENTER>` ... `</CENTER>`

Metin biçimlendirme etiketleri, köprü metni bağlantıları belirtme etiketleri, ses ve resim öğelerini dahil etme etiketleri, etkileşimli sayfalar için giriş alanları tanımlama etiketleri vardır. Buradaki tek şey Hipermekanın İşaretleme Dili, karakter varlıklarını ve işaretleme etiketleri. Ancak, bu varlık ve etiket sistemi gelişmektedir. HTML büyük / küçük harf duyarlı değildir.

3.1.1. Ana Komutlar

Yukarıda anlatıldığı gibi bir HTML programı (HTML komut dosyası olarak da bilinir) üç tür jeton dizisidir: normal metin karakterleri, etiketler ve özel semboller.

HTML' deki ana komutlar,

Yapı Etiketleri

<HTML> ... </HTML> HTML belgesinin tamamını kaplar

<HEAD> ... </HEAD> HTML belgesinin başını çevreler

<BODY> ... </BODY> HTML belgesinin gövdesini (metin ve etiketler) çevreler

Bir özellik

BGCOLOR = "..." biçiminde belirtilir.

Başlıklar ve Sayfa Başlığı

<H1> ... </H1> Birinci seviye bir başlık

<H2> ... </H2> İkinci seviye bir başlık

<H3> ... </H3> Üçüncü seviye bir başlık

<H4> ... </H4> Dördüncü düzey bir başlık

<H5> ... </H5> Beşinci seviye bir başlık

<H6> ... </H6> Altıncı seviye başlık

<TITLE> ... </TITLE> Dokümanın başlığını belirtir. <HEAD> ile kullanılır.

Tüm başlık etiketleri ALIGN = "..." özniteliğini kabul eder. Olası değerler CENTER, LEFT, RIGHT' tır.

Yorumlar

<COMMENT> Yorumlar </COMMENT>

<! - Yorumlar ->

Paragraflar ve Bölgeler

<P> ... </P> Düz bir paragraf. Kapanış etiketi (</P>) isteğe bağlıdır.

ALIGN = "..." özniteliği metni merkeze, sola, sağa göre hizalar.

<DIV> ... </DIV> Biçimlendirilecek bir metin bölgesi.

ALIGN = "..." özniteliği metni merkeze, sola, sağa göre hizalar.

Listeler

 ... Sıralı (numaralı) bir liste (ile işaretli öğeler)

 ... Sırasız (madde imli) bir liste (ile işaretli öğeler)

<MENU> ... </MENU> Öğelerin menü listesi

<DIR> ... </DIR> Bir dizin listesi

 , , <MENU>, <DIR> ile kullanım için bir liste ögesi

<DL> ... </DL> Bir tanım veya sözlük listesi

Metin Blokları

<BLOCKQUOTE> ... </BLOCKQUOTE> Başka bir kaynaktan alıntı

<ADDRESS> ... </ADDRESS> Belge yazarı tanımlaması için sıkılıkla kullanılan adres

Karakter Biçimlendirme

<CODE> ... </CODE> Kod örneği (genellikle Courier)

 ... Kalın yazı

<I> ... </I> Italik metin

<TT> ... </TT> Daktilo metni

 ... Vurgu (genellikle italicik)

<CITE> ... </CITE> Bir alıntı

<PRE> ... </PRE> Önceden biçimlendirilmiş metin (satır biçimini korur)

<BIG> ... </BIG> Metin normalden biraz daha büyük

<SMALL> ... </SMALL> Metin normalden biraz daha küçük

_{...} Alt simge

^{...} Üst simge

<KBD> ... </KBD> Klavye girişi

Diğer Elementler

<HR> Yatay bir kural çizgisi

Öznitelikler

SIZE = "..." Çizginin kalınlığı, piksel cinsinden

WIDTH = "..." Çizginin genişliği, piksel cinsinden

ALIGN = "..." Kural çizgisinin sayfada nasıl hizalanacağını belirtir

NOSHADE = "..." Kural çizgisinin düz bir çizgi olarak çizilmesine neden olur

 Bir satır sonu

<CENTER> ... </CENTER> Metin veya görüntüleri merkezler

<BLINK> ... </BLINK> Metnin dikkat çekici şekilde yanıp sönmesine neden olur

 ... Metnin yazı tipinin boyutunu değiştirir

Öznitelikler

SIZE = "..." Yazı tipinin boyutu

COLOR = "..." Metnin rengini değiştirir

FACE = "..." Kullanılacak fontun adı

Tablolar

<TABLE> ... </TABLE> Başlık (<CAPTION>) ve istediğiniz sayıda satır (<TR>) içerebilen bir tablo oluşturur

<CAPTION> ... </CAPTION> Tablonun başlığı

<TR> ... </TR> Başlıklar ve veriler içeren bir tablo satırı tanımlar.

<TH> ... </TH> Tablo başlık hücreğini tanımlar

<TD> ... </TD> Tablo veri hücreğini tanımlar

Görüntüler, Sesler ve Gömülü Ortamlar

 Belgeye satır içi bir resim ekler

Öznitelikler

SRC = "..." Resmin URL'si

ALIGN = "..." Verilen görüntünün hizalamasını belirler

VSPACE = "..." Resim ile üstündeki ya da altındaki metin arasındaki boşluk

HSPACE = "..." Resim ve metin arasındaki sola veya sağa boşluk

WIDTH = "..." Resmin piksel cinsinden genişliği

HEIGHT = "..." Resmin piksel cinsinden yüksekliği

Bağlantılar

<A> ... HREF özelliği ile başka bir belgeye veya dayanak noktasına bir bağlantı oluşturur. NAME özelliği ile dayanak noktası olabilecek bir bağlantı oluşturur.

Öznitelikler

HREF = "..." Bağlantı etkinleştirildiğinde çağrılacak belgenin URL'si

NAME = "..." Bağlantının adı

Formlar

<FORM> ... </FORM> Giriş formunu gösterir

Öznitelikler

ACTION = "..." Bu form girişini işleyen komut dosyasının URL'si

METHOD = "..." Form girişinin sunucu tarafındaki ağ geçidine nasıl gönderileceği. Olası değerler GET ve POST 'tur.

<INPUT> Belirli bir türe ve sembolik isme sahip giriş elemanı

Öznitelikler

TYPE = "..." Bu giriş bileşeninin türü. Olası değerler CHECKBOX, HIDDEN, RADIO, RESET, SUBMIT, TEXT, SENDER, FILE, IMAGE

NAME = "..." Bu ögenin adı

VALUE = "..." Bir metin veya gizli bileşen için varsayılan değer; bir onay kutusu veya radyo düğmesi için formla birlikte verilecek değer; Submit veya Send düğmeleri için düğmenin kendisinin etiketi

SRC = "..." Bir resmin kaynak dosyası

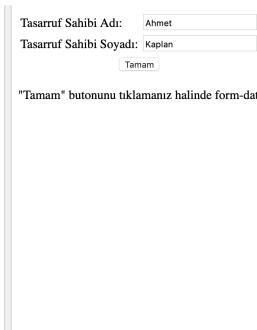
SIZE = "..." Bir metin bileşeninin karakter olarak boyutu

<TEXTAREA> Metin alanı (düzenlenebilir metnin satırları)

<SELECT> Sembolik isimle eleman seçme

<OPTION> SELECT ile kullanılan seçenek öğesi

Şekil 4' de buraya kadar ele alınan HTML komutlarının birkaçını kapsayan bir örnek ve tarayıcının komutları çözümlemesiyle oluşacak görüntü verilmiştir.



The screenshot shows a simple HTML form with two text input fields. The first field is labeled 'Tasarruf Sahibi Adı:' and contains the value 'Ahmet'. The second field is labeled 'Tasarruf Sahibi Soyadı:' and contains the value 'Kaplan'. Below the inputs is a 'Tamam' button. To the right of the form, there is explanatory text in Turkish: "'Tamam' butonunu tıklamanız halinde form-data sunucudaki '/action_page.php' isimli sayfaya gönderecektir.'

```
<!DOCTYPE html>
<html>
<body>
<form action="/action_page.php">
  <table>
    <tr>
      <td>Tasarruf Sahibi Adı:</td>
      <td><input type="text" name="Adı" value="Ahmet"></td>
    </tr>
    <tr>
      <td>Tasarruf Sahibi Soyadı:</td>
      <td><input type="text" name="Soyadı" value="Kaplan"></td>
    </tr>
    <tr>
      <td colspan="2" align="center"><input type="submit" value="Tamam"></td>
    </tr>
  </table>
</form>


'Tamam' butonunu tıklamanız halinde form-data sunucudaki '/action_page.php' isimli sayfaya gönderecektir.'


</body>
</html>
```

Şekil 4 HTML örneği

3.1.2. Diğer Belgelere Bağlanma

HTML, metin, dosya veya grafik nesnelerinin bazı bölgelerini belgemize bağlayabilmemizi sağlayan hiper metin bağlantılarını tanımlayabilir. Bunu HTML bağlantısı sekmesi üzerinden yaparız. Örneğin, bir HTML belgesindeki aşağıdaki pasajın, diğer HTML belgelerine iki bağlantısı vardır.

Uluslararası Ticaret Okulu

Londra Üniversitesi’nde

 diploma

İmkâni sağlamaktadır.

İlki, geçerli HTML belgesiyle aynı dizinde bulunması gereken SciSchool.html' dir. İlk bağlantı etiketi, belge için bilgileri içerir ve bu etiket ile arasındaki metin, Web tarayıcısı tarafından farklı bir renkte görüntülenir (normalde mavi). Kullanıcı fareyle bu vurgulanan metnin üzerine tıklarsa, tarayıcı ilgili belgeyi

alır. İkinci bağlantı etiketi, dizin hiyerarşisinde daha düşük bir dalda bulunan bir diploma isimli dizinde yer alan diploma.html dosyasıdır. Dolayısıyla, geçerli belge C:\webdocs\scschool dizinindeyse, Web tarayıcısı C:\webdocs\diploma\diploma.html dosyasını arayacaktır. Tüm bağlantıları bu şekilde göreceli yapmak iyi bir fikirdir, böylece web sitesini farklı bir yere taşırsak, tüm bağlantılar sağlam kalır. Ayrıca diğer web sitelerindeki belgelerle de bağlantılar kurabiliriz. Örneğin;

 ISSC

Elbette, başka bir web sitesinin ana sayfasından başka belirli bir sayfaya bağlantı da kurabiliriz. Büyük HTML belgeleri söz konusu olduğunda bu belgeler içindeki belirli bir bölüme atlamak çok faydalı olabilir. Bu, ilgili belgenin içinde adlandırılmış bir bağlantı sağlayarak ve # karakterini kullanarak adlandırılmış bu bağlantıya bir sıçrama belirterek gerçekleştirilir.

 Giriş

 İlk Bölüm

Kullanıcı bağlantıyı tıklarsa web tarayıcısı belgenin ilgili bölümünü gösterir.

<H3> Giriş </H3>

Bu tanımının gövdesidir ...

<P>

<H3> İlk Bölüm </H3>

Bu ilk bölümün gövdesi ...

3.2. XML

HTML belgelerini işaretlemek için kullanılan etiketler ve HTML belgelerinin yapısı önceden tanımlanmıştır. Örneğin, ... HTML etiketi metni kalın yazı tipi stilinde oluşturur. HTML belgelerinin yazarı, yalnızca HTML standardında tanımlanan etiketleri kullanabilir. XML ise yazarın kendi etiketlerini ve kendi belge yapısını tanımlamasını sağlar.

XML, verileri HTML' den ayrı tutabilir. XML, HTML belgelerinin içindeki verileri depolamak için kullanılabilir. XML bilgi alışverişinde bir format olarak kullanılabilir. XML, dosyalarda veya veri tabanlarında tarih depolamak için kullanılabilir. Böylece, sınırsız sayıda kendi özel etiket ve özniteligimizi oluşturmak, her bir etikete ve özniteligi bir veri türü atamak, etiketlerle ilişkilendirilmiş değerlere erişmek ve bunları oluşturduğumuz özel bir yapılandırılmış biçimde gerçekleştirmek için XML kullanabiliriz.

Yukarıda açıklandığı gibi, HTML sayfaları verileri görüntülemek için kullanılır. Veriler genellikle HTML sayfalarında saklanır. XML ile bu veriler ayrı bir XML dosyasında saklanabilir. Bu şekilde, biçimlendirme ve görüntüleme için HTML kullanmaya odaklanabiliriz ve temel verilerdeki değişikliklerin HTML kodumuzun hiçbir yerinde değişiklik yapmaya zorlamayacağından emin olabiliriz.

Ancak, XML verileri HTML belgelerinin içinde veri adaları olarak adlandırılan bir şekilde depolayabilir.

Bilgisayar sistemleri ve veri tabanları uyumsuz formatlardaki verileri içerir. En büyük sorunlardan biri, İnternet üzerinden bu sistemler arasında veri alışverişi yapmaktadır. Verileri XML'e dönüştürmek, bu karmaşıklığı büyük ölçüde azaltabilir ve farklı türdeki uygulamalar tarafından okunabilecek veriler oluşturabilir.

XML, verileri dosyalarda veya veri tabanlarında depolamak için de kullanılabilir. Mağazadan bilgi almak ve depolamak için uygulamalar yazılabilir ve verileri görüntülemek için genel uygulamalar kullanılabilir.

Bir XML belgesi için aşağıdaki koşullar yerine getirilmelidir.

1. Tüm XML öğelerinin bir kapanış etiketi olmalıdır.
2. XML etiketleri büyük / küçük harf duyarlıdır. Etiketlerin açılması ve kapatılması aynı durumda yazılmalıdır. Örneğin, bir XML çözümleyici <BOOK> ve <book> öğelerini iki farklı etiket olarak yorumlar. HTML etiketlerinin büyük / küçük harf duyarlı olmadığını hatırlayın.
3. Tüm XML öğeleri düzgün bir şekilde iç içe geçmiş olmalıdır.
4. Tüm XML belgelerinin bir kök etiketi olmalıdır. Bir XML belgesinin sıralı, etiketlenmiş bir ağaç yapısında olduğunu unutulmamalıdır.
5. Öznitelikler her zaman tırnak içine yazılmalıdır. XML öğelerinin, HTML' deki gibi ad / değer çiftindeki öznitelikleri olabilir. XML' de öznitelik değeri her zaman tırnak içine yazılmalıdır. XML özellikleri normal olarak XML öğelerini tanımlamak veya öğeler hakkında ek bilgi sağlamak için kullanılır.

Her XML belgesinin ilk satırı, aşağıda örneği verilmiş olduğu gibi XML bildirimi olmalıdır.

```
<?xml version="1.0" standalone="yes" encoding="ISO-8859-1"?>
```

Burada <? bir işleme talimatı başlatır. xml bunun bir XML komutu olduğunu belirtir. version, kullanılan XML spesifikasyonunun sürümünü belirtir. Sürüm 1.0, yalnızca geçerli sürümdür, bu nedenle değerin 1.0 olması gereklidir. standalone işaretlemesi, bu dosyanın tamamen kendi başına olup olmadığını belirtir. Bir no değeri, ayırtıcıya harici bir DTD (Belge Türü Tanımı) gibi diğer biçimlendirme dosyalarını içe aktarması gerektiğini bildirir. İyi biçimlendirilmiş XML belgelerinin kendisi tamamlanmış olduğundan, bu özellik değeri yes olur. encoding komutu hangi uluslararası karakter setinin kullanıldığını gösterir. ISO-8859-1 varsayılandır ve temelde İngilizce karakter setidir. ?> işaretti, işlem talimatlarının sonlandığını belirtir.

Yorumlar şu şekilde yazılır: <!--bu bir yorumdur -->

XML belgeleri bir kök ögesi gerektirir. Kök öge yalnızca bir kez görünür ve <HTML> ... </HTML> yazımının bir HTML belgesinin içeriğini içermesi gibi tüm XML öğelerini ve verilerini belgedeki içine alır.

Aşağıda bir örnek verilmektedir. İlk satır XML bildirimidir. Belgenin XML sürümünü tanımlar. Ayrıca, Kutuphane sadece bir kez görünen ve diğer tüm öğeleri içeren kök öğedir. XML' i herhangi bir dosya adı ve .xml uzantısıyla kaydederiz.

```
<? xml version = "1.0" bağımsız = "evet"?>
<? xmlstylesheet type = "text / xsl" href = "library.xsl"?>
<! - Dosya adı: library.xml ->
<! - address: file: // c: \ books \ java \ library.xml ->
<Kutuphane>
<Kitap>
    <baslik> SymbolicC ++ </baslik>
    <altbaslik> Bilgisayar Cebirine Giriş </altbaslik>
    <yazar> Tan Kiat Shi, Willi Hans Steeb, Yorick Hardy </yazar>
    <ISBN> 1852332603 </ISBN>
    <yayinci> Springer </yayinci>
    <sehir> Londra </sehir>
    <yil> 2000 </yil>
</ Kitap>
```

<Kitap>

<baslik> Klasik ve Kuantum Hesaplama </baslik>
<altbaslik>, C ++ ve Java Simülasyonları ile </altbaslik>
<yazar> Yorick Hardy ve Willi Hans Steeb </yazar>
<ISBN> 3764366109 </ISBN>
<yayinci> Birkhauser Publishing </yayinci>
<sehir> Basel </sehir>
<yil> 2001 </yil>

</ Kitap>

</ Kutuphane>

XML dosyasını bir Excel tablosu açısından ele alırsak, her <Kitap> ... </Kitap> bloğu Kutuphane tablosunda bir satır olacaktır.

$$p(x,y)=2x^3y^2 - 4y^5$$

Polinomu aşağıdaki XML şeklinde kodlanabilir.

<? xml version = "1.0"?>

<Polinom>

<Terim>

<katsayi> 2 </katsayi>

<Degisken>

<isim> x </isim>

<ussu> 3 </ussu>

</ Degisken>

<Degisken>

<isim> y </isim>

<ussu> 2 </ussu>

</ Değişken>

</ Terim>

<Terim>

<katsayi> -4 </katsayi>

<Degisken>

<isim> x </isim>

```
<ussu> 0 </ussu>
</ Degisken>
<Degisken>
<isim> y </isim>
<ussu> 5 </ussu>
</ Degisken>
</ Terim>
</ Polinom>
```

Bölüm Özeti

Bu bölümde sunucu – istemci arasındaki Web haberleşmesinin biçimlendirmeyi ve anlamlandırmayı sağlayan html ve xml ele alınmıştır. Bu dil ile ilgili detaylı inceleme yapılmış ana komutlar ve ve bağlantı kurmayla ilgili bilgiler verilmiştir. Ayrıca genişletilebilir işaretleme dili olan xml de incelenmiş ve örnekler verilmiştir.

Kaynakça

- Alvestrand, H., “Tags for the Identification of Languages” RFC 1766, March 1995.
- Horton, M., and R. Adams. “Standard for Interchange of USENET Messages,” RFC 1036, December 1987.
- Laudon K.C., Laudon J.P., Yönetim Bilişim Sistemleri, çeviri editörü: Prof. Dr. Uğur Yozgat, 12. Baskı, 2018.
- Masinter, L., Sollins, K. Functional Requirements for Uniform Resource Names. RFC 1737, 1994.
- Oz E., Management Information Systems, Thomson, Beşinci baskı, 2006.
- Postel, J., “Media Type Registration Procedure,” RFC 1590, November 1996.

4. FTP, RSS, BLOG, PODCASTING, ANINDA MESAJLAŞMA VE ÇEREZLER

Birlikte Düşünelim

1. FTP nedir? Önemini belirtiniz.
2. RSS nedir? E-ticaret için önemimi belirtiniz.
3. Blog’ların E-ticaret içerisindeki yeri nedir?
4. Podcasting’ın kullanım alanları nedir?
5. Anında mesajlaşmanın iş dünyasındaki yeri nedir?

Başlamadan Önce

Günümüzde pek çok kişi evinden kendine ait iş sahibi olabilmektedir. Bunların bazıları büyük çaplı işler

dahi yapabilmektedir. Tabiatıyla bu tür tek başına ve evden yapılan işler FTP, RSS, Blog, Podcasting ve cookie' ler hakkında genel kavramların bilinmesi ihtiyacını ortaya çıkarmaktadır. Zira bahsettiğimiz işler saydığımız teknolojilerle mümkün olabilmektedir. Bu nedenle öncelikle FTP' nin temel özellikleri ve yapısı hakkında bilgi verilecektir. RSS' in temel özellikleri ve kullanım alanları bu bölümün konularındandır. Ayrıca Blogların kullanım biçimleri de son derece önemli bir yer tutmaktadır. Belki de en önemli olan iş yapma türlerinden birisi olan Podcasting' in temel özellikleri bu bölümde ele alınacak bir diğer konudur. Genellikle İnternet üzerinde iş yapan herkesin bilgi sahibi olması gereken Cookie' lerin temel özellikleri ve yapısı hakkında da bilgi verilecektir.

Giriş

FTP, Dosya Aktarım Protokolü'nün kısaltmasıdır. Bir FTP sunucusu ve Internet üzerindeki bir FTP istemci bilgisayarı arasında dosya aktarmak için kullanılan bir protokoldür. FTP, en çok World Wide Web'den dosya indirmek için kullanılır. Dosyaları FTP sunucularına indirmek ve yüklemek için HTTP protokolüne alternatif bir seçenekdir.

RSS "Gerçekten Basit Dağıtım" anlamına gelir. Haber okuyucular olarak adlandırılan RSS programları, bilgisayarlarınızın belirttiğiniz web sayfalarına otomatik olarak bakmasını sağlar ve bu sayfaların ne zaman yeni bilgileri olduğunu size bildirir.

Blog, girişlerin (veya gönderilerin) ters kronolojik sıraya göre düzenlendiği çok basit bir web sayfası olarak tanımlanabilir. Bir blog genellikle herkese açıktır ve çoğu kişi oluşturmaktan özgürdür. Genellikle bir yazarın eseridir, ancak grup blogları nadir değildir. Bloglar ayrıca okuyucularla etkileşimde bulunma fırsatı da sunabilir.

"Podcasting" kelimesi, XML tabanlı bir RSS beslemesi kullanarak ses dosyalarının İnternet üzerinden dağıtıımı ile ilişkilendirilmiştir. Bireyler bir "podcast" e abone olabilir ve kişisel bilgisayarlarına indirebilir. Bir kişi bir podcast'e abone olduktan sonra, geliştirilip yayınlandıça otomatik olarak yeni şovlar alır. Bu dosyalar iPod, MP3 çalar, cep telefonu veya PDA gibi küçük taşınabilir cihazlara da aktarılabilir. Bazı podcasting siteleri podcasting'i netcasting olarak adlandırır.

Tipik bir Üniversite öğrencisine en sevdiği iletişim araçlarının sorulması halinde, birçoğunun anında mesajlaşmanın (IM) en çok tercih edilen seçeneklerden biri olduğunu yanıtlaması muhtemeldir. Dünya genelinde, üniversiteler giderek daha fazla teknoloji bağımlı hale geldikçe, IM'yi birincil iletişim kaynağı olarak kullanan öğrenci sayısı da hızla artmaktadır.

Çerezler, web sitesinden kişinin terminaline (genellikle tarayıcıya) gönderilen ve aynı kullanıcının tekrar ziyaret ettiği aynı web sitesinde tekrar kullanılmak için depolandıkları küçük metin dosyalarıdır. Bir çerez, kullanıcının sabit diskinden, bilgisayar virüsleri geçirmeyi sağlayan ya da e-posta adreslerini yakalayan hiçbir veriyi alamaz. Her bir çerez, kullanıcının web tarayıcısına özgüdür. Çerezlerin bazı işlevleri diğer teknolojilere aktarılabilir.

4.1. Dosya Aktarımı

Bilgisayar kullanımının ilk zamanlarında, interneti kullanmak için karmaşık komut setlerinin öğrenilmesi gerekiyordu. 1970'lerin başında icat edilen FTP, dosyaları sistemler arasında aktarmak için standart bir protokol oldu.

İnternet standarı için kullanılan FTP protokolleri, İnternet Mühendisliği Görev Gücü Komitesi tarafından bir dizi RFC (Yorum İsteği) resmi belgesi olarak hazırlanmıştır. 1971'de FTP protokolü RFC 114 yayınlanmıştır. Yıllar geçtikçe FTP, bu protokolü iyileştirmek için değişiklik yapan yeni sürümlerle revize edildi. RFC 959, 1985'te yayınlandı ve mevcut standart şartname haline geldi. RFC belgesi, FTP güvenliğini artırma ve yeni teknolojiler için destek eklemeleri ile günümüzde hala değiştirilmektedir.

Günümüzde dosya aktarım protokolünün (FTP) bir bilgisayardan diğerine dosya taşımak için kullanılan en yaygın standart olduğuna daha önce dechinilmiştir. Öyle ki artık tüm Web tarayıcılarında FTP uygulaması

gömülüdür. Başka bir ifadeyle bir sunucudan dosya almak isteyen herhangi bir kullanıcı bu işlemi doğrudan tarayıcı arayıcılığıyla yapabilir.

Böylece birçok işletme FTP uygulamalarını kullanarak maliyet ve zaman kaybından kurtulmaktadır. Örneğin e-posta aracılığıyla gönderilemeyecek kadar büyük dosyalar olabilmektedir. Böyle bir durumda personel, işletmenin sunucusuna FTP aracılığıyla istediği bir dosyayı aktarabilmektedir. Böylece ilgililer söz konusu dosyayı kullanabileceklerdir. Bunun tam tersi olarak sunucu üzerine konulmuş dosyaların kullanıcılar tarafından alınmasına da sıklıkla ihtiyaç duyulmaktadır. Örneğin artık birçok yazılım firması ücret karşılığı olarak kullanıcıların programları Web sayfalarından yükleyebilmelerini sağlamaktadır. Bant genişlikleri ne kadar artmış olursa olsun halen çok büyük yazılımların bu yöntemle temin edilmesi zor olabilir. Buna rağmen en azından böyle büyük yazılımların bazı güncellemlerini FTP aracılığıyla elde edilebilmektedir. Böylece yazılım firması programların saklama ortamına kopyalanması, saklama ortamının ambalajlanması ve kullanıcıya gönderilmesi gibi önemli maliyetlerden kurtulmaktadır. Hemen hatırlatmak gereklidir ki bu yöntemle donanımların sürücülerini ve çeşitli ürünlerin kılavuzları temin edilebilmektedir. Daha ötesi bazı Web sayfalarından ücret karşılığı müzik ve film dosyaları FTP aracılığıyla elde edilmektedir.

FTP özetle şunlar için kullanılır:

- Internette yayınlanmak üzere web sayfalarını web sunucularına yükleme
- Genel yazılım sitelerindeki dosyalara göz atma ve indirme
- E-posta ekleri için çok büyük olan dosyaları iki taraf arasında aktarma
- Kurum için görevlendirmeler gibi içerikleri bir FTP sunucusu üzerinden indirmek ve yüklemek
- Programların en son sürümlerinin yazılım geliştiriciler tarafından dağıtılması

4.2. Gerçekten Basit Dağıtım (RSS)

Bazı yazarlarca gerçekten basit dağıtım (really simple syndication veya RSS) olarak tercüme edilmiş olan bu terim orijinal karşılığının kısaltmasıyla daha çok kullanılmaktadır. Bu BT aslında XML ailesinin bir parçasıdır. Bilindiği gibi XML sayesinde Web sayfaları arasında veri paylaşımı kolaylaşmaktadır. Böylece RSS hizmetinden yararlanan bir kullanıcıya ilgili Web sayfasındaki çeşitli güncellemler gönderilmektedir. Aslında kullanıcı bu veriyi almaktadır demek daha doğru olacaktır. Zira kullanıcı bunu bilinçli olarak yapmasa da güncelleme sorgusunu kendi bilgisayarı onun adına yapmaktadır. Genellikle haber içerikli Web sitelerinde kullanılmaktadır. Kuşkusuz kullanıcının abone olduğu Web sitelerindeki güncellemler denetlenmektedir. Ayrıca bu denetim tüm Web sayfasının güncellenmesiyle ilgilenmez. Onun yerine belirlenmiş verilerdeki güncellemelere odaklanmaktadır. Örneğin bir haber sitesinde yalnızca hava durumu verisine odaklanılabilir.

Daha önce de debynildiği gibi, RSS programları haber okuyucu olarak da adlandırılır. İki tür haber okuyucusu vardır: çevrimiçi ve bilgisayar üzerinde. Her birinin avantajları ve dezavantajları vardır.

Doğrudan bilgisayarınıza yüklenen haber okuyucusu programları, belirlediğiniz web sayfalarını düzenli aralıklarla denetleyerek arka planda sessizce çalışır. Değişen bir web sayfası bulduklarında, ekranınızda Outlook'un ve diğer birçok e-posta programının olduğu gibi küçük bir bildirim açılır. Daha fazla ayrıntı görmek ve daha fazla özelliği etkinleştirmek için açılır pencereye tıklayıp ana programı açabilirsiniz. En büyük dezavantajı, izlemek istediğiniz web sayfalarının listesinin yalnızca programın yüklü olduğu bilgisayarda bulunmasıdır. Verileri güncellemenin ve birkaç bilgisayara senkronize etmenin yolları vardır, ancak bu biraz sakıncalıdır.

Google Reader gibi çevrimiçi çalışan haber okuyucularla, faturalarınızın listesi gerçekte bir sunucuda çevrimiçi kalır ve programa herhangi bir zamanda herhangi bir bilgisayardan erişilebilir. Birkaç bilgisayardaki listeleri güncellemeye gerek yoktur. En büyük dezavantaj, programın bilgisayarınızdaki arka planda çalışmaması ve işler değiştiğinde açılır mesajlar sağlanamamasıdır. Periyodik olarak web'e gitmeli, çevrimiçi haber okuyucunuza açmalı ve programda en son baktığınızdan beri nelerin değiştigini gözlemlemelisiniz. Değişiklikler genellikle programa en son girildiği günden bu yana değişen bildirimleri koyulaştırarak gösterilir. Ek olarak, veri dosyalarınız İnternet Servis Sağlayıcısının bilgisayarında tutulur ve bu bazlarının güvenlik endişelerine neden olabilir.

4.2.1. Neden E-posta Bildirimi Yerine RSS?

Fatura izlemeyi kullanan çok sayıda kişiyi, izlenen ortalama fatura sayısı ve olası tüm fatura durumu değişiklikleri ile birleştirdiğimizi düşünelim; böyle bir durumda çok sayıda e-postayı İnternet üzerinden oluşturma ve yayılama potansiyeline sahip olurdunuz. Çok sayıda e-posta yayınlayan bir internet kullanıcısı, kolayca bir spamçı olarak etiketlenebilir ve kara listeye alınabilir; RSS böyle bir riski ortadan kaldırır.

4.3. Web Günlüğü (Blog)

Orijinal karşılığı olan “blog” Web günlüğü anlamına gelen “Web log” teriminin kısaltmasıdır. Bir Web günlüğü belirli bir konu üzerine çeşitli günlükçilerin yorumlarda bulunmasını ve fikirlerini paylaşması için ortam sağlar. Ancak bunların çoğu eğlence amaçlıdır. Coğu günlük sitesi kolaylıkla tasarıma izin vermektedir. Hemen belirtmek gereki ki burada tasarım son derece sınırlıdır. Genellikle paylaşılmak istenen bilgilerin bir metin olarak günlüğe yerleştirilmesini kapsar. Günümüzde bazı organizasyonlar bu tür Web günlükleri oluşturarak çalışanların bunlara katılmalarını teşvik etmektedirler. Böylece yararlı fikirler elde edilebilmektedir. Ancak genellikle bunların amacından saplığı da gözlenmektedir. Zira Web günlükleri fazlasıyla gayri resmi ve denetimsizdirler.

Web günlüklerinin sunmuş olduğu bir başka hizmet ise *geri beslemedir* (track back). Böylece günlüğün konusuna başka Web sitelerinde değinildiyse bu siteler de günlükte gösterilmektedir. Coğu Web günlüğünde bu amaca yönelik bir düğme olur. Kullanıcı bunu tıkladığı takdirde açılan bir pencerede konuya değinilmiş siteler listelenir.

Organizasyonlar açısından bir başka önemli amaç için de Web günlükleri kullanılmaktadır. Bu günlüklerde her tür konuya ilgili yorum ve fikirlerin paylaşımı söz konusu olduğundan, organizasyonlar ve/veya onların ürünleriyle ilgili olanlar da paylaşımaktadır. Böylece pek çok organizasyon Web günlüklerini tarayan yazılımlar kullanmakta, kendi isimlerinin geçtiği günlükleri bir çalışanın incelemesini sağlamaktadır. Örneğin isimsiz bir günlükü Kryptonite bisiklet kilitlerini kalemlle kırabildiğiyle övünmüştür, bunu izleyen birkaç hafta içerisinde firma kilitleri geri çağrılmıştır [bkz. Oz-2006].

4.4. Ses Yayıncılığı (Podcasting)

Web günlükleriyle metin ve resim gibi görsel veriler paylaşılırken, ses yayıncılığı ile ses verisi paylaşılmaktadır. Ancak günümüzde yükselen bant genişlikleri sayesinde ses verisine görüntü verisi de eklenmiştir. Hatırlanacağı gibi RSS ile bazı verilerin güncellendiğinde otomatik olarak kullanıcı bundan haberdar edilmektedir. Bunlar da ekseriyetle haber sitelerinde gözlemlenmekteydi. Aynı haber sitelerinde ses verilerine de yer verilebilir. Bu verilerdeki güncellemelerden kullanıcıların haberdar edilmesi sağlanabilir.

Böylece ses yayıncılığına bir ses kaydının, bir Web sitesinden temin edilerek dinlenmesi denilebilir. Genellikle söz konusu ses kayıtları MP3 biçimindedir ki bu biçim günümüzde hemen her tür ses çalıcı tarafından desteklenmektedir.

Ses yayıncılığı farklı amaçlar için kullanılabilmektedir. Bunlardan birisi de radyo yayınlarının Web üzerinden yapılmasıdır. Örneğin bazı okullarda, derslerin ses kayıtları Web sitesine konulmakta, öğrenciler içeriğine bağlı olarak istedikleri dersleri, İnternet bağlantısı sağladıkları herhangi bir yerden dinleyebilmektedirler. Bunun için herhangi bir radyo lisans ücreti de gerekmemektedir.

4.4.1. Podcasting Türleri

Günümüzde Podcasting için en yaygın dosya formları ses veya videodur, ancak dosyalar, görüntüler ve PDF dahil olmak üzere herhangi bir dosya türünde de olabilir. Üç ana podcast türü vardır:

1. Sadece ses: Bunlar radyo gibidir. Genellikle mp3 biçimindedir. Hem PC'lerde hem de Mac'larda oluşturulabilir ve bir mp3 çalar kullanılarak kullanılabilirler. Bu, oluşturmak için nispeten kolay oldukları ve sonuçta elde edilen dosyalar genellikle küçük olduğu için popüler bir podcast biçimidir.

2. Sesli ve hareketsiz görüntüler: Bunlara gelişmiş podcast' ler de denir. Bunların en yaygın eğitim amaçlı kullanım biçimi, sesli bir parça içeren bir PowerPoint veya Keynote sunumudur. Bu tür podcast'lerin ana avantajları videoya kıyasla küçük dosya boyutları oluşu ve bölümlendirilebilmeleridir. Geliştirilmiş podcast'ler, izleyicilerin podcast'te görüntülemek istedikleri slaya doğrudan atlamasını sağlar.
3. Video: Bunlar TV' ye benzerler. Genellikle büyük dosya boyutlarındadırlar. Bunları oluştururken, yanlış formatı seçmek, dosyanın hem Mac' larda hem de PC'lerde oynatılmasını imkansızlaştıracağından sıkıştırma teknolojilerine dikkat etmek gereklidir.

4.5. Anında Mesajlaşma

Anında mesajlaşma aslında e-posta ile haberleşmeye benzetilebilir. Yalnız bunun çok önemli bir farkı karşı taraftaki alıcı çevrimiçidir. Böylece aldığı e-posta' ya gerekliyorsa anında cevap verebilir. Anında mesajlaşmayı gerçekleştirebilmek için her iki tarafta da bir yazılımın bulunması gereklidir. Bu haliyle bir FTP yazılımı gibi düşünülebilir. Çünkü mesajlaşma yazılımları yalnızca mesaj değil, aynı zamanda dosya aktarımını da sağlamaktadır. Bunun da ötesinde mikrofon ve kamera donanımlarının desteğiyle ses ve görüntü de aktarılabilir. Bunlar yalnızca iki kişi arasında gerçekleşen mesajlaşmalar biçiminde düşünülmemelidir. Zira bir grup arasında gerçekleşmesi mümkündür.

Gayri resmi yüz yüze iletişimim, organizasyonlarda karmaşık koordinasyon, problem çözme ve sosyal öğrenme dahil olmak üzere birçok önemli fonksiyona hizmet ettiği gösterilmiştir [Whittaker, Frohlich ve Daly-Jones - 1994]. Gayri resmi iletişimini desteklemek için ilk araçlar oluşturma girişimleri, ses ve video ortamlarına odaklanmıştır. Ancak, bu girişimler çekirdek kullanıcı görevlerine yönelik destek eksikliği, maliyet, gizlilik sorunları ve uygulama zorlukları dahil olmak üzere pek çok nedenden ötürü geniş çapta benimsenmemiştir [Whittaker - 1995]. Bunun aksine, anlık mesajlaşma modern toplum için büyük ilgi çekmiştir, çünkü gayri resmi iletişimini başarıyla destekleyen bir araçtır [Nardi, Whittaker ve Bradner - 2000].

Ücretsiz anında mesajlaşma uygulamaları genellikle bir veya birkaç sunucu üzerinden iletişimini sağlarlar. Buna göre AIM (AOL Instant Messenger), Yahoo! Messenger, MSN Messenger ve ICQ gibi bazı anında mesajlaşma uygulamaları milyonlarca kişiyi buluşturmaktadır. Bu sebeple de reklâm vericiler açısından çekiciliği artmaktadır. Günümüzde farklı uygulamaların birbirileşmesini sağlayacak ilerlemeler de kaydedilmektedir. Örneğin AIM kullanıcıları MSN Messenger kullanıcılarıyla mesajlaşabilmektedir. Ayrıca akıllı telefonlarda sıkılıkla kullanılan WhatsApp gibi uygulamalar da anında mesajlaşmanın en son geldiği noktadır. Dünya üzerinde en fazla kullanıcısı olan uygulamalar arasında yer almaktadır.

Anında mesajlaşma, ağırlıklı olarak eğlence ve sohbet amaçlı kullanılıyor gibi görünse de artık bazı organizasyonlar iş amaçlı kullanmaktadır. Farklı bölgelerde bulunan çalışanların anında mesajlaşma ile toplantı yapması sağlanabilir. Böylece işler önemli ölçüde hızlanacaktır. Ayrıca bilindiği gibi müşteriye destek önemlidir ve bu nedenle çoğu organizasyon telefon aracılığıyla (çağrı merkezleri) müşterilere destek sağlamaktadır. Bunlardan bazıları müşteriye destek vermeyi anında mesajlaşma yöntemiyle gerçekleştirilmektedir. Böylece telefon maliyetinin önüne geçilmektedir. Bunun da ötesinde Web üzerinden satış yapan bir firmada müşterilerin en önemli şikayetleri satış personeli eksikliğidir. Genellikle tüketiciler bir ürünü almadan önce bazı soruların cevaplanması isterler. İşte anında mesajlaşma böyle durumlarda tüketicinin satın almayı düşündüğü ürünle ilgili sorularını sorabilmesi için kullanılabilmektedir.

4.6. Çerezler (Cookies)

Çerezler Web' te gezinen bir kullanıcının bazı bilgilerini saklamak için başvurulan bir yöntemdir. Başka bir ifadeyle, kullanıcı bir defa, bir Web sitesini ziyaret ettiğinde bazı bilgileri saklanır ve bundan sonraki ziyaretlerinde bu bilgilerden yararlanılır. Söz konusu bilgiler kullanıcı bilgisayarında çerez (cookie) klasöründe saklanan bir dosyada tutulur. Bu dosya ilgili sitenin URL' sine karşılık gelen ve o site tarafından saklanması istenen bilgileri barındırır. Aslında tarayıcının ayarlar kısmında çerezlere izin verilip verilmeyeceğini içeren bir ayar olsa da genellikle kullanıcılar ayarı değiştirerek çerezleri engellemezler. Tarayıcı ilk kurulduğunda ayarlar kendiliğinden çerezlere izin vermektedir.

Çerezlerin e-ticaret açısından büyük önemi vardır. Her şeyden önce kullanıcının bazı bilgilerini defalarca girmesine gerek kalmaz. Çünkü bir defa girildiğinde bunlar saklanır. Hemen hatırlatmak gereklidir ki bankalar

gibi güvenliğin çok daha fazla önem kazandığı bağlantılar genellikle şifre gibi kullanıcı bilgileri cerezlerde saklanmaz. Buna rağmen e-posta kullanıcı bilgilerinin ve şifrelerinin saklandığı görülmektedir. Bu bilgiler cerez dosyası silinmediği sürece kullanıcı bilgisayarında saklanabileceği gibi anlık ta olabilir. Buna göre bazı cerezler bir oturum boyunca saklanmakta, oturum sonunda silinmektedir.

Cerezler, kullanıcının daha önce ziyaret ettiği bir sitede hep aynı reklamı görmesini engelleme amacıyla da kullanılabilir. Benzer biçimde bir kullanıcının daha önceki tercihlerine bağlı olarak ilgilendiği ürünleri görmesi sağlanabilir. Örneğin bazı kitap satıcıları; bir defa ürünleri inceleyen bir okuyucu tekrar o satıcının sitesine girdiğinde, okuyucunun ilgilendiği kitapları açılış sayfasında karşısına getirmektedir. Böylece site içerisinde kendisi aradığında gözünden kaçabilecek ürünler tüketicinin dikkatini çekerectir.

Cerezler yalnızca bir Web sitesinin tüketiciyle arasındaki ilişkiyi kayıt altına alabildiği gibi birçok Web sitesindeki davranışları da saklayabilir. Buna göre kullanıcıların, bir organizasyonun abonesi olan Web sitelerindeki davranışları saklanmaktadır. Böyle bir yaklaşım kötü amaçlı *casus programlarının* (spyware) da öncülüğünü yapmaktadır. Bir casus program kullanıcı bilgisayarında bulunan ve kullanıcının hareketlerini takip ederek bunları ağ üzerinden başka yerlere aktaran bir programdır. Bunlardan bazıları kullanıcının rızasını almaktır, ancak önemli bir kısmı ise kullanıcının haberini olmaksızın faaliyet göstermektedir. Günümüzde anti virüs yazılımları bu tür casus programlarının önemli bir kısmını engelleme yeteneğine sahiptir.

Bölüm Özeti

Bu bölümde e-ticaret için son derece önemli konular arasında yer alan dosya aktarımı, gerçekten basit dağıtım, web günlükleri, ses yayıncılığı, anında mesajlaşma ve cerezler incelenmiştir. Her bir başlık detaylarıyla ele alınarak tartışılmış ve örnekler verilerek konu anlatımı zenginleştirilmiştir.

Kaynakça

Nardi, B., Whittaker, S., & Bradner, E. Interaction and outeraction: instant messaging in action. Proceedings of CSCW '00, 79-88, Philadelphia, PA, 2000.

Whittaker, S. Rethinking video as a technology for interpersonal communications: theory and design implications. International Journal of Man-Machine Studies , 42 (5), 501-529, 1995.

Whittaker, S., Frohlich, D., & Daly-Jones, W. Informal. Proceedings of CHI '94 4, 131-137, Boston, MA, 1994.

5. BİR WEB SİTESİNİN KURULMASI I

Birlikte Düşünelim

1. Web sayfalarının Makro tasarımını denince ne anlıyorsunuz?
2. Web sayfalarının Mikro tasarımını denince ne alınıyorsunuz?
3. Web saflarının tasarımında görsel ve işitsel öğelerin önemi nedir?

Başlamadan Önce

Bu bölümde temel olarak Web sayfalarının tasarımlıyla ilgili genel kavramsal bilgi edinimi sağlanacaktır. Zira günümüzde İnternet ve e-ticaret denildiği zaman akla gelen web sayfalarıdır. Hatta başarılı e-ticaret'in çok önemli bir kısmı başarılı web sayfası tasarımına dayanmaktadır. Bu nedenle web sayfalarının tasarımının genel hatlarıyla ele alınması ve genel tasarım hatalarının öğrenilmesi öncelikli konudur. Devamında da yine çok önemli bir konu olan web sayfalarının içeriğinin tasarlanması bu bölüm içerisinde detaylarıyla ele alınacaktır.

Giriş

Bir web sitesi kuruluşunuz hakkında bilgilerin herkese yayınlanmasına yardımcı olabilir. Kuruluşunuz hakkındaki farkındalığı artırmaya, çalışanlarınıza veya gönüllülerinize bilgi aktarmanıza veya kaynak yaratma etkinliklerini duyurmanıza yardımcı olabilir. Bu doğrultuda, iyi organize edilmiş bir web sitesini hızlı ve kolay bir şekilde oluşturabilirsiniz.

Günümüzde çeşitli Web teknolojilerinden yararlanan organizasyonlar için en önemli e-ticaret unsuru kendilerine ait bir Web sitesinin olmasıdır. Öyle ki bazı işletmeler yalnızca kurdukları Web sitesi üzerinden faaliyet göstermektedirler. Hatta bazı bireysel kullanıcılar da kendi Web sitelerini kurarak kendilerine iş imkânları yaratmaktadır. Zira artık Web sitesi adeta bir kartvizit haline gelmiştir. Özellikle google.com gibi gelişen arama motorları sayesinde site içinde geçen sözcüklere bağlı olarak bir müşteriyle irtibata geçilebilmektedir.

Bilgi ve iletişim teknolojisi dünyasının genişlemesiyle, İnternet bugünlerde en önemli pazar haline gelmektedir ve derinden günlük hayatımıza girmektedir. İnternet, bilginin yaygınlaştırılmasının geleneksel politik, coğrafi, ekonomik ve kültürel engelleri aşmasını sağlayan tamamen yeni bir bilgi alışverişi yolunu temsil eder. Zamanında, ucuz, mükemmel etkileşim ve tutarlılık karakterleriyle yeni bir sosyal medya olarak web sayfaları internette önemli bir rol oynamaktadır.

Kimse bu pazarın büyüklüğünü söyleyemez, şu ana kadar sınırsız gözükmektedir ve herkes kendi web sitesini kurarak biraz kar etmek istemektedir. Ancak, web siteleri oluşturmak kulağa kolay geldiği gibi gerçekten de kolaydır, birkaç kod satırı ve birkaç resim basit bir web sitesi oluşturabilir, ancak bunu karlı hale getirmek zordur.

5.1. Web Sayfalarının Makro Tasarımı

Web tasarım, içeriğin (genellikle hiper metin veya hiper medya), bir web tarayıcısı veya başka bir web özellikle yazılım kullanılarak görüntülenen World Wide Web üzerinden bir son kullanıcıya sunulma biçimini kapsayacak şekilde kullanılan geniş bir terimdir. Web tasarımının amacı, web sunucularında bulunan belgeler ve uygulamalar dahil olmak üzere çevrimiçi içerik koleksiyonunu temsil eden bir web sitesi oluşturmaktır. Bir web sitesi metin, resimler, sesler ve diğer içerikleri içerebilir ve etkileşimli olabilir.

Web tasarım, bilgi mimarisi (navigasyon şemaları ve adlandırma kuralları), düzen veya sayfaların (sayfa şemaları dahil olmak üzere web sitesi) yapısı ile öğelerini ve markalandırma ile kavramsal tasarımları içerir.

Tüm web siteleri açık bir stratejiyle başlatılmalıdır, böylece ne ifade etmeye çalışıkları açıkça görülebilecektir. Strateji daha sonra tasarımın tanımlanmış hedeflere ulaşmasını sağlar.

Web tasarımında en sık karşılaşılan sorun tasarımcıların hareket etmemeleri ve web sitelerini kullanıcı bakış açısından deneyimlemeleridir. Genellikle, öyle olmadığı halde web sitelerinin aslında doğru web sitesi olduğunu düşünmeye yanlıtlılar. Web siteleri kullanıcılar için inşa edilmiştir ve tasarımcılar yalnızca, aşina oldukları için bu izlenime sahip olabilirler.

Birkaç site incelemek, nasıl tasarlandığını, ne kadar kolay geçilebileceğini vb. görmek adına faydalı olacaktır. Herhangi bir sayfanın kullanılabilirliğini göz önünde bulundurarak; tasarımcı kendisine bu siteyi kullanmanın iyi olup olmadığını sormalıdır.

5.1.1. Web Sitesi Tasarımında Genel Hatalar

Otomatik ses: Kullanıcılara daima müziği veya kaydedilmiş bilgileri kapatma seçeneğini verilmelidir. Ziyaretçilerin sesle veya sitenizdeki müziğinizle büyüleneceğini otomatik olarak varsaymayın.

Dönen veya yanıp sönen reklamlar: Kimse yanıp sönen reklamlarla dolu siteleri beğenmemektedir. Yanıp sönen şerit reklamlar, adeta insanları bir gösteriye çekmeye çalışan bir karnaval çırtıkanının eşdeğeridir. Mesajınızı duyurabilmek için insanlara bağırmınız gerekmek. Reklamların sınırlı, bunun yerine içeriğinizi bol tutulması gereklidir.

Gezilemez siteler: Hedef bilgilerinize ulaşmak için ziyaretçilerinizden asla "bağlantı halkaları" ndan geçmelerini beklemeyin. Kullanıcılar basit düşünmeyi sever. Ürününüzü, fiyatınızı veya hizmetinizi net, kesin ve bulması kolay hale getirmelisiniz. İçeriğinizi, ilkokul çocukların bile sitenizi anlayabileceği biçimde tasarllayın.

Aşırı pop-up'lar: Günümüzde pop-up'lar kaçınılmazdır. Ancak ziyaretçilerin sitenize ulaşmak için birden fazla pop-up'ı kapatması gerekiyorsa, ayrılabilir ve bir daha geri gelmeyebilirler. Aynı, ekran üzerinde zıplayan "kayan metin" veya "vurgular" (üzerine gelindiğinde büyük açıklama çıkar) için de geçerlidir. Herhangi bir şey kullanmak zorundaysanız, bir ziyaretçi sitenizi terk ettiğinde açılan bir pop-up ekleyebilirsiniz.

Ölü bağlantı ve 404 hata iletişiyle dolu bir sayfa: Bağlantılarınızı güncel tutun ve artık etkin olmayan bağlantıları kaldırın. Yeni linklerdeki hata mesajları gerçekten can sıkıcıdır.

Karanlık bir arka plan üzerinde koyu metin: Okuyucularınızın içeriğinizi bu şekilde okumaları için çaba harcamasını beklemeyin, çünkü okumazlar. İhtiyaç duydukları bilgileri başka bir yerde bulacaklardır.

Flash'ı dikkatli kullanın: Detaylı bir görsel açıklama gerektiren bir film siteniz veya bir ürününüz yoksa, Flash dosyaları karmaşık ve yavaş yüklenigidenden e-ticaret sayfalarında Flash kullanmamaya özen gösterin. Flash kullanmanız gerekiyorsa, dosyalarınızı olabildiğince küçük ve hızlı yüklenebilir olmasına dikkat edin. Sitenizi bir ürün satmak için kullanıyorsanız, yüksek kaliteli ve hızlı yüklenen fotoğrafları ve her bir ögenin dikkat çekici açıklamalarını kullanın.

Paragraflar arasında boşluk bırakmadan dolu metin blokları: Böyle bir tasarım basılı olarak işe yarayabilir, ancak Web üzerinden kullanışsızdır. İnsanlar çevrimiçi içeriği, çevrimdışı okudukları şekilde okumamaktadır. Sunumlarda da aynı şekilde izleyiciler kısa sürede fikir edinmek isterler. Koyu metin blokları uygulamak okuyucularınızı sıkır. İçeriğinizi net ve özlü hale getirin. Paragraflarınızı ayırin ve bol miktarda boşluk kullanın.

Kötü arama: tam metin arama motorları, yazım kurallarını, çoğul ifadeleri, kısa çizgileri ve sorgu terimlerinin diğer türevlerini kullanamadıkları için arama kalitesini azaltır. Başka bir ifadeyle "e-posta" ile "eposta" aramada iki farklı metin olarak ele alınacaktır böyle bir arama motorunda. Bu tür arama motorları herkesi rahatsız eder. Bununla ilgili bir diğer sorun, arama motorlarının sonuçları her bir belgenin önemi yerine, yalnızca içerdikleri sorgu terimlerini esas alarak önceliklendirmesidir. Arama motorunuz, özellikle ürünlerinizin adları gibi önemli sorgular için listenin en üstündeki "en iyi tahminler" kısmını çağırırsa çok daha iyidir. Başka bir ifadeyle içeriği terimlere göre değil terimle ilgisine göre önceliklendirme yapan arama motorlarına yer vermelisiniz. Gezinme başarısız olduğunda arama, kullanıcının hayat kurtarıcisıdır. Gelişmiş arama bazen yardımcı olsa da, basit arama genellikle en iyi sonucu verir ve kullanıcıların aradığı şey bu olduğundan, aranan terimin içine yazıldığı basit bir kutu olarak sunulması gereklidir.

Ziyaret edilen bağlantıların rengini değiştirmeme: Geçmiş navigasyonun iyi bir şekilde anlaşılmaması, kullanıcıların mevcut konumlarını ve durumlarını anlamalarına yardımcı olur, çünkü yolculuklarının doruk noktasıdır. Geçmiş ve şimdiki konumlarınızı sırayla bilmek, nereye gideceğinize karar vermeyi kolaylaştırır. Bağlantılar bu süreçte kilit rol oynar. Kullanıcılar önceki ziyaretlerinde sonuçsuz kaldıkları bağlantıları dışlayabilir. Ayrıca, geçmişte faydalı buldukları bağlantıları tekrar ziyaret edebilirler. Dahası, hangi sayfaları daha önce ziyaret ettiklerini bilmek, kullanıcıları aynı sayfaları tekrar ziyaret etmelerinden kurtarır.

Sabit yazı tipi boyutu: Basamaklı stil sayfaları (CSS) ile, içerik gönderme formlarına önemli miktarda esneklik sağlanabilir. Aynı zamanda bu, web sitelerine Web tarayıcısının "yazı tipi boyutunu değiştir" düğmesini devre dışı bırakma ve sabit bir yazı tipi boyutu belirleme imkânı da verir. Ve çoğu zaman, bu sabit boyut küçüktür, bazı insanlar için okunabilirliği önemli ölçüde azaltır.

Düşük arama motoru görünürlüğüne sahip sayfa başlıkları (SEO): Arama, kullanıcıların aradıkları web sitelerini keşfetmeleri için en yaygın yoldur. Arama ayrıca kullanıcıların bireysel web sitelerinde yollarını bulma yöntemleri arasında en önemlilerinden biridir. Mütevazi bir sayfa başlığı, arama listelerinden yeni ziyaretçiler çekmek ve mevcut kullanıcılarınızın ihtiyaç duydukları belirli sayfaları bulmalarına yardımcı olmak için ana aracınızdır. Sayfa başlıkları, kullanıcılar bir siteyi yer imlerine eklediğinde sık kullanılanlar içerisindeki varsayılan giriş olarak kabul edilir. Örneğin ana sayfanız için, şirket adıyla başlayın, ardından sitenin kısa bir tanımını yapın. "H" altında alfabetik olarak sıralanmak istemediğiniz sürece, "Hoş Geldiniz"

gibi kelimelerle asla başlamayın. Ana sayfa dışındaki diğer sayfalarınız için, bu sayfada kullanıcıların ne bulacağını açıklayan en belirgin bilgi taşıma kelimelerinin birkaçıyla başlığa başlayın. Sayfa başlığı tarayıcıda pencere başlığı olarak kullanıldığından, Windows altındaki görev çubuğu o pencerenin etiketi olarak da kullanılır. Bu, ileri düzeydeki kullanıcıların, her sayfa başlığının ilk bir veya iki kelimesinin rehberliğinde birden çok pencere arasında hareket edeceği anlamına gelir.

Tasarım geleneklerini ihlal etme: Tutarlılık, en güçlü kullanılabilirlik ilkelerinden biridir. İşler her zaman aynı olduğunda, kullanıcılar ne olacağının konusunda endişelenmek zorunda kalmazlar. Bunun yerine, daha önceki deneyime dayanarak ne olacağını bilirler. Kullanıcıların beklentileri ne kadar çok haklı çıkarsa, sistemin kontrolünü o kadar fazla hissedeceler ve o kadar çok begeneceklerdir. Sistem kullanıcıların beklentilerini ne kadar fazla kırarsa, kendilerini o kadar güvensiz hissederler.

5.2. Web Sayfalarının Mikro Tasarımı

Web sitesi temel elemanlar topluluğu olarak tanımlanabilir, bu elemanların tasarımı ve üretimi web sitesi yapımının temel aşamasıdır.

5.2.1. Görsel ve İşitsel Öğeler

Bu elemanların çoğu, herhangi bir harici program veya modül olmadan tarayıcının kendisi tarafından işlenebilir. Örneğin, çoğu tarayıcı PNG, JPG ve GIF formatını görüntüleyebilir, FLV ve SWF gibi video dosyaları da tarayıcılar tarafından oynatılabilir, ancak mp3 gibi müzik dosyalarının yerel diske indirilmesi ve uygun harici programla çalıştırılması gereklidir. Tarayıcı eklentileri, multimedya dosyalarının daha fazla formatını çababilir. Örneğin Microsoft IE çekirdek tarayıcısı, multimedya sayfaları için OLE-ActiveX tabanlı teknoloji sağlar. ActiveX artık popüler bir teknoloji haline gelmiştir. Diğer bir multimedya teknolojisi Java Applet'tir. Eklentilere ve ActiveX'e kıyasla daha küçük olan web uygulamalarında kullanılır ve Java dilinde yazılmıştır. Java Applet daha fazla esnekliğe ve iyi bir platform bağımsız özelliğine sahiptir.

5.2.1.1. Metin

Metnin görsel iletişimde birincil işlevi, bilgiyi iletmek ve hedefe ulaşmaktır. Metnin tasarımında karmaşık bozukluktan kaçınmalıdır, bu da insanların kolayca tanıyalabileceği ve anlayabileceğini anlamına gelir.

Yazmanın temel amacı, yazarın temayı ve fikirlerini ifade etme biçimini iletmek için daha iyi ve daha etkili tasarım yapmaktadır. Örneğin, hiç kimse büyük miktarda karmaşık metin içeren bir web sayfasını okumaya istekli değildir.

Yazı tipi boyutu bir diğer önemli konudur. Punto ve piksel gibi farklı şekillerde hesaplanabilir. Bilgisayar tabanlı piksel teknolojisinin yazdırma birimine göre puntoyu dönüştürecekinden, punto önerilir.

Sayfa gövdesi için en uygun yazı tipi boyutu 12 puntodur. Daha fazla görüntülenmesi gereken sayfalarda genellikle 9 punto büyüğünde yazı tipi kullanılır. Başlığı veya diğer yerleri vurgulamak için daha büyük yazı tipi, altbilgi ve destekleyici bilgiler için daha küçük yazı tipleri kullanılabilir.

Satır aralığındaki değişikliklerin, metnin okunabilirliği üzerinde de büyük etkisi olacaktır. Genel olarak, satır boşluğunu yazı tipi boyutuna yakın ayarlamak uygundur. Düzenli aralık oranı 10:12'dir. Bu, satır aralığı 10 punto yazı tipi büyüğüğe 12 punto satır aralığı anlamına gelir.

Düzenli boşluk bırakma, izleyicinin dikkatini çekmek için kayda değer bir boş alan oluşturur, ancak çok geniş aralıklar süreklilik kaybeder.

Web tasarımda, tasarımcılar metni, metin bağlantılarını, ziyaret edilen bağlantıları seçilen bağlantıları ve çeşitli renklerle ayarlayabilir. Soğuk ve sıcak renklerin parlaklığını, saflığını ve karşılaştırmasını içeren metin rengi kontrastının önemini unutmayın. Bunlar sadece metnin okunabilirliğinde bir rol oynamamaktadır. Daha da önemlisi, istenen tasarım etkisini elde etmek için renk kullanılabilir.

5.2.1.2. Arka Plan

Web sayfası tasarımının arka planı, özellikle kişisel ana sayfa için çok önemlidir. Bir evin arka planı, bir odanın duvarına eşittir; iyi bir arka plan, ziyaretçilere yalnızca iyi izlenimler bırakmaz, aynı zamanda ziyaretçilerin içeriği kabul etmesini de etkiler. Farklı sayfalara sahip bir sitenin bile çeşitli arka plan tasarımlına sahip olacağını rahatlıkla görülebilir.

Düz renk arka planı basittir. Ancak, en yaygın kullanılanıdır, çünkü arka plan resmine kıyasla, ekran hızı avantajına sahiptir. Sayfa dosyası, genellikle sayfanın arka plan rengini belirlemek için <body> etiketini kullanır. HTML sözdizimi şöyledir:

```
<body bgcolor = "color">
```

"color", farklı renkler anlamına gelir ve çeşitli renkler kullanabilirsiniz. En kolay yol, "blue", "yellow", "black" vb. gibi genel İngilizce adlarını yazmanızdır. Aynı zamanda #0000FF, #FFFF00, #000000 vb. gibi onaltılık renk kodlarını da kullanabilirsiniz

Dalgalanma arka planı sadece aynı görüntülerin birleşimi ve tekrarıdır, bu tür arka planda, kum taneciği yaygın olarak kullanılır. Yeni başlayanlar her zaman böyle bir deneyimi yaşarlar. Kendi resimlerini sayfaların arka planı olarak koymaya çalışırken, tarayıcıda görüntülenenler yalnızca bir fotoğraf değil, aynı görüntünün tekrarlı düzende yatay ve dikey yönde de gösterilmesidir. Bu, tarayıcının arka plan görüntüsü ile nasıl ilgilendiğini gösterir. Bu kural nedeniyle, arka plan sayfası olarak küçük boyutlu bir resim kullanabiliriz, otomatik olarak tüm sayfayı tekrarlar ve kapsar, böylece web sayfalarının indirme süresini büyük ölçüde azaltır.

Şerit arka planı ve dalgalanma arka planı oldukça benzerdir. Sayfa arka planı için yatay veya dikey yön olarak uygulanır ve tekrar tekrar düzenlenir. Dalgalanma arka planından tek farkı tek yönde tekrarlanmasıdır.

Web sayfalarının tasarımlında yaygın olarak kullanılan sayfaların arka planı olarak fotoğraf eklenebilir.

5.2.1.3. Resimler (Düğmeler, Simgeler)

Düğme tasarımını genellikle göz ardı edilir. Ancak, iyi tasarlanmış bir düğme sitenizi genel olarak profesyonel yapabilir. Düğmelerin renkleri, kalan renkler ile karşılaşıldığında belirgin olmalıdır, yüksek kontrast ve parlak renkler düğmeler için sıkılıkla kullanılır. Şekil 5' te örneği görülmektedir.



Şekil 5 facebook için Kaydol düğmesi

Düğmeler ve simgeler, ekranın sol üst kısmı, başlık, gezinme alanı gibi kullanıcıların kolay bulabilecekleri yerlere yerleştirilmelidir.

Düğmelere yazılan metinler de tasarımın önemli bir parçasıdır. Açık ve kısa olması metnin amacıdır ve her zaman, kayıt ol, indir, dene gibi bir fiil ile başlar.

Düğme üzerindeki metin çekici hale getirilebilir, örneğin “ÜCRETSİZ” sihirli kelimesi genellikle düğmelerde kullanılır.

Düğmenin boyutu genellikle ne kadar önemli olduğuna bağlıdır, gözler ne kadar büyük olursa o kadar önemli olur.



Şekil 6 Netflix düğmesi

Farklı ekranların çeşitli boyut ve DPI değerlerine sahip olması nedeniyle, simgelerin boyutunun ekranı da ayarlaması gereklidir.

Bir web sitesinin tüm unsurları kullanıcıların tarayıcıları tarafından indirilir, böylece bu resimlerin ve simgelerin dosya boyutu indirme hızını doğrudan etkiler. Günümüzde JPEG (JPG), BMP, PNG, GIF vb. gibi çeşitli resim biçimleri kullanılmaktadır. Tüm bu türler tarayıcılarda görüntülenebilir. Web siteleri için JPG, GIF ve PNG yaygın olarak kullanılır. Hem JPG, hem de GIF formatı, diğer formatlara kıyasla aynı görsel efekt seviyesinde daha küçük olacak şekilde büyük bir sıkıştırma derecesine sahiptir. GIF veya JPG kullanılıp kullanılmayacağı renk miktarına bağlıdır. GIF'in daha az renkli resimler için uygun olduğu söylenebilir.

PNG formatı saydam bir arka plan verebilir ve bu onu kullanmanın temel nedenidir. Saydam arka plan sayesinde başka bir görüntünün üzerine getirildiğinde arka plan otomatik olarak görüntü olacaktır.

5.2.1.4. Renkler

Web üzerindeki renk, web tasarımlarını iyileştirmenin mükemmel bir yoludur. Bir bilgisayar ekranı, üç gruba ayrılmış binlerce küçük kırmızı, yeşil ve mavi noktadan oluşur. Bu küçük noktalar birbirine çok yakındır. O kadar yakın ki onları gerçekten nokta olarak görmüyoruz. Ama daha ziyade gözlerimiz homojen bir renk oluşturmak için onları karıştırıyor.

Spektrumdaki tüm renkler bu nokta sistemiyle üretilebilir. Örneğin, kırmızı ve yeşil noktalar kuvvetlerinin %100'ünde parlıyorsa, mavi hiç yanmıyorsa, sonucu saf sarı bir renk olarak görürsünüz.

HTML renkleri, kırmızı, yeşil ve mavi ışığın bir kombinasyonu olarak görüntülenir. Renkler için kullanabileceğiniz bir RGB kodunu almak için her değer için 0 - 255 arasında herhangi bir sayı kullanabilirsiniz. Bu da seçim yapabileceğiniz 16 milyondan fazla renk verir (256 x 256 x 256). Modern monitörlerin çoğu en az 16.000 renk gösterebilir. Eğer aradığınız rengi bu kadar seçenek arasında bulamıyorsanız, muhtemelen öyle bir rengin olmadığı söylenebilir.

Sayfalarınıza renk eklemenin farklı yolları vardır. En kolay yollardan biri, adlandırılmış renkleri kullanmaktır. Bunlar kullanıcıların düşündüğü gibi: kırmızı, mavi, yeşil ve mor gibi renk isimleridir. Ancak bazı tarayıcılar renk adlarına izin vermez veya bunları diğer tarayıcılardan farklı olarak oluşturur. Bu nedenle, renk adlarınız için onaltılık kodları kullanmak daha iyidir.

Onaltılık kodları renkleri tanımlayan kodlardır. Bunlar üç adet onaltılık (taban 16) sayı grupları şeklindedir. İlk sayı kırmızı, ikinci yeşil ve üçüncü mavidir. Her onaltılık sayı bir RGB üçlüsüdür. 00 en düşük renk tonu ve FF en yüksek renk tonudur. Örneğin, kırmızı ff0000, mavi 0000ff, yeşil 00ff00 ve mor 990099'dur.

Renk eklemek için CSS' yi kullanmak oldukça kolaydır. Belgelerinize renk koymanın en yaygın yöntemi <style> özniteligidinden yararlanmaktır. Metninizin olmasını istediğiniz rengi seçin, bir etiketi içine alın ve bir renk atamak için style özelliğini kullanın.

Renk eklemenin başka bir yolu, tablolarda arka plan rengi olarak kullanmaktır. Arka plan rengi stilini tabloya, tr, td veya th etiketlerine eklemeniz yeterli olacaktır.

Sayfanızı 3-5 veya daha fazla renkle doldurmak cazip gelebilir, ancak hepsi aynı renk tonu olsalar bile, bir sayfada ne kadar çok renk varsa, bazı durumlarda yanlış görünümleri o kadar olasıdır. Bu nedenle basit tutmak ve az renk kullanmak daha sağlıklı olacaktır.

5.2.1.5. Web Sitesi Gezintileri

İyi bir web sitesi gezintisi oluşturmak, bir web tasarımcısının web tasarım sürecinde başarması gereken en önemli görevdir. Web sitesi gezintisi, insanların siteler arasında gezinmek için kullandıkları yoldur. İyi yapılandırılmış, kullanımı kolay ve sezgisel olmalıdır. Kötü bir gezinti tasarımlı, kullanıcılaraya yardımcı olmaz ve sık sık, sitenizin diğerlerinden daha az erişilebilir olduğu izlenimi verir.

İyi gezinme, iyi web tasarımlı için çok önemlidir. Hem ticari sitelerde hem de bilgilendirme sitelerinde, kullanıcılar kolayca bilgi bulabilmelidir. Gezinmin kullanımını kolay veya sezgisel değilse, kullanıcılar bilgi aramada hızlıca başka bir yere giderler. Kullanıcı olarak sonraki adımda ne yapacağımızı bilmeden kendimizi sık sık web sitelerinin önünde görürüz. Gezinti o kadar iyi saklanmıştır ki bazı kullanıcılar onu nasıl kullanacaklarını bilmezler. Gezinme, erişilebilir ve kullanılabilir web siteleri oluşturmada en önemli unsurdur.

Çoğu zaman, insanlar bilgilerle dolu bir site oluşturur ve onunla herhangi bir şey yapmadan önce siteyi çalışacak şekilde sunarlar. Bilgilerinizi mantıklı bir yapı içinde düzenlemek gerekli olsa da insanların en basit ve en açık yapılardan başka bir şeyi anlamalarının zor olduğunu unutmayın.

5.2.1.5. Müzik

Temel olarak, web sayfasına <bgsound> etiketi veya <embed> etiketi ile müzik eklemenin iki yolu vardır.

<bgsound> "balance" sol ve sağ kanal dengesini, "delay" sesi çalmayı geciktirmeyi, "loop" tekrar sayısını, "volume" ses seviyesini ve "src" dosyanın yolunu belirten 5 özellik içerir. Bu, arka plan müziği eklemenin en yaygın yoludur ve MP3, WAV ve MID gibi çeşitli müzik dosyası türlerini destekler.

Web sayfasına müzik yerleştirmenin başka bir yolu da <embed> etiketini kullanmaktadır. Bu, bazı müzik eklentileriyle birlikte <bgsound> 'dan daha güçlündür. Bir web oynatıcı biçiminde görüntüleyebilir.

<embed> öğesinin özellikleri de <bgsound> öğesinden daha fazladır. Örneğin, kod <embed src = "music.mp3" autostart = "true" loop = "true" hidden = "true"> </embed> gibi olabilir.

Bölüm Özeti

Bu bölümde e-ticaret için belki en önemli konu olan web sitelerinde sayfa tasarımlı ele alınmıştır. Sayfa tasarımlı önce genel bir perspektiften incelenmiştir. Böylece sayfaların makro tasarımlı tartışılan ilk konu olmuştur. Daha sonra sayfalardaki içeriği en başarılı şekilde müşteriye vermek için dikkat edilmesi gereken hususlar ele alınmış ve mikro sayfa tasarımlı tartışılmıştır.

Kaynakça

Nantel J., Glaser E., "The impact of language and culture on perceived website usability", J. Eng. Technol. Manage. 25, 112–122, 2008.

Oz E., Management Information Systems, Thomson, Beşinci baskı, 2006.

Tang Q., Huang J., "Impact of Web Site Functions on E-Business Success in Chinese Wholesale and Retail Industries", Tsinghua Science and Technology 13, 368-373, 2008.

6. BİR WEB SİTESİNİN KURULMASI II

Birlikte Düşünelim

1. Bir web sitesi oluşturulurken yapılan planlamada hangi adımlar izlenir?
2. Bir web sitesinin tasarılanması ve oluşturulması denince ne anlıyorsunuz?
3. Bir web sitesinin fiziksel olarak kurulması ne demektir? Nelere dikkat edilmelidir?

Başlamadan Önce

Bir Web sitesi ziyaretçilerinin dikkatini çeken, sağlam bir kullanıcı deneyimi sağlayan ve işi mümkün olan en iyi biçimde sunan iyi bir web sitesi oluşturmaktır. Bu nedenle başarısız bir sonuç tüm yatırımin boş gitmesi anlamını taşıyacaktır. Diğer adımlardan herhangi birini tamamlamadan önce, web sitenizin nasıl olacağı hakkında iyi bir fikre sahip olunmalıdır. Bu nedenle harekete geçmeden önce etkili bir planlanmanın önemi bu bölümde izah edilecektir. Kavramsal tüm tasarım planlama safhasında tamamlandıktan sonra kullanıcıların erişebileceği fiziksel bir ortam gereklidir. Bunun da ziyaretçi sayısına etkileri olacaktır. Dolayısıyla fiziksel kurulumun önemini kavranması bu bölümün kazanımlarındandır.

Giriş

Bir web sitesinin nasıl oluşturulacağını öğrenmek kolaydır. Ziyaretçilerinizin dikkatini çeken, sağlam bir kullanıcı deneyimi sunan ve işinizi mümkün olan en iyi biçimde sunan iyi bir web sitesi oluşturmak biraz daha fazla iş gerektirecektir.

Temelde, bir web sitesi oldukça basit görünür. Genellikle, birkaç sayfa metin ve web tarayıcınızda görüntülenen bazı resimlerden biraz daha fazlasıdır.

Web sitenizi oluştururken, mümkün olan en iyi siteyi oluşturmanıza yardımcı olacak bazı temel bilgilere sahip olmak önemlidir.

Bu kısımda, başarılı bir şekilde ziyaretçilerinize hizmet edecek bir site oluşturmanıza yardımcı olacak birkaç planlama önleminden başlayacak ve en sonunda web sitelerinin fiziki olarak kurulumu ele alınacaktır.

6.1. Planlama

Diğer adımlardan herhangi birini tamamlamadan önce, web sitenizin nasıl olacağı hakkında iyi bir fikriniz olmalıdır. Diğer web tasarım adımlarına başlamadan önce aşağıdaki sorular dikkatlice düşünülmelidir:

- Web sitesinin amaçları nelerdir?
- Web sitesinin ana konusu nedir?
- Neden bir web sitesi yapmak istiyorsunuz?
- Hedef kitlenizin web sitesine karşılık olarak ne yapmasını istiyorsunuz?

Web sitenizin tasarımını ve içeriği yukarıdaki sorulara verdığınız cevapları yansıtmalıdır.

Web sitesi planlamasının bir diğer önemli yönü de devam edegeen bakımdır. İçeriği sık sık değiştirerek bilgileri güncel tutmaya çalışmalısınız.

Bir web sitesi yürütmek için yapmanız gerekenler:

- Bir alan adı edinin.
- Sitenizi barındırmak için bir web sunucusunu güvenlikli hale getirin.
- Web sitesi oluşturmak için güvenli bir yazılım edinin.
- Web sitesini tasarlın ve oluşturun.
- Web sitesi kullanımını takip edin. Web Analytics belgesini düzenli inceleyin.

6.1.1. Alan Adı Edinme

Alan adı, web sitenizin içeriğini yükleyebileceğiniz Internet sunucusu için “hatırlanması kolay bir takma addır”. Ayrıca, kişilerin web sitenize ulaşmak için kullanacakları web sitesi adresinin bir parçasıdır.

Uygun şekilde bir alan adı seçin. Alan adınızın aşağıdaki gibi olduğundan emin olarak web sitenizin ziyaretçiler için kolayca bulunabilir olmasını hedeflemelisiniz:

- Kısa
- Açıklayıcı
- Hatırlanması kolay (alan adlarınız, kuruluş adınıza mümkün olduğunda benzer olmalıdır)
- Tire içermeyen
- Sayı içermeyen
- Anlamlı
- Benzersiz
- Anahtar kelimeleri kullanan

İdeal durum, kuruluşunuzun kendi adının aynı zamanda alan adı olmasıdır:

- İnsanlar alan adınızı kolayca hatırlayabilecekler. Böylece sadece kuruluşunuzu yazacak ve web sayfanıza ulaşmak için bir uzantı ekleyecektir.
- Bu alan adı başka bir kuruluş tarafından tescilli olmayacağından, alan adına ilişkin haklara da sahip olacaksınız. Unutmayın ki Web siteniz ticari markalı ifadeler veya kelimeler kullanıyorsa, bu ifadeye sahip olan şirket tarafından kolayca dava edilebilir.

Alanı adlarına örnekler Google.com, İstanbul.edu.tr, Meb.gov.tr, Walmart.com biçiminde verilebilir.

İstediğiniz alan adına karar verdikten sonra, adın ICANN (Internet Tahsisli Sayılar ve İşimler Kurumu) organizasyonuna kayıtlı olması gereklidir. Bunu bir etki alanı adı kayıt şirketi aracılığıyla yapabilirsiniz. Alan adınızın kullanımı için kayıt şirketine yıllık ödemeler yapılır. Popüler kayıt şirketleri Register.com ve GoDaddy.com'dur. Eğer Web sitenizi bir Anasistem hizmet sağlayıcısında barındırıdysanız, bu barındırma organizasyonu alan adınızı da kaydedebilir.

Kuruluşunuz için uygun etki alanı uzantısını kullanın. Yaygın uzantılar aşağıda listelenmiştir. Ziyaretçilerin üç uzantıda da web sitesini bulabilmesi ve böylece web sitenizin daha fazla tanıtım alabilmesi için üç uzantıyı da kaydettirmeniz gereklidir.

- .org – kâr amacı gütmeyen kuruluşlar ve diğer kuruluş türleri için kullanılır.
- .com - en yaygın olanıdır.
- .net - genellikle internet servis sağlayıcıları, web anasistem hizmet sağlayıcıları veya internet altyapısına dahil olan diğer işletmeler tarafından kullanılır; ayrıca, işletmeler tarafından intranet web siteleri için sıklıkla kullanılır.

6.1.2. Web Sitenizi Barındıracak Bir Web Sunucusunu Güvenli Kılma

Web sitenizin dosyalarının internette erişilebilir olacağı bir sunucuda yer sağlamalısınız. Bu sunucu bir anasistem hizmet sağlayıcı olarak adlandırılabilir. Barındırma için iki seçenek vardır: kendi web sitenizi

barındırabilir veya çevrimiçi bir web barındırma hizmetinden alan satın alabilirsiniz. Bu konu daha detaylı olarak kısım 6.2' de ele alınacaktır.

6.1.3. Web Sitesi Oluşturmak İçin Güvenli Bir Yazılım Edinme

Bir sonraki adım, kendi web sitesi kodunuzu yazmanıza ya da web sitenizi oluşturmak için ihtiyacınız olan kodu otomatik olarak oluşturmanıza izin veren bir yazılım kurmaktır. HTML kodlamasını, bir web geliştirme yazılım paketini veya bir içerik yönetim sistemini kullanabilirsiniz.

Web sayfaları, Notepad gibi basit bir metin editörü kullanarak yazabileceğiniz HTML denilen bir dilde oluşturulabilir. Bu seçenekle, web sitenizi tamamen kendi ihtiyaçlarınıza göre özelleştirebilirsiniz. Ancak, HTML kodunu öğrenmek nispeten fazla zaman, uzmanlık ve para gerektirir.

HTML kodunu yazmak zorunda kalmadan kendi özel tasarımlarınızı oluşturmak için yazılım paketleri satın alabilirsiniz. HTML kodlamaya benzer şekilde, web geliştirme yazılımı bir web sitesinin yürütülmesinde zorluklar yaşatabilir, ancak web sitenizin her ögesini rahatlıkla tasarlamanıza olanak tanır.

Örneğin, Adobe Dreamweaver bir web tasarım ve geliştirme uygulamasıdır. Web sitenizi HTML kodu yazmadan ve kod editörü kullanmadan tasarlayabilmeniz için görsel bir editör sağlar. Böylece web siteniz için Dreamweaver'in ürettiği HTML kodunu değiştirebilirsiniz. Bu yöntem sizin için daha kolaydır çünkü HTML kodunu kendiniz yazmak zorunda kalmazsınız. Ancak uygulama araçlarını öğrenmek zaman alır ve fiyatı nispeten yüksektir. Dreamweaver, hem Mac'lerde hem de PC'lerde çalışır ve daha karmaşık web siteleri için iyidir.

Microsoft Expression Web, web siteleri oluşturmak, tasarlamak ve düzenlemek için başka bir araçtır. Dreamweaver gibi, HTML kodlaması da sizin için web sitesine grafik öğeleri eklerken uygulama tarafından oluşturulur. Ancak, satın alındığında Dreamweaver' dan daha ucuzu gelmektedir. Kendi başınıza (PC ortamında) oluşturduğunuz çok basit web siteleri için, Microsoft Expression Web iyi bir seçimdir.

İçerik yönetim sistemi (CMS), bir web sitesini hızlı ve kolay bir şekilde oluşturmanızı ve korumanızı sağlayan bir yazılım paketidir. Web sitesi içeriğini bir veri tabanında saklar ve içeriği otomatik olarak dışarı çıkarabilir. Ayrıca önceden oluşturduğunuz tasarıma göre uygun sayfalarda gösterebilir. Bu, CMS' nin web sitesinin içeriğini web sitesinin grafiklerinden ayırmamasını sağlar. Tablo 1' de, birkaç içerik yönetim sistemi kısaca açıklanmaktadır.

Tablo 1 Bazı içerik yönetim sistemleri

| İçerik yönetim sistemi | Açıklama | Kullanım alanı |
|---|---|--|
| SquareSpace - SquareSpace ile kolayca multimedya içeren basit bir web sitesi oluşturabilirsiniz. | <ul style="list-style-type: none">Sürükle ve bırak editörü seçeneğiMultimedya kontrolüMobil dostu siteler oluştururŞablon düzenleriStandart plan - 10 \$/ay; Sınırsız plan - 20 \$/ay | Daha görsel ve yönelikli siteler |
| WordPress - WordPress kurulumu ve anlaşılması en kolay sistemlerden biridir. Binlerce ücretsiz önceden tanımlanmış tema mevcut olduğundan, web sitenizi uzmanlık olmadan koruyabilir ve güncelleyebilirsiniz. | <ul style="list-style-type: none">Eklentiler mevcutKarmaşık siteleri çok iyi işleyemez sitelerÖzel içerikler eklemek zor olabilir | Bloglar, küçük işletme siteleri ve kişisel |

- Çekirdek işlevleri özelleştirmek zor olabilir
- Web sitesi oluşturmak ve barındırmak ücretsizdir.
- Önceden belirlenmiş biçimler
- Kod yazmaya gerek yok
- Basit sürükle ve Küçük işletme bırak özelliği siteleri
- Web sitesi oluşturmak ve barındırmak ücretsizdir.
- Tek tıklamayla bir sayfa oluşturun
- Düzeni kendinize göre özelleştirebilir veya bir şablon seçebilirsiniz. Kuruluş içi bilgi paylaşımı
- Web sitesi oluşturmak ve barındırmak ücretsizdir.

Wix - Sürükle ve bırak editörü ile Wix, içerik yönetim sistemi kullanımı açısından oldukça kolaydır.

Google Sites - Google Sites ile çalışanlarınızın ve gönüllülerinizin bilgileri görüntülemesi için bir web sitesi oluşturabilirsiniz.

6.1.4. Web Sitesinin Tasarlanması ve Oluşturulması

6.1.4.1. Web Sitesi Temasının Tasarlanması

Tüm web sitenizin görünümünü ve izlenimini temsil eden temayı seçin veya oluşturun. Bir tema oluşturmak önemlidir, çünkü bu web sitenizi uyumlu hale getirir ve tüm web sayfalarını birbirine bağlar. İçerik yönetim sistemi kullanmanın yararı, seçip özelleştirebileceğiniz temaları önceden belirlemiş olmalıdır.

Görsel çekicilik için resim ve grafikler ekleyin. Ancak fazla ileri gitmeyin çünkü web sitesinde gezinmeyi yavaşlatabilir.

Web sitenize takvimler, anketler, sosyal medya sitelerine bağlantılar, alışveriş sepeti vb. yetenekler eklemek için eklentiler kullanın (bazen bileşen olarak adlandırılır).

6.1.4.2. Web Sitenize İçerik Ekleme

İçerik aslında kişilerin web sitenizi ne için ziyaret ettikleridir. Web tasarımının en önemli yanıdır, bu yüzden hangi bilgiyi içereceğini düşünmek için önemli miktarda zaman harcamanız önemlidir. İçerik yönetimi devam eden bir süreçtir ve web sitenizi sık sık güncellemenizi gerektirir. Olası bir konu listesi oluşturmalı ve bunları ayrı sayfalara dağıtmalısınız.

Site haritası, tüm farklı sayfaların bir listesidir. Kâr amacı gütmeyen bir kuruluşla ilgili herhangi bir web sitesine dahil etmeniz gereken temel web sayfalarını göstermek için aşağıda bir örnek verilmiştir. Bu, web siteniz için yalnızca bir başlangıç noktası olarak kullanılmalıdır; kuruluşunuza özgü sayfalar ekleyebilirsiniz, daha da önemlisi eklemelisiniz.

- Ana Sayfa
- Hakkımızda
- Misyon ve Tarihçe
- Kaynak Toplama / Bağışlar
- Gönüllü fırsatlar
- Personel
- Topluluktaki Çalışmalarımızdan Örnekler
- İstihdam ve Staj Olanakları
- Bizimle iletişime geçin

6.1.4.3. Temel Web Tasarım Konuları

Web sitelerinin etkili ve etkisiz tasarımları hakkında çok fazla kaynak bulunmaktadır. Bunun detaylı tartışıması bu kısmın kapsamı dışındadır. Örneğin “Kâr amacı gütmeyen web sitesi tasarımının en iyi uygulamaları” üzerine yapılan hızlı bir İnternet araştırması, konuya ilgili birkaç makale sağlayacaktır. Buna rağmen birkaç tasarım düşüncesi aşağıda listelenmiştir:

- Kuruluşunuzun misyonu hemen anlaşılmalıdır. Ana sayfanıza odaklanın. Kuruluşunuzun amacını herkese açık hale getirmek ve web sitenizin ana hedeflerini vurgulamak için kenarlıklar, başlıklar ve resimler gibi farklı tasarım öğelerini kullanabilirsiniz. Ayrıca, ana sayfanızın basit olduğundan emin olun. İnsanları çekmek ve kuruluşunuzun web sitesine ilgi çekmek için büyük resimler ve büyük metinler kullanın.
- Web sitenizi kuruluşla ilgili diğer materyalinizle tutarlı hale getirin. Bu organizasyonunuzun ve misyonunun tanınmasını artıracaktır.
- Her sayfaya bir "Şimdi Bağış Yapın!" düğmesini ekleyebilirsiniz. Böylece, ziyaretçiler farklı sayfaları ziyaret ederse, her sayfadan bağış yapabilirler.
- Bir web sayfasına çok fazla metin eklemeyin. İdeal olanı, ziyaretçilerinizin önemli bilgiler için sayfayı aşağı kaydırmasının gerekmemesidir. Metin yoğun bir web sayfasında ziyaretçileriniz ilgisini kaybedecektiler. Sayfalar arama motorları tarafından taranabilir ve özgün olmalıdır.

Etkili bir web sitesi ile kuruluşunuzu daha iyi tanıtabilirsiniz. Bu, daha fazla bağış almanıza, fon toplayıcılara daha fazla katılım sağlamanıza, potansiyel müşterilerinizi bulmanıza ve kuruluşunuzla ilgili bilgileri çalışanlara ve web kullanıcılarına başarıyla iletmenize yardımcı olacaktır.

6.2. Bir Web Sitesinin Fiziksel Olarak Kurulması

Günümüzde çeşitli Web teknolojilerinden yararlanan organizasyonlar için en önemli e-ticaret unsuru kendilerine ait bir Web sitesinin olmasıdır. Öyle ki bazı işletmeler yalnızca kurdukları Web sitesi üzerinden faaliyet göstermektedirler. Hatta bazı bireysel kullanıcılar dahi kendi Web sitelerini kurarak kendilerine iş imkânları yaratmaktadır. Zira artık Web sitesi adeta bir kartvizit haline gelmiştir. Özellikle google.com gibi gelişen arama motorları sayesinde site içinde geçen sözcüklerle bağlı olarak bir müşteriyle irtibata geçilebilmektedir.

Tang ve Huang [2008] Web sitelerinin çeşitli fonksiyonlarının işletmelerin performanslarına doğrudan ya da dolaylı etki ettiğini savunmuşlardır. Söz konusu fonksiyonlar ürün katalogunun sunulması, ürünle ilgili yorumların sunulması, tüketici tercihlerine yönelik özelleştirme imkânının olması ve hesap yönetiminin bulunması gibi tüketici merkezli fonksiyonlardır. Buna göre araştırmacılar; daha fazla fonksiyona sahip olan Web sitelerinin daha çok kullanıcı çektiğini, daha fazla kullanıcı çekken Web sitelerinde müşteri tatmininin daha yüksek olduğunu ve müşteri tatmini daha yüksek olan işletmelerin performansının da arttığını göstermişlerdir. Bununla birlikte *Nantel ve Glaser [2008]* Web sitelerinin yabancı dilde veya çeviri biçiminde değil, doğrudan kullanıcıların ana dilinde tasarlanması, sitenin kullanılabilirliğini artırduğunu öne sürmüştür. Bunun sebebi olarak da çevirinin mükemmel olsa bile kullanıcıyla kültürel bir mesafe yarattığını

belirtmişlerdir. Ayrıca araştırmacılar bunun tam tersi olarak, lisanın daha az önemli olduğu bir teklifin değerlendirilmesi gibi durumlarda Web sitesi lisanının satın alma kararına herhangi bir etkisinin olmadığını görmüşlerdir.

Bir Web sitesinin kurulması aslında öyle çabuk ve kolay bir süreç değildir. Bilindiği gibi bir Web sitesi birden fazla Web sayfasından oluşur ve bunların bir uzman tarafından tasarlanması gerekmektedir. Söz konusu tasarım genellikle HTML ve XML gibi dilleri temel alan çeşitli yazılımlarla yapılmaktadır. Bunlardan en fazla bilineni Adobe Dreamweaver' dir. Bununla beraber bir veritabanı da yüksek ihtimalle gerekeceğinden, veritabanı tasarımcısı da ihtiyaca uygun veritabanını tasarlmalıdır.

Yukarıda anlatılan Web sitesi kurulumu için gereken ihtiyaçlar belli ölçülerde uzman olmayan, fakat yeterli bilgi sahibi olan bir kişi tarafından karşılanabilir. Her şeye rağmen kurulum tamamlanmamıştır. Bilindiği gibi İnternet ve dolayısıyla Web bir ağdır. Öyleyse söz konusu Web sayfaları ve veritabanı ağa bağlı bir bilgisayara yerleştirilmelidir. Hatırlanacağı üzere ağa doğrudan bağlı olan bilgisayarlara anasistem denmektedir. Bu durumda Web sitesi kurulumunun en önemli ihtiyacı bir anasistem hizmetinin sağlanmasıdır.

Anasistem sağlama ihtiyacı iki temel yönteme göre karşılanır. Bunlardan birincisi anasistemin organizasyon tarafından satın alınması ve 'İnternet' e bağlanmasıdır. Anasistem bir İnternet sunucusu olduğundan, sunucu yeteneklerine sahip maliyetli bir bilgisayar satın alınması gerekmektedir. Ayrıca birçok kullanıcı eş zamanlı olarak Web sitesine bağlanabileceğinden, söz konusu anasistemin bant genişliği yüksek olmalıdır. Bu maliyetler maalesef anasistem satın alınması durumunda katlanılacak tüm maliyetleri gösternemektedir. Zira anasistem üzerindeki donanım ve yazılımların yönetilmesiyle bakımının yapılması gerekecektir. Bunun için de uzmanlar istihdam edilmelidir. Tüm bunlara ilaveten güç kesintilerine karşı jeneratör gibi güç kaynakları bulundurulmalıdır. Gerek anasistemin İnternet ile olan bağlantısının kesintisiz olmasını, gerekse de anasistem üzerindeki yükün azaltılmasını sağlamak için bu yöntemi benimseyen organizasyonlar genellikle *yük dengeleme* (load balancing) yoluna giderler. Buna göre anasistemden birden fazla kopya olur. Bu kopyaların her birine *ikiz* (mirror) denir. Böylece trafiği denetleyen bir yazılım hangi ikizde trafik daha düşükse kullanıcıları oraya yönlendirir. Ayrıca ikizlerden herhangi birisi iş göremez hale geldiğinde diğerine bağlantı devam edebilir. Anasistemin tanınabilmesi için bir de alan adı alınması gereklidir. Anasistemin satın alınarak kurulması çok maliyetli olduğundan genellikle az tercih edilen bir yöntemdir.

İkinci yöntem ise bir anasistem hizmet sağlayıcısından yararlanmaktadır. Böylece maliyetler oldukça düşer. Çünkü anasistem satın alınmaz. Bunun yerine sistemi kurmuş olan bir hizmet vericiden hizmet kiralanır. Web anasistem sağlama (Web hosting) hizmetleri kendi arasında dört gruba ayrılır:

- *Paylaşımı Anasistemler* (Shared hosting). Bu yöntemde Web sitesi hizmet sağlayıcının aynı anasistem üzerinde diğer Web siteleri ile birlikte yer alır. Bu nedenle alan adı, alt alan adı biçiminde ifade edilmek zorunda kalınabilir ("isletme.istanbul.edu.tr" alan adında "isletme" alt alan adıdır). Ancak her ne kadar aynı anasistem tek bir IP adresine sahip olsa da bazı yazılımlar, gelen istemleri uygun Web sitesine yönlendirerek müşterinin kendi alan adını belirleyebilmesini sağlamaktadırlar. Daha çok küçük işletmelerin tercih ettiği bir yöntemdir.
- *Sanal Anasistemler* (Virtual private server veya VPS). Bu yöntem aslında paylaşımı anasistemle aynıdır. Tek fark müşteri kendisine ait bir anasisteme sahipmiş izlenimi edinir. Tek bir anasistem, sanal olarak birden fazla kendi alan adı ve adresi olan birer anasistem gibi görünür.
- *Adanmış Anasistemler* (Dedicated hosting). Bazı müşteriler bir anasistemin fiziksel olarak tamamen kendilerine ayrılmamasını isterler. Başka bir ifadeyle anasistemi kimseyle paylaşmamaktadırlar. Böylece hizmet sağlayıcıya ait anasistemlerden birisi müşteriye tahsis edilir. Ancak anasistemin idaresi hizmet sağlayıcıya aittir. Diğer iki yönteme göre biraz daha maliyetlidir.
- *Sunucu Barındırma (Co-location) Hizmeti*. Bazen organizasyonlar tamamen anasistemi yönetmek isterler. Ancak sağlayıcının ağ ve güvenlik yönetimi tecrübelerinden de yararlanmak isterler. Bu durumda firma anasistemi satın alır ve fiziksel olarak hizmet sağlayıcının bulunduğu alana konumlandırır. Diğer üçüyle kıyaslandığında en pahalı yöntemdir.

6.2.1. Ana Sistem Seçim Unsurları

Dünya genelinde organizasyonların çok önemli bir kısmı kendi anasistemi satın almak yerine bir hizmet sağlayıcıdan hizmet almayı tercih etmektedir. Bu hizmet vericiler durumun önemini o derece farkındadır ki

müşterilerinin anasistemleriyle ilgili tüm bilgilerin saklanmasına özen göstermektedir. Öyle ki artık çoğu hizmet sağlayıcı anasistemlerin fiziksel konumlarını bile gizli tutmaktadır. Böylece organizasyonların anasistem hizmetini tercih etmeleri de kolaylaşmaktadır.

Bilindiği gibi hizmet sağlayıcılar dört farklı tipte hizmet sunmaktadır. Bunlardan hangisinin tercih edileceği ise yöneticilerin kararıdır. Yine de Oz [2006] yapılacak olan tercihin bazı unsurlara bağlı olduğunu ileri sürmüştür. Böylece bu unsurların yöneticiler tarafından değerlendirilerek dikkate alınabileceğini savunmuştur. Bu unsurlar aşağıda verilmektedir:

- *Web sitesinde sunulacak uygulamaların tipi ve kalitesi.* Özellikle paylaşılmış anasistem ve sanal anasistem hizmetlerinde müşterinin Web sitesi yönetim yazılımlarına müdahale etmesi ve bunları değiştirebilmesi mümkün değildir. Ayrıca bazı hizmet sağlayıcılar VTYS hizmeti vermeyebilir. Bazıları Linux işletim sistemine sahip olabilir. Böyle bir durumda müşteri Web sitesinde ASP teknolojisini kullanamayacaktır. Çünkü ASP, Microsoft Windows işletim sistemi gerektirmektedir.
- *Depolama alanı.* Web sitesinde çok fazla ürün ve/veya enformasyon sunulabilmesi için anasistem üzerinde fazla yer sağlanması gerekmektedir.
- *Teknik destek kalitesi.* Çoğu hizmet sağlayıcı 7/24 kesintisiz teknik destek sunmaktadır. Teknik desteğin kalitesi ise teknik elemanların bilgi ve becerileri ile kullanılan takım-donanımın kalitesini kapsamaktadır.
- *Trafik kısıtları.* Bazı hizmet sağlayıcılar önceden belirlenmiş bir miktarın üzerinde veri iletimi olması durumunda ilave ücret talep edebilmektedirler. Bazıları da önceden belirlenmiş bir sayının üzerinde, sitenin (dolayısıyla anasistemin) ziyaret edilmesi durumunda ilave ücret talep etmektedir.
- *E-posta hizmetinin varlığı.* Çoğu hizmet sağlayıcı çeşitli e-posta adresleriyle birlikte abonelik sağlamaktadır. Sunulan bu hizmette; e-posta kutularının kapasitesi, otomatik cevaplama ve posta adres listelerine (mailing lists) erişim, değerlemede dikkate alınan unsurlardır.
- *Ölçeklenirlik.* Müşteriler yapısal değişimlerine uygun olarak Web sitelerini güncellemek isterler. Bu durumda hizmet sağlayıcı ölçeklenebilir bir hizmet sunmalıdır. Örneğin büyuen bir firmada daha fazla alan kullanımı ve veri iletim trafiginin gözlenmesi muhtemeldir. Hizmet sağlayıcının müşterinin bu yeni ihtiyaçlarını karşılayabilmesi gereklidir.
- *Sayfa tasarım desteği.* Genellikle müşteriler Web sayfalarını kendi imkânlarıyla tasarlasalar da bazı (genellikle küçük) firmalar hizmet sağlayıcılarından sayfa tasarımında yardım bekleyebilmektedir.
- *Güvenlik.* Hizmet sağlayıcının özellikle İnternet üzerinden gelebilecek her tür saldırıya karşı yeterli güvenliği sağlama gereklidir.
- *Açık kalma oranı.* Herhangi bir sebepten ötürü anasistemin kapalı kalması, üzerindeki Web sitesine (veya sitelerine) erişimin de engellenmesi demektir. Bu nedenle hizmet sağlayıcının garanti ettiği açık kalma oranı önemlidir. Buna göre %99 açıklık oranı veren bir hizmet sağlayıcıda bir anasistem yılda toplam 87 saat 36 dakikaya kadar erişim dışı kalabilir.
- Kurulum ücreti ve aylık ödemeler.

Bölüm Özeti

Bu bölümde e-ticaret için çok önemli bir konu olan web sitelerinin oluşturulmasında planlama ve fiziksel kurulum ele alınmıştır. Planlama kendi içerisinde detaylandırılarak alt başlıklar halinde incelenmiştir. Böylece planlama altında dört konu ele alınmıştır. Daha sonra web sitelerinin en başarılı şekilde müşteriye ulaştırmak için ihtiyaç duyulan fiziksel ortamın oluşturulması ele alınmış.

Kaynakça

Nantel J., Glaser E., “The impact of language and culture on perceived website usability”, J. Eng. Technol. Manage. 25, 112–122, 2008.

Oz E., Management Information Systems, Thomson, Beşinci baskı, 2006.

Tang Q., Huang J., “Impact of Web Site Functions on E-Business Success in Chinese Wholesale and Retail Industries”, Tsinghua Science and Technology 13, 368-373, 2008.

7. ELEKTRONİK TİCARETİN SINIFLANDIRILMASI I

Birlikte Düşünelim

1. İşletmeden işletmeye e-ticaret nedir?
2. E-ticaret ile reklamcılığın nasıl bir ilişkisi vardır?
3. E-ticaret' te mübadeleler ve müzayedeler nasıl gerçekleşir?
4. Çevrimiçi işletme birleşmesi nedir?

Başlamadan Önce

E-ticaret sınıflandırmasının genel başlıklarının öğrenilmesi son derece önemlidir. Zira çok çeşitli e-ticaret türleri vardır. Başarılı bir e-ticaret grişiminini ilk olarak hedeflereini doğru koyması ve daha sonra bunlara odaklanması gereklidir. Bunu başarabilmenin yolu hangi tür e-ticaret'e yüneleceğinin kararını doğru vermekten geçmektedir. Ayrıca işletmeden işletmeye e-ticaret ile ilgili temel kavamlar bu bölümde daha detaylı ele alınacaktır. Zira tutar olarak bakıldığından dünya genelinde e-ticaret'in büyük kısmını işletmeden işletmeye e-ticaret oluşturmaktadır. Bu nedenle yalnızca işletmelerin taraf olduğu e-ticaret türünün detaylarıyla öğrenilmesi bu bölümün başlıca hedefidir.

Giriş

Elektronik ticaret genellikle, ticarete dâhil olan taraflara göre sınıflandırılır. Buna göre ticaret, işletmeler ve tüketiciler arasında gerçekleşen bir faaliyettir. Ancak bazı yazarlar, her ne kadar para ve mal alışverişini biçiminde olmasa da devletin de elektronik ticarette diğerleri ile ilişkiye girdiğini düşünmektedirler [bkz. Gökçen-2005]. Aslında ilgili literatür incelendiğinde devletin e-ticaret iş modellerinden yeterince yararlanmadığı görülmektedir. Her ne kadar bazı araştırmacılar e-devlet için e-ticaret iş modelleri önerse de bunlar henüz çok yenidir. İş modeli kısaca organizasyonun stratejisiyle e-ticaret sistemi arasındaki ilişkinin şekillendirilmesidir [bkz. Janssen ve ark.-2008].

Göründüğü gibi e-ticaret üç tarafın ikili olarak birbiriyle ilişkiye girmesi biçiminde sınıflandırılır. Örneğin işletmeden işletmeye e-ticaret olabileceği gibi işletmeden tüketiciye ve işletmeden devlete de e-ticaret olmaktadır. Böylece üçün ikili kombinasyonu olan dokuz sınıf ticaret söz konusudur:

- İşletmeden işletmeye e-ticaret (Business to business veya B2B).
- İşletmeden tüketiciye e-ticaret (Business to consumer veya B2C).
- Tüketiciden tüketiciye e-ticaret (C2C).
- İşletmeden devlete e-ticaret (Business to government veya B2G).
- Tüketiciden devlete e-ticaret (C2G).
- Devletten devlete e-ticaret (G2G).
- Devletten işletmeye e-ticaret (G2B).
- Devletten tüketiciye e-ticaret (G2C).
- Tüketiciden işletmeye e-ticaret (C2B).

İlk iki madde herkes tarafından kabul edilen, ticaret taraflarıyla ticaretin yönünü kapsamaktadır ve e-ticaretin en önemli türlerini oluşturmaktadır. Bu nedenle ilk iki maddede yer alan B2B ve B2C daha detaylı olarak inceleneciktir. Ancak diğer e-ticaret türleri üzerinden kısaca geçilmesinde yarar vardır.

Tüketiciden tüketiciye e-ticaret, bir tüketicinin başka bir tüketiciye Web aracılığıyla satış yapmasını içerir. Ancak yapılan satışların gerçekleşebilmesi için genellikle bir işletme komisyonkarlığı veya karşılıksız olarak aracı olmaktadır. Örneğin araba satmak isteyen bir kişi; ilanını, satıcı ve alıcıları bir araya getiren

çeşitli işletmelerin Web siteleri üzerinden vermektedir. Dolayısıyla bu tür e-ticaret bazı yazarlar tarafından B2C sınıfına dahil edilmektedir [bkz. Oz-2006].

İşletmeden devlete e-ticarete devletin çeşitli ihaleleri Web sitesinden ilan ettikten sonra ön başvuruları Web aracılığıyla kabul etmesi söz konusudur.

Tüketiciden devlete e-ticaret, vergi ödemeleri, ehliyet ve pasaport başvuruları, sigorta prim ödemeleri gibi işlemlerin Web aracılığıyla yapılmasını kapsamaktadır. Ancak yaygın olarak kullanımı henüz görülmemektedir.

Devletten devlete e-ticaret, bir devlet kurumunun diğer bir devlet kurumuna Web aracılığıyla satış yapmasını kapsar. Devlet Malzeme Ofisi' nin diğer devlet kurumlarına Web sitesinden yaptığı satışlar buna örnek olarak verilebilir.

Devletten işletmeye e-ticaret, yalnızca devletin çeşitli bilgilendirmeleri elektronik olarak yapmasını kapsar. Örneğin ihale ilanı böyle bir e-ticarettir. Unutmamak gereklidir ki bu ilana elektronik olarak yapılan işletme ön başvurusu B2G sınıfında yer alacaktır.

Devletten tüketiciye e-ticaret, devletin sıradan vatandaşlarına sunduğu elektronik hizmetleri kapsar. Bunların çok büyük bir kısmı bilgilendirmeye dayanmaktadır. Örneğin taşıt vergi borcu veya TC kimlik numarasının sorgulanması bu tür e-ticaret kapsamındadır.

Tüketiciden işletmeye e-ticaret, tüketicilerin Web aracılığıyla farklı işletmelerin ürünlerini karşılaştırması sürecidir.

7.1. İşletmeden İşletmeye Elektronik Ticaret

Yalnızca işletmelerin taraf olduğu e-ticaret türüne verilen isim işletmeden işletmeye e-ticarettir. Mal ve hizmetlerin sunulduğu son kullanıcılar buna dâhil edilmez. Her ne kadar tamamı olmasa da dünya genelinde işletmeler arasındaki ticaretin önemli bir kısmı artık e-ticaret biçimindedir. Öyle ki hacim olarak işletmeler arasındaki e-ticaret, işletmeden tüketiciye ticaretin yaklaşık on katını bulmaktadır [Oz-2006].

Firmalar hem imalat hem de ticaret fonksiyonlarını yerine getirir. Sanayi devrimi firmaların üretim fonksiyonlarının mekanikleşme özelliğini görmüş, bilgi devrimi de perakende, toptan ve girdi alımları dahil olmak üzere tüccar işlevlerini otomatikleştirmek resmi tamamlamıştır. Bilgi devrimi, işlem maliyetlerini önemli ölçüde düşürmüştür ve firmaların hem alım hem de satış yapma şeklini değiştirmiştir.

İnternette dikkat çeken ilk işletmeler, Amazon ve CDNow gibi tüketiciye özel (B2C) perakende kurumlarıydı; eBay açık artırma sitesi, tüketiciden tüketiciye (C2C) ticareti de kolaylaştırmıştır. Buna karşın, bu kısmın odağı, mevcut hacmi ve büyümeye potansiyeli perakende işlemlerden çok daha fazla olduğu için, büyük ilgi ve yatırım sermayesi çeken işletmeler arası (B2B) e-ticarettir. B2B e-ticaretin büyülüğu ve büyümeye oranına ilişkin tahminler çok değişkendir. Bu tahminlerin inşası, işletmeler arası işlemlerin büyülüğu ile ilgili gözlemlerle başlar. Örneğin, İnternet'in hızla gelişmeye başladığı 1999'da ABD'deki işletmeler arasındaki işlemlerin yaklaşık 14 trilyon dolara eşit olduğu ve bunun yaklaşık 90 milyar dolarının İnternet üzerinden yapıldığı bildirilmiştir [Uchitelle - 2000].

Şirketlerarası ticaretin İnternet üzerinden gelecekteki büyümesi, şirketler arası işlemlerin genel olarak büyümeye oranının öngörülmesine ve bu tür bir işletmenin İnternet'e geçme oranına ilişkin varsayımlara dayanmaktadır. Günümüzde B2B e-ticaretin yalnızca ABD'de 1 trilyon doların üzerine çıktıığı bilinmektedir [Forrester - 2019].

İşlemlerin maliyetlerini düşürmek için elektronik ticareti kullanabilir. Böylece büyük pazarlardaki B2B e-ticaret şirketleri, alıcılar ve satıcılar arasındaki doğrudan alışverişe kıyasla maliyet verimliliğini sağlayabilir. B2B e-ticaret firmalarının, işletmeler arasındaki işlemlerde aracı olarak hareket ettiğini vurgulamak gereklidir. Bir aracı, satıcılarından alıcılarla satış yapmak için satın alan veya alıcıların ve satıcıların buluşmalarını ve işlemlerini yapmalarına yardımcı olan ekonomik bir figürdür.

Piyasa yapıcılar, talep ve arzı toplar, alıcı ve satıcıları eşleştirir, fiyatlandırma için mekanizmalar sağlar ve ek piyasa likiditesi gereğinde alım ve satıma girerler. Piyasa yapıcılarının finansal piyasalarındaki rolü genel olarak kabul edilmekle birlikte, mal ve hizmet piyasalarındaki şirketler de piyasa yapıcı işlevleri yerine getirirler; alış-veriş, stok tutma ve imalat faaliyetleri yoluyla takas imkânı yaratır, fiyatları düzenler ve mal tahsis eder. Buna göre, B2B şirketleri etkin pazarlarda pazar oluşturucularıdır. Nitekim B2B şirketler, bazı durumlarda finansal piyasalara benzeyen borsalar kurarak mal ve hizmetler için elektronik pazarlar yaratmaya çalışırlar.

İşletmeden işletmeye terimi, toptan ticaretin yanı sıra şirketin hizmet alımları, kaynaklar, teknoloji, üretilmiş parçalar ve bileşenler ile sermaye ekipmanı dahil olmak üzere çok çeşitli şirketler arası işlemleri ifade eder. B2B ayrıca reasürans, ticari kredi ve ticari bonolar, menkul kıymetler ve diğer finansal varlıklar için elektronik ağlar gibi şirketler arasında birçok finansal işlemi de içermektedir. B2B işlemleri, perakende satış, müşteri değişimi ve istihdam gibi konuları içermez.

B2B e-ticaret alanındaki şirketler yenilikçi ekonomik işlemler sunmaktadır. Aracı olarak faaliyet gösteren birçok B2B şirketi, bir şirket ile birçok tedarikçisi arasında yenilikçi işlemler tasarlamayı önerir; yani tedarik zincirlerini yeniden düzenler. Örneğin, bazı B2B şirketleri merkezi çevrimiçi açık artırmalar sunar. Bu tür merkezileştirilmiş pazarlar, alıcı ve satıcıların birbirlerini aradığı ve ikili pazarlığa katıldığı merkezi olmayan pazarları değiştirmeyi amaçlar. Merkezileştirilmiş pazarlar, arama maliyetlerini azaltarak, alıcı ve satıcıların merkezi değişim yoluyla birbirleriyle tanışmalarını sağlayarak verimliliği artırrılar. Çok sayıda alıcı ve satıcı olduğunda, merkezi piyasalar, ikili müzakereleri resmi ihale mekanizmaları ve işlem fiyatları hakkındaki bilgilerle değiştirerek zaman maliyetlerini azatabilir.

7.1.1. Reklamcılık

İşletmeler arası e-ticaretin önemli bileşenlerinden birisi çevrimiçi reklamcılıktır. Bilindiği gibi işletmeler günümüzde her tür iletişim aracıyla reklâm yapmaktadır. Bunların başını da televizyon ve radyo çekmektedir. Her ne kadar televizyon izleyicilerinin sayısına ulaşmamış olsa da İnternet kullanıcıları her geçen gün artmaktadır. Bu durum çoğu işletmenin reklâm yapmak için dikkatini çekmiştir. Böylece Web aracılığıyla işletmelerin reklâmlarını yapmaları söz konusudur. Web aracılığıyla yapılan reklâmlar iki ana başlık altında ele alınabilir:

- Arama reklamcılığı,
- Reklâm bantları (banner).

Arama reklamcılığı esasen çevrimiçi arama amaçlı Web sitelerinde yapılan reklamcılığı kapsar. Bu tür sitelerde ekseriyetle işletmelerin reklamları yapılsa da bazı istisnalarına rastlanılabilir. Arama sitelerinin bazıları (film, arkadaş vb.) özel amaçlı olabildiği gibi çoğunluğu genel amaçlıdır. Başka bir ifadeyle genel amaçlı bir arama sitesi, aranan kelimelerin geçtiği yerleri tüm Web üzerinde tarar. Tarama işlemi sonlandığında arama siteleri bazı sonuçları ilk sıralarda sunar. Bazlarını ise daha dikkat çekici bir biçimde (koyu, renkli vb.) sunar. Tabiatıyla bu tür ayrıcalıkların sağlanması için işletmelerin arama sitesine belli bir ücret ödemeleri gerekmektedir. Böylece, arama sitesi de bir işletme olduğundan B2B e-ticaret söz konusu olur.

Reklam bantları, herhangi bir sitenin üzerine yerleştirilen ve içinde başka bir işletmenin bir tür afişini barındıran görüntülerdir. Özellikle çevrimiçi yayılan dergi ve gazetelerde sıkılıkla bu tür reklamcılığın uygulandığı gözlemlenmektedir. Türkçe *basım sayısı*[1] (impression) olarak ifade edilebilecek olan ve bir Web sitesinin gördüğü ilgiyi belirten bir ölçüt vardır. Web sitelerinin aldığı ücret ve/veya reklâm bandı talebinin sayısı da bu ölçüte göre değişmektedir. Kısaca basım sayısı, herhangi bir tarayıcının reklâm bandını içeren siteyi her yükleyişinde arttırılan bir sayaçtır. Ancak A.B.D. gibi bazı ülkelerde, TV kanalı derecelendirmesine benzer biçimde çeşitli şirketler tarafından daha sahaklı ölçütler sunulmaktadır.

Basım sayısına ilaveten derecelendirme şirketlerinin sunduğu diğer ölçütlerden ilki *ay başına eşsiz ziyaretçi sayısıdır* (unique visitors per month). Bazı reklâm vericiler için farklı ziyaretçilerin o siteyi ziyaret etmiş olması daha önemlidir. Zira reklâm bandını aynı ziyaretçinin defalarca görmesini önemsemeyebilirler. Bu ölçüt bir ay içerisinde Web sitesini ziyaret eden farklı kullanıcı sayısını vermektedir. Bir diğer ölçüt ise *eşsiz ziyaretçi sayfalarıdır* (unique visitor pages). Bu ölçüt tek bir kullanıcının aynı sitede ziyaret ettiği farklı sayfa sayısını gösterir. Eşsiz ziyaretçi sayfaları değeri yüksek olan bir sitede, ziyaretçiler farklı reklâm bantlarını

(farklı sayfalardaki) görebileceklerdir. *Erişim oranı* ise bir önceki ay siteyi ziyaret edenlerin sayısının tüm Web kullanıcılarının sayısına bölünmesiyle elde edilir.

Hemen hatırlatmak gereklidir ki Web günlüklerinde birçok defa bir konuya değinilmesi reklâmin bedava yapılması biçiminde algılanabilir. Başka bir ifadeyle, ürün ya da işletme çeşitli Web günlüklerinde gündeme getirilebilir ve reklâmını ücretsiz yapabilir.

7.1.2. Mübadeleler ve Muzayedeler

İktisatçılar için mükemmel pazar, tek bir satıcının veya tek bir alıcıının olmadığı gerek alıcıların gerekse de satıcıların hızla doğru veriye ulaşabildiği pazardır. Böyle bir pazarın oluşturulabilmesi gerçek yaşamda çok sık karşılaşılabilen bir durum değildir. Ayrıca bazı özel mal veya hizmetler için pazar oluşturulması bile yakın geçmişe kadar bazen mümkün olamamaktaydı. Örneğin kullanılmış “matematik” kitabı almak isteyen bir alıcı böyle bir malın satıcısını bulmak zorundadır. Bunun için pazar araştırmalıdır. Pazarla ulaşsa bile bu uzun bir zaman alacak, ayrıca çeşitli araçların ücretleriyle birlikte malın fiyatı, hak ettiğinin üzerine çıkacaktır. Oysa günümüzde Internet sayesinde yukarıda tanıtılmış olan mükemmel pazar yaratılabilir. Böylece pek çok satıcı, ihtiyaç duyulan tüm enformasyona anında erişerek pek çok alıcıyla buluşabilmektedir.

Mübadele (exchange) bilindiği gibi değişim tokusu faaliyetidir. Kuşkusuz değişim tokusa mal ya da hizmetler konu olur. Ancak unutmamak gereklidir ki mübadelede bir mal ya da hizmeti kesinlikle başka bir mal ya da hizmet ile değiştirmek zorunluluğu yoktur. Başka bir ifadeyle parayla da değiştirilebilir. Buna göre Extranet, B2B e-ticaret ortamı da olabilmektedir. Çünkü belirli mal ve hizmetler için mübadele ortamını sunan işletme, sınırlı sayıda kullanıcıya izin verir. Başka bir ifadeyle Web üzerindeki bu mübadele ortamına girebilmek için genellikle bir üyelik aidatı ödemek gerekmektedir. Böylece söz konusu ortamda alıcı çeşitli ürünlere teklifler verebilir. Bunun dışında bağımsız bir işletme tarafından alıcı ve satıcıları buluşturma amaçlı ortam (Web sitesi) yaratılabilir. Bu durumda gerçekleşen mübadelelerden, ortamı sunan işletme komisyon alır. Komisyonu hem alıcıdan hem de satıcıdan almaktadır.

Müzayedede (auction) ise genellikle herkese açıktır. Bu tür Web sitelerine her türlü alıcı ya da satıcı girebilir. Ancak genellikle müzayedede siteleri banka garantisini veya bir ön ödeme istemektedirler. Alıcı istediği bir ürünü fiyat teklifi verir ve belirli bir süre içerisinde en yüksek fiyatını verene satış gerçekleşir.

Web aracılığıyla sağlanan mübadele ve müzayedede sitelerinde işletmeler, her yıl milyarlarca dolarlık e-ticaret yapmaktadır [bkz. Oz-2006].

7.1.3. Çevrimiçi İşletme Birleşmeleri

Bilindiği gibi birbirinden ayrı işletmelerin birleşmeleri, bunların tek bir raporlama (gelir tablosu ve bilanço gibi) kuruluşu olarak bir araya getirilmesidir. Başka bir ifadeyle bir araya gelen işletmelerin kimlikleri de birleşmektedir. Ancak işletmelerin çevrimiçi birleşimini tam olarak bu tarz bir birleşim olarak görmek doğru değildir. Bunun yerine, bir tür güç birliği yapmak veya birbirine müttefik olmak biçiminde izah etmek daha doğrudur.

Birden fazla işletmenin mal ve hizmetlerini satmak ve çeşitli mal ve hizmetleri satın almak için bir arada oluşturdukları Web siteleri *çevrimiçi işletme birleşmeleri* kapsamında yer alır. Söz konusu işletmelerin birbirine rakip olduklarını unutmamak gereklidir. Birbirine rakip olan işletmelerin böyle bir ortaklığa gitmesinin birinci nedeni, satın almalarında tek alıcı durumuna gelebilmektir. Başka bir ifadeyle birbirine rakip olan ve çoğu ihtiyacı birbiriyle aynı olan işletmeler, bu ihtiyaçlarını ortak bir Web sitesi üzerinden temin edebilirler.

İşletmeleri çevrimiçi ortaklığa iten diğer neden ise bazı durumlarda farklı satıcıların, tüketici için tek bir adreste bulunmalarının birleşmeye bütün olarak katkı sağlayabilmesidir. Özellikle bu duruma güzel bir örnek emlak sektöründen verilebilir. Genellikle emlak komisyoncularının elinde müşteri için yeterli alternatif olmadığı durumlar gözlenebilir. Bu durumda söz konusu emlak komisyoncusu çevrimiçi birleşmeden yararlanarak başka bir komisyoncuya bulunan gayrimenkulü satabilir veya kiralayabilir. Tabiatıyla komisyoncular elde edilen komisyon gelirini aralarında paylaşacaklardır.

Bununla beraber tüketicilerin böyle tek bir adresin olduğuna ikna olmaları, o adresten başka yerde arama yapmamalarını ve alacakları ürünü daha çabuk kabul etmelerini sağlar. Özellikle pahalı ürünler söz konusuysa ve tüketiciler alternatif bir satıcının daha olduğunu bildikleri sürece, genellikle o satıcıyı da incelemeden satın alma kararını vermezler. Böylece, bulundukları adreste tüm satıcıların mevcut olduğuna inanırlarsa tabiatıyla daha çabuk satın alma kararını vereceklerdir.

Yukarıda verilmiş olan örnekler ilave olarak, havayolu işletmelerinin, otellerin, yiyecek işletmelerinin ve perakendecilik sektöründeki işletmelerin de çevrimiçi birleşmeler yaptıkları görülmektedir. Çevrimiçi birleşmeler özellikle dünya genelindeki havayolu taşımacılığı için daha büyük önem taşıyabilmektedir. Zira yolcular bir noktadan diğerine çeşitli sebeplerden ötürü aktarımı gitmek durumunda kalabilmektedir. Bu aktarımların hepsi aynı havayolu işletmesi tarafından da yapılamaz. Bu durumda tüketici gideceği yere kadar tüm bağlantıları kendi tasarlamak istemeyecektir. Örneğin İstanbul' dan Atlanta' ya gitmek isteyen bir yolcu için en uygun seçenek İstanbul ile Frankfurt arasını THY, Frankfurt ile New York arasını Lufthansa ve New York ile Atlanta arasını Delta Air Lines ile seyahat etmek olabilir. Bu verilerin hepsini ayrı ayrı THY, Lufthansa ve Delta Air Lines Web sitelerinden elde etmek tüketici için çok zahmetli bir süreç olacaktır.

Bölüm Özeti

Bu bölümde e-ticaret'in sınıflandırılması ele alınmıştır. Tüm e-ticaret türleri incelenmiş ve bunlardan özellikle işletmeden işletmeye e-ticaretin önemi vurgulanmıştır. İşletmeden işletmeye e-ticaret kendi içerisinde detaylandırılarak alt başlıklar halinde incelenmiştir. Böylece işletmeden işletmeye e-ticaret altında üç konu ele alınmıştır.

Kaynakça

Forrester Research, Inc., <https://www.forrester.com>, 2019

Gökçen H., Yönetim Bilgi Sistemleri, EPİ Yayıncıları, 2005.

Janssen M., Kuk G., Wagenaar R. W., "A survey of Web-based business models for e-government in the Netherlands", *Government Information Quarterly* 25, 202–220, 2008.

Oz E., Management Information Systems, Thomson, Beşinci baskı, 2006.

Uchitelle, Louis, 2000, "It's Just the Beginning," New York Times, Special Section on E-Commerce, June 7, p. E1.

8. ELEKTRONİK TİCARETİN SINIFLANDIRILMASI II

Birlikte Düşünelim

1. İşletmeden tüketiciye e-ticaret nedir?
2. Elektronik veri değişimi nedir?

Başlamadan Önce

Bu bölümde genel olarak işletmeden tüketiciye e-ticaret ve elektronik veri değişimiyle ilgili ön bilgi verilecektir. Zira günümüzde işletmeden tüketiciye e-ticaret işlem sayısı bakımından ilk sıradadır. Fakat tutar olarak ilk sıradan olmasa da işletmeden işletmeye e-ticaretin hemen arkasında yer almaktadır. Özellikle son yıllarda eğilim incelendiğinde önumüzdeki yıllarda daha da önem kazanacağı ortadadır. Bu nedenle işletmeden tüketiciye e-ticaret'in detaylarıyla incelenmesi önemli görülmüş ve bu bölümde yer verilmiştir.

İşletmelerin hataları azaltmak, işlemleri hızlandırmak ve stok kontrolünü doğru yapabilmek için son zamanlarda çok önem verdikleri elektronik veri değişimi üzerine kapsamlı bir inceleme yapılması bu bölümün kazanımları arasındadır.

Giriş

Her ne kadar e-ticaretin önemli bir kısmını B2B e-ticaret, B2C e-ticaret ile kıyaslanamayacak kadar fazla bir biçimde kaplaza da B2C e-ticaret te hızla artmaktadır. Öyle ki 2007' de A.B.D. için B2C toplam e-ticaret gelirleri 175,000,000,000\$' ı bulmuştur. *Mulpuru ve ark.* [2008] ise A.B.D.' inde 2012' ye kadar söz konusu gelirlerin 335,000,000,000\$' ı bulacağını tahmin etmişlerdir.

B2C Ticaret bir işletme ile tüketici arasındaki mal ve hizmet alım satımına ilişkin etkileşimler perakende işlemlerdir.

“Yenilik”, perakende işleminin geleneksel olarak mağazada adeta “tuğla ve harç” işi biçiminde yapılmasından ziyade İnternet üzerinden yapılmasıdır.

B2C'nin bu geleneksel modelden teknik evrimi yakın zamanda ortaya çıkmamıştır.

Birkaç yüz yıldır ticaret yazılı belgelerin hareketine dayanıyordu. Bu belgeler, bir şirketin iş yapmak için başka bir şirkete iletmesi gereken bilgileri içeriyordu. Belli bir süre zarfında, Fatura, Kredi Notu ve Sipariş gibi standart adlar almaya başladı. Ancak, belgeler kesinlikle standart bir düzende değildi. Olmaları gerekmeyordu çünkü alıcı her zaman bir insandı ve insanlar okuma, yorumlama ve rasyonelleştirme yeteneğine sahipti. Örneğin, bir fatura belgesinde söylelenebilecek her şey hakkında, söz konusu taraflarla ilgili başlık bilgileri, ürünler, miktarlar ve fiyatlar hakkındaki detay satırları ve son olarak bazı toplam bilgiler içereceği şeklindedir.

1950'lerin başında, bilgisayarlar büyük şirketler tarafından muhasebe ve bordro ihtiyaçları için kullanılmaya başlandı. Takip eden on yıllar boyunca, bilgisayarlar yalnızca muhasebe ile değil, üretim, yönetim ve diğer tüm ticaret alanlarına dahil olana kadar işlem sonrası ilave görev üstlendiler. Ancak bir şey değişmedi. Bilgisayarlar hala standart olmayan formatlarda basılı belgeler üretiyorlardı.

Bu durum, belge gönderenler için çok kötü değildi ancak alıcı için çok daha kötüydi. Bir şirketin bilgisayarından ticaret ortağının bilgisayarına birçok belge gönderilmesi gerekmektedir. Bilgisayarlar yazılı belgeleri kolayca okuyamaz ve okuduklarında neyin ne olduğunu anlamalarını sağlamak neredeyse imkânsız bir görevdir, bu nedenle alıcı şirketin alınan belgelerdeki bilgileri şirketin bilgisayar sistemine yeniden girmek için personel çalıştırması gereklidir. İşte bu sorunların çözülmerek başarılı bir e-ticaretin yürütülebilmesi için özellikle B2B e-ticarette Elektronik Veri Değişimi (EDI) standartlarına ihtiyaç duyulmuştur.

8.1. İşletmeden Tüketicisiye Elektronik Ticaret

8.1.1. İnternet Perakendeciliği (E-Retailing)

İnternet perakendeciliği sayesinde artık tüketiciler sanal olarak mağazaları gezebilmektedir. Söz konusu mağazalar o kadar çeşitlenmiştir ki yiyecekten beyaz eşyaya kadar örnekler verilebilir. Böylece tüketiciler, bilgisayarları başında mağazayı gezerken aynı zamanda çok fazla enformasyon toplayabilmekte ve farklı mağazalarla fiyat ve kalite kıtaslarına göre kıyaslamalar yapabilmektedirler. Ayrıca tüm bunları kısa bir zamanda yapabilmektedirler. Tüketicinin almaya karar verdiği ürünün ücreti, kredi kartıyla veya PayPal gibi aracı hizmetlerle ödendikten sonra adresine ulaştırılır. Anlatılan bu sebepler nedeniyle de İnternet perakendeciliği dünya genelinde her geçen gün daha fazla tüketiciyi çekmektedir. Çekiciliğinin artmasındaki en önemli sebeplerden bir diğeri de kuşkusuz tüketicilerde artık İnternet perakendeciliğine karşı güven oluşmasıdır. İlk yıllarda kredi kartı bilgilerinin kötü amaçlı kullanılabileceği korkusu tüketicileri uzak tutmaktadır. Günümüzde İnternet perakendeciliğinin oldukça fazla rağbet görmesinin nedenleri aşağıdaki gibi özetlenebilir:

- Her an, herhangi bir mağazadan alışveriş yapabilme rahatlığı.
- Sayısız çevrimiçi mağazayı birkaç dakikada gezerek zaman kazanma.

- Çeşitli arama siteleri vasıtasıyla, belirli bir ürünü kimin sattığını saniyeler içerisinde bulabilme imkânı.
- Birçok satıcı arasında fiyat ve kalite karşılaştırmasını hızlıca yapabilme imkânı.
- Ürünle ilgili diğer tüketicilerin ve bağımsız uzmanların yazmış olduğu yorumları okuma imkânı.

Bununla birlikte satıcı açısından bakıldığında İnternet perakendeciliğinde mağaza gezicisinin alıcı yapılması, alıcının da sürekli alıcı yapılması amaçlanır. Bunun için cerezler yardımıyla tüketicilerin davranışları takip edilmekte hatta bir ürünü ne kadar incelediği dahi saklanmaktadır. Bazı hukukçular bunun hukuka aykırı bir davranış olduğunu savunmaktadır. Zira genellikle tüketiciler bu tür kayıtların tutulduğundan haberdar değildirler. Daha ötesi bazı İnternet perakendeciliği işletmeleri, elde ettikleri tüketici bilgilerini başka işletmelere de para karşılığı satmaktadır. Bu biçimde elde ettikleri verilere göre İnternet perakendeciliği işletmeleri, tüketcilere e-posta yoluyla ilgili olabilecekleri ürünlerde yönelik kampanya, promosyon ve iskonto gibi haberleri iletir. Ayrıca tüketici aynı çevrimiçi mağazaya girdiğinde, ilgi duyduğu ürünlerde yönelik sayfalar daha çabuk ve açılış sayfası biçiminde karşısına çıkabilmektedir.

Satıcı açısından İnternet perakendeciliğinin tek yararı tüketici bilgilerinin kayıt altına alınması değildir. Genellikle bu tür işletmelerin maliyetleri de azalmaktadır. Zira bir çevrimiçi mağazanın, merkezi bir yerde gerçek bir mağaza kiralamasına gerek yoktur. Şehir dışında malları koymak için bir depo yeterlidir. Hatta bazıları buna dahi gerek duymazlar. Satış bilgilerini doğrudan üreticiye yönlendirip, üreticinin malı alıcıya göndermesini sağlarlar. Ayrıca çevrimiçi mağazalarda satışçılara ve kasiyerlere de gerek yoktur. İnternet perakendeciliği yapan işletmelerin maliyetleri, paketleyici elemanların ücretleri ile paketleme ve gönderme giderleridir.

İnternet perakendeciliğinde *iştiraklerden* de sıkılıkla yararlanılır. İştirakler diğer e-ticaret türlerinde de olduğu gibi reklâm bantlarının bazı Web sayfalarına konulmasıdır. Web sayfası sahipleri de iştirak olarak tanımlanır. İştirakler üç kısṭasına göre ücretlendirilir:

- Bant aracılığıyla gelen ziyaretçi alımlığında gerçekleşen ve en fazla tercih edilen *satış başına ödeme*,
- Reklâm bandının her tıklandığında gerçekleşen *tıklama başına ödeme*,
- Bant aracılığıyla gelen ziyaretçinin alım yapmamasına rağmen, perakendeci işletmenin üyelik formunu doldurmasıyla gerçekleşen *üyelik başına ödeme*.

Ayrıca bazı İnternet perakendecilerinin *rakiplerine* de komisyon karşılığında hizmet verdiği görülmektedir. Örneğin Amazon.com, Web sitesinde yapılan bir aramanın sonuçları arasında rakiplerinin ürünlerini de sunmaktadır. Böylece rakibinin bir ürününü sattığında komisyon almaktadır. Bunun avantajlarından birisi tüketicilerin daha fazla ürün bulabildikleri için ilk ziyaret ettikleri yerin olabilmesidir. Diğer bir avantaj ise rakiplerinin satışlarından da kazanç elde edebilmesidir.

8.1.2. Müzayedeler ve Ters Müzayedeler

İşletmeler arası e-ticarette bahsedilen müzayedede, Web aracılığıyla bireyler arasında da gerçekleşmektedir. Ancak söz konusu müzayededen yapılabilmesi için mal veya hizmet sahibi bireylerin bunları sunabilecekleri bir Web sitesi gerekmektedir. Genellikle bu amaçla tasarlanmış Web sitelerinden birisinde bir kişi mal veya hizmetini diğer bireylere sunar. Ardından belirli bir süre boyunca teklifler verilir. Satış gerçekleştiğinde Web sitesi sahibine belli bir ödeme yapılır. Görüldüğü gibi bu C2C e-ticarete benzemektedir ve bazı yazarlar da öyle isimlendirmiştirlerdir. Oysa aradaki Web sitesinin başka bir ifadeyle işletmenin, ürünün satılmasında çok önemli etkileri söz konusudur. Örneğin diğer tüketicilerin ürünü yönelik yorumları, benzer ürünler ve bazı ilave enformasyon sunmaktadır. Ayrıca bir mal veya hizmet satıldığında ödeme genellikle Web sitesi sahibine yapılır. Bu anlamda Web sitesi sahibi gerçek satıcı gibi, mal veya hizmeti satan birey ise bir üretici gibi düşünülebilir. Böylece bunun gerçekte B2C e-ticaret olduğu görülmektedir.

Ters müzayedede ise tüketici talep ettiği hizmet için düşündüğü ücreti böyle bir amaca yönelik olarak tasarlanmış Web sitesinde sunar. Ardından Web sitesi söz konusu hizmeti verenleri davet eder. İlk kabul eden ürünü satar. Bazen hiçbir üretici teklifi kabul etmeyebilir. Mal satışlarında fazla görülmeyen bir yöntemdir. Genellikle seyahat, konaklama ve araba kiralama gibi hizmetlerde gözlenir.

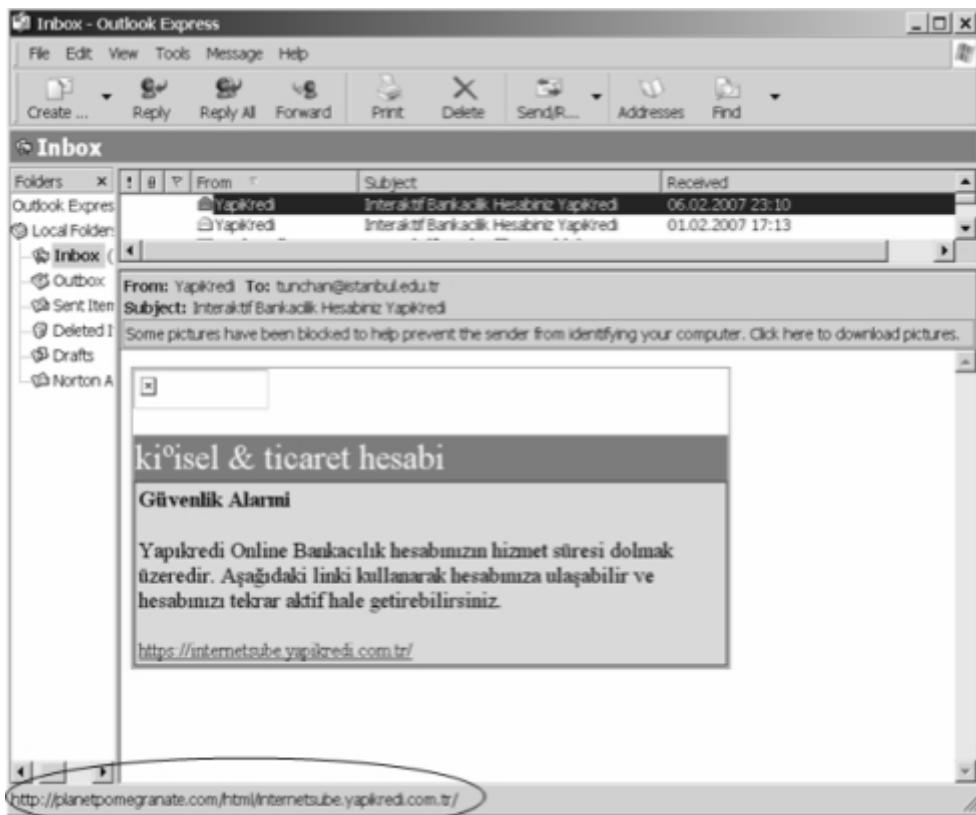
8.1.3. İçerik Sağlayıcılar

Web için “icerik” denildiğinde; haber, araştırma sonuçları, çeşitli istatistikler gibi enformasyonlar ile müzik, resim ve görüntü parçaları (video klip) gibi sanatsal çalışmalar akla gelir. Bazı yazarlar bu gruba seri ilanları dahil etmektedir. Her ne kadar haberler bedavaya temin edilebilse de bazı makaleler edilememektedir. Bu nedenle bazı İnternet kullanıcıları, özellikle uzmanlık alanlarındaki bu tür içerikler için ücret ödemeye razıdır. Hatta çoğu kişi içeriklerin çevrimiçi sağlanması tercih etmektedir. Çünkü aradıkları belirli bir husustaki içeriğe elektronik ortamda hızla erişebilmektedirler [bkz. Oz-2006].

8.1.4. Fatura Gönderme ve Ödeme

İnternet’ in gelişimi ve yaygınlaşmasıyla birlikte artık çevrimiçi olarak paranın bir banka hesabından başka bir banka hesabına aktarılması ve fatura gibi enformasyonların e-posta yoluyla aktarılması son derece kolaylaşmıştır. Bu nedenle özellikle belirli dönemlerde düzenli olarak gerçekleşen ödemelerin faturalarını, tüketiciler e-posta yoluyla almaya genellikle rıza gösterirler. Söz konusu düzenli ödemeler; telefon faturasının, kablolu TV faturasının ve çeşitli üyelik aidatlarının ödenmesi biçiminde görülebilir. Bu türden hizmet veren şirketler masrafları düşürmek için tüketicilerin *elektronik fatura* (e-fatura) yöntemini kabul etmesi için onları teşvik dahi ederler. Örneğin uz iletişim şirketleri bu yöntemi kabul eden tüketicilerinin telefon görüşme ücretlerinde indirimler yapabilmektedir.

Tüketiciler e-faturaya kolaylıkla rıza göstergelerine rağmen, banka hesaplarından otomatik olarak fatura bedellerinin çekilmesine aynı kolaylıkta rıza göstermemektedirler. Hatta bankalar da bazı ödemelerin otomatik gerçekleşmesini istemeyebilmekte ve bunlara bazı sınırlamalar getirebilmektedirler. Daha ötesi tüketicilerin bazıları herhangi bir elektronik ödeme yapmaktan bile çekinmektedir. Bunların en önemli sebebi kuşkusuz İnternet aracılığıyla kişilerin banka hesap bilgilerinin çalınma vakalarının yaşanmış olmasıdır. Kısaca *şifre çalma* (phishing) olarak adlandırılan bu kanun dışı girişimlerde genellikle tüketicilerin hesap bilgilerini, bankanıkiyle aynı tasaranmış, ancak farklı bir Web sitesine girmesi sağlanır. Böylece tüketicinin hesap numarası ve şifreleri kaydedilir. Şekil 7’ de bu amaçla tüketiciye gönderilmiş bir e-posta görülmektedir. E-posta alıcısının söz konusu bankada hesabı dahi olmayabilir. Görüldüğü gibi tüketicinin yönlendirildiği bir bağlantı (altı çizili) vardır. Bu bağlantı HTTPS standartlarına uygun, güvenli bir Web sitesine yönlendirmiştir gibi görünse de aslında yönlendirilen Web sitesi yuvarlak içine alınmış olan HTTP standartlarındadır.



Şekil 7 Bir şifre çalma girişim örneği

8.1.5. Mobil Ticaret

Daha önce kablosuz ağların önemine deðinilmiþti. Hatırlanacaðı gibi bu tür ağlar sayesinde kullanıcılar ofis ya da evlerinin dışından da ağlara, daha da önemlisi Web' e bağlanabilmektedirler. Böylece kullanıcılarla büyük kolaylıklar sunulabilmektedir. Kablosuz BT' leri mobil ticareti (m-ticaret) sağlamaktadır. Söz konusu kablosuz aðın var olduğu ve tabiatıyla elde tutulur bir bilgisayarın cep telefonu özelliklerine sahip olduğu düşünülsün. Böylece herhangi bir satıcının maðazasına 20 – 30 metre kadar yaklaşan tüketicinin bilgisayarına, kablosuz aðı vasıtasiyla promosyonlar ve ilanlar gelebilecektir.

Mobil ticaret 1997 yılında, ilk kez iki cep telefonu firmasının Coca Cola otomat makinelerinin Finlandiya'daki Helsinki bölgesinde kurulmasını sağlamasıyla doğdu. Makineler SMS metin mesajı ile ödeme kabul ediyordu. İlk cep telefonu tabanlı bankacılık hizmeti 1997 yılında Finlandiya Merita Bankası tarafından SMS kullanılarak da başlatıldı. 1998 yılında ilk dijital indirilebilir içeriðin cep telefonlarına indirilmesi olarak, Finlandiya'daki ilk ticari indirilebilir zil sesleri Radiolinja (şimdi Elisa Oyj'in bir parçası) tarafından piyasaya sürüldüğünde mümkün oldu.

1999'da mobil ticaret için iki büyük ulusal ticari platform başlatıldı: Filipinler'de Akıllı Para ve Japonya'da NTT DoCoMo'nun i-Mode Internet hizmeti. I-Mode, NTT DoCoMo'nun kullanıcıların içerik için ödediği ücretin yüzde 9'unu tuttuðu ve içerik sahibine yüzde 91'ini iade ettiði devrim niteliðinde bir gelir paylaşım planı sunmaktadır. Mobil ticaretle ilgili hizmetler 2000'lerin başında hızla yayıldı. Norveç mobil otopark ödemeleri başlattı. Avusturya mobil cihaz üzerinden tren biletini sundu. Japonya, mobil uçak biletini satmaya başladı.

Şu an için dünya geneli açısından bakıldığında henüz geleceðin BT gibi görünse de somut olarak bazı ülkelerde m-ticaret örnekleri yaşanmaktadır. Örneðin Japonya' da tüketicilerin cep telefonları aracılığıyla verdikleri çeşitli hızlı servis (fast food) siparişleri telefon faturalarına yansıtılmaktadır. Bir başka örnek A.B.D. Florida' dan verilebilir: sürücüler otomobillerini park ettiklerinde park makinesinin numarasını telefonlarına girmektedirler. Böylece park süresi dolduðunda makine telefona mesaj göndermektedir, sürücü de Web aracılığıyla ödemeyi gerçekleştirmektedir. Görüldüğü gibi park makinesine acele etmek zorunda kalmamaktadır. Ayrıca m-ticaret alanında ilerlemiş diğer bir ülkenin de Güney Kore olduğunu söylemek gerekmektedir [Oz-2006].

8.2. Elektronik Veri Deðiþimi

Özellikle tedarik zinciri açısından bakıldığında işletmelerin birbiriyle güvenilir ve hızlı bir yoldan haberleşmeleri, başka bir ifadeyle veri deðiþimi yapmaları gerekmektedir. İnternet' in ilk dönemlerinden itibaren işletmeler birbirlerine çeşitli dokümanları elektronik olarak aktarmaya başlamışlardır. Böylece *elektronik veri deðiþiminin* (electonic data interchange veya EDI) temelleri atılmıştır.

Günümüzde işletmelerin özellikle tedarikçileri, müşterileri ve iş ortakları gibi pek çok işletmeyle elektronik olarak veri deðiþimi söz konusu olabilmektedir. Başka bir ifadeyle EDI' nin amacı elektronik olarak sipariş alınması, ticari sözleşme ve faturaların hazırlanmasıyla; gümrük, bankacılık ve benzer işlemlerin yapılmasında tekrarların önlenmesi ve bu işlemlerin en az hatayla, en kısa zamanda tamamlanmasıdır [Gökçen-2005]. Böylece EDI, aktarılacak verinin biçimini için belirlenmiş standartları, veriyi tercüme edecek yazılımı ve verinin üzerinde akacağı aðı kapsamaktadır.

Bilindiði gibi aktarılan veri bir dokümandır. Böyle bir durumda bu dokümanın bir standarda göre biçimlendirilmesi gereklidir. Bu biçim sipariþ, ödeme ve fatura gibi verileri birbirinden ayırt edip anlamlı hale gelmesini sağlayacaktır. Söz konusu standartlara uyan veri bir yazılım tarafından kolayca okunarak hemen işleme aktarılabilir. Örneðin bir sipariþ emrini okuyan yazılım, veritabanındaki siparişlere bunu hemen ekleyebilir. Bu tür standartlar ve yazılım olmasaydı, o zaman çalışanlar aktarılan her veriyi tek tek incelemek durumunda kalırlardı.

Elektronik veri deðiþimi genellikle tüm İnternet' e açık değildir. Başka bir ifadeyle, EDI için Extranet tercih edilmektedir. Ayrıca söz konusu Extranet üzerindeki EDI traðığını denetleyen ve düzenleyen bir uz iletişim işletmesi de üyelerine hizmet verir. Bunlar yaptıkları iş karşılıðında üyelerinden ücret alırlar.

Elektronik veri değişiminde yüksek güvenlik ve reddedilmeme (non-repudiation)[2] gibi önemli avantajlar sağlansa da EDI' nin diğer bir türü XML standartlarından yararlanarak, Web sayfaları aracılığıyla gerçekleştirilebilir. Bilindiği gibi XML veri paylaşımını kolaylaştırmaktadır. Ancak hemen belirtmek gerekir ki gizli verilerin bu yöntemle paylaşılması sakıncalı olacaktır. Veri değişimi için Web sayfaları kullanılmayabilir. Böylece güvenli yollarla XML biçimindeki verinin değişimi söz konusu olmaktadır. Zira EDI artık veri değişiminde kullanılan ASC X12 ve EDIFACT gibi biçimleri temsil etmek için kullanılmaktadır. Buna göre klasik EDI biçimini sektörler arası veri değişiminde hala güçlü bir yere sahipken, sektörel veri değişiminde XML biçiminde veri değişimi ön plana çıkmaktadır [Nurmilaakso - 2008].

8.2.1. Mesaj Standartları

Telefon benzetimiyle, telefon arayanlarımız bilgi alışverişinde bulunabilir, çünkü her iki taraf da aynı dili konuşur ve kullanmaları gereken protokoller anlar, ancak ne yazık ki uzaktaki taraf arayanın verdiği bilgiyi anlamamaktadır, çünkü farklı sektörlerdir. Ayrıca her sektörün kendi jargonu ve bilgi dizisi veya düzeni vardır. Bu sorunun üstesinden gelmek için, ticarete özgü ve genel belge standartlarının kontrol eden kuruluşlar tarafından tanımlanan EDI modelinin son katmanı geliştirilmiştir.

Mesaj standartlarının başlangıçta farklı yorumlara tabi olduğu ve hatta bazı ticaret ortakları tarafından kötüye kullanıldığı belirtilmelidir. Bu nedenle ortaklar arasında yalnızca kullanılan mesaj standartları ile ilgili değil, aynı zamanda bilgilerin tam anlamı ve içeriği hakkında da anlaşmalar yapılması hayatı önem taşır. EDI değişimleri gerçekleşmeden önce birçok büyük kuruluş, kullanmak istedikleri standartlar için kendi uygulama kılavuzlarını yayınlar. Bu standartlar endüstri standarı bir mesaja dayansalar da düzen ve formatta küçük farklılıklar olabilir.

EDIFACT mesaj standardı Avrupa'da en yaygın kullanılanıdır. EDIFACT, belirli bir sektörle bağlantılı olmayan genel kullanım standartıdır ve hem EAN numaralandırma birlükleri hem de Birleşmiş Milletler ile yakından ilişkili olan EDIFACT kurulu tarafından kontrol edilir. Otomotiv endüstrisi için ODETTE (Avrupa'da Telekomünikasyon ile Veri Değişimi Organizasyonu) ve VDA (Verband der Automobilindustrie) bulunmaktadır. ODETTE tarihsel olarak kendi standartlarına sahipti, ancak 2000'den beri EDIFACT standartlarını benimsedi. VDA standarı gerçek bir EDI standarı değildir ancak birçok benzerliği vardır. Kimya endüstrisi için CEFIC ve perakende endüstrisi için TRADACOM gibi başka birçok endüstriye özel standartizasyon mevcuttur.

Her mesaj standarı ilk önce bölümleri ve bunların nerede olduğunu ayırt eden bir yapı şeması ile tanımlanır. Yapı şemasındaki her bölüm, birkaç özelliğe sahip olarak tanımlanır. Bunlar bir isim, azami oluşum sayısı ve zorunlu veya şartlı durumdur. Mesaj segmenti Şekil 8' de gösterildiği gibi bir şemada çizilmiştir.



Şekil 8 Bir EDI mesaj segmenti

Burada, segment adının diyagramın üst bölümünde tutulduğu bir NAD segmenti (NAD segmentleri EDIFACT tarafından adları ve adresleri tanımlamak için kullanılır) görüyoruz. M segmenti zorunludur, mesajda en az bir kere gerçekleşmesi gerektiği anlamına gelir. Zorunlu olanın alternatifisi, içinde tutulacak herhangi bir bilgi yoksa oluşması gerekmeyen koşullu (C) segmenttir. R segmenti yinelenir, bu bir defadan fazla olabileceği anlamına gelir. Örneğin, bir faturada birden fazla kalemler satırı olabilir. Buna alternatif, yalnızca bir kez gerçekleşebilecek bir bölüm için 1 sayısı veya bölümün olabileceği maksimum süreyi tanımlayan sayıdır.

Bölüm Özeti

Bu bölümde e-ticaret'in sınıflandırılması konusuna devam edilmiştir. Ağırlıklı olarak işletmeden tüketiciye e-ticaretin önemi vurgulanmıştır. İşletmeden tüketiciye e-ticaret kendi içerisinde detaylandırılarak alt başlıklar halinde incelenmiştir. Böylece işletmeden tüketiciye e-ticaret altında beş konu ele alınmıştır. Ayrıca işletmeden işletmeye başta olmak üzere tüm e-ticaret türleri için önemli bir kavram olan elektronik veri değişimi konusu ele alınmıştır.

Kaynakça

Gökçen H., Yönetim Bilgi Sistemleri, EPİ Yayıncıları, 2005.

Mulpuru S., Johnson C., McGowan B., Wright S., "US eCommerce Forecast: 2008 To 2012", *Forrester Research Inc.*, <http://www.forrester.com/FactSheet>, 18 Ocak 2008.

Nurmilaakso J. M., "EDI, XML and e-business frameworks: A survey", *Computers in Industry* 59, 370–379, 2008.

Oz E., Management Information Systems, Thomson, Beşinci baskı, 2006.

9. VERİ TABANI YÖNETİM SİSTEMLERİ I

Birlikte Düşünelim

1. Varlık, özellik ve anahtar ne demektir?
2. Veritabanı yaklaşımı ile neler sağlanmıştır?
3. En çok bilinen veritabanı modelleri nelerdir?

Başlamadan Önce

Veritabanı Yönetim Sistemleri günümüzün en önemli bilişim teknolojisi konularından birisidir. Zira, son zamanlarda güncel bir konu olan veri analizi, veri madenciliği, müşteri ilişkileri yönetimi ve kurumsal kaynak planlaması gibi konuların başarılı yapılabilmesi için ön koşul başarılı bir veri tabanına sahip olunmasıdır. Bu nedenle bu kısımda veritabanı yönetim sistemleri hakkında genel bilgi verilecektir. Bununla birlikte özellikle ilişkisel model olmak üzere tüm veritabanı modelleri inclenecektir. Klasik dosyalama yaklaşımı ilk zamanlar uygulanan veri saklama ve yönetme biçimiydi. Fakat son derece zahmetli ve sorunluydu. Dolayısıyla, bu yöntemden sonra veritabanı yaklaşımının sağladığı önemli avantajlar bu bölümde tartışılacaktır.

Giriş

Günümüzde artık işletmeler açısından en önemli konulardan birisi çok fazla veriyi saklamak, bunlara hızla erişmek ve söz konusu verilerden enformasyon elde etmektir. Normalde bir verinin saklanması klasik dosyalama mantığına göre gerçekleşir. Buna göre dosyalar bir dizi byte olarak görülmektedir. Bu durum ileride de görüleceği gibi bazı zorluklara neden olmaktadır. Bu zorlukları ortadan kaldırmak için çeşitli veritabanı yönetim sistemlerinden (VTYS) yararlanılır. Bunlar esasen birer yazılımdır ve görevi veriyi idare etme işini son derece kolaylaştırmaktır. VTYS'leri dördüncü nesil programlama dilleri olarak kabul etmek mümkündür. Kısaca veri yönetimi, verinin saklanması, çağrıması ve üzerinde birtakım işlemlerin yapılmasını içerir [Gökçen–2005]. Laudon ve Laudon [2006] ise VTYS için veriyi bir merkezde toplayan, onu etkin bir biçimde yöneten ve saklanan verilere uygulama programlarının erişmesine izin veren yazılım tanımını yapmıştır. Görüldüğü gibi çeşitli programlara hizmet veren bir sistem programı olarak ta görmek mümkündür.

Şurası kesin bir gerçektir ki günümüzün gelişmiş BT' si, işletmeleri akla gelemeyecek kadar çok veriyi saklamaya yöneltmiştir. Bir tüketici hemen her yerde kayıt altına alınmaktadır. Bugün çoğu işletme müşterilerinin ilk alışverişlerinde adresi, yaşı, telefonu vb. çeşitli bilgilerini alarak bunları saklamaktadır. Daha sonra o müşterinin tüketim davranışını da her alışverişten sonra saklamaya devam etmektedir. Benzer şekilde Web üzerinden ticaret yapan işletmeler kullanıcıların geçmiş tüketim davranışlarını veya inceledikleri ürünleri saklamaktadır. Buradan da sadece işletmenin alacak, verecek ve mal stoku gibi temel fonksiyonlarına yönelik verilerin saklanmadığı açıkça görülmektedir.

Özetle bir VTYS, veri ile uygulama programları veya doğrudan kullanıcı arasında ara yüz görevi yapmaktadır. Her ne kadar kullanıcının veriye doğrudan erişebileceği varsayılmış olsa da bu daha çok profesyonel bilgi gerektirdiğinden az rastlanan bir durumdur. Fakat günümüzde her düzeydeki kullanıcılar çeşitli uygulama yazılımları aracılığıyla veriye erişmektedir. *Geleneksel dosya yaklaşımına* göre uygulama yazılımıyla veri arasında böyle bir ara yüz söz konusu değildir. Az da olsa geçmişte geliştirilmiş bazı yazılımlar halen böyle çalışabilmektedir. Bu tür yazılımlar kendi içerisinde her işi halletmektedirler. Buna göre bir dosyada personel bilgileri saklandığı düşünülsün. Tablo 2' de görüldüğü gibi dosyadaki her bir kayıt bir çalışanı temsil etmektedir. Böyle bir dosyada kolon başlıklar gibi bir kavram yoktur. Dolayısıyla programcı dosya biçimini kendi tasarladığından hangi verinin hangi pozisyonlar arasında yer aldığıni bilmektedir. Buna göre birinci ve ikinci pozisyonlarda personelin numarası, üçüncü ve yedinci pozisyonlar arasında adı, sekizinci ve 13. pozisyonlar arasında soyadı, 14. ve 21. pozisyonlar arasında da doğum tarihi yer almaktadır. Eğer uygulama yazılımında çalışanların yalnızca adı ve doğum tarihleri gösterilmek istenirse, programcı yazılımin, sırasıyla üçüncü ve yedinci pozisyonlar ile 14. ve 21. pozisyonlar arasındaki verileri çıktı olarak vermesini sağlayacak kodu yazmış olması gereklidir.

Tablo 2 Personel verilerinin tutulduğu dosya biçimi

| Kayıt No | Pozisyon | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
|----------|---------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 1 1 Ç a ğ l a C u r a | 0 | 1 | 0 | 2 | 1 | 9 | 7 | 9 | | | | | | | | | | | | | |
| 2 | 1 5 S e n e m A r s 1 a n | 2 | 2 | 0 | 4 | 1 | 9 | 8 | 2 | | | | | | | | | | | | | |

Görüldüğü gibi geleneksel dosya yaklaşımında uygulama yazılımıyla veri çok bağımlıdır. Veritabanındaki (geleneksel yaklaşım için dosyadaki) en ufak bir tasarım değişikliği için program koduna müdahale edilmesi gerekmektedir. Bununla beraber aynı veriler birbirinden farklı yerlerde birçok defa tekrar edebilir. Bu durum dosyalarda şısmelere neden olur. Örneğin *insan kaynakları* dosyasında verileri bulunan bir personelin *satışlar* dosyasında adı ve soyadı gibi verileri tekrar edecektir. Bu dosyada ayrıca ilgili personelin hangi üründen ne kadar sattığı da yer olacaktır.

Geleneksel dosya yaklaşımının çok önemli bir diğer eksikliği ise hataya fazlasıyla elverişli olmasıdır. Yukarıda da izah edildiği gibi kayıtlar byte dizileri biçiminde saklanmaktadır. Burada her bir byte bir karakter gibi düşünülebilir. Böylece karakterlerin pozisyonları sayılarak veri tüm kayıttan çekilmektedir. Bu sebeple pozisyonların sayılması ufak bir hata, ismi "Çağla" olan personelin "1Çağl" veya "ağlaC" biçiminde gösterilmesine neden olabilir.

Herhangi bir veri üzerinde yapılacak değişikliğin, o verinin bulunduğu tüm dosyalardaki tüm kayıtlara uygulanması gerekmektedir. Örneğin isminin yanlış girildiği fark edilen bir personelin ismi, *insan kaynakları* dosyasında düzeltilmekle kalmaz, satışlar vb. tüm dosyalarda da düzeltilir. Tabi böyle bir düzeltme için tüm kayıtlar teker teker incelenmek durumundadır. Böylece işler oldukça yavaşlayacaktır. Günümüz uygulamalarında geleneksel dosya yaklaşım örnekleri yok denecek kadar azdır. Olanlar çok eski programlar veya çok küçük bir veri tabanına ihtiyaç duyan programlardır.

9.1. Varlık (Entity), Özellik (Attribute) ve Anahtar

Her ne kadar günümüz veri tabanı yaklaşımları, geleneksel dosya yaklaşımından çok farklı olarak ifade edilse de fiziksel olarak veriler birçok dosyada tutulmaya devam etmektedir. Ancak farklı olarak söz konusu dosyalara doğrudan uygulama yazılımı tarafından erişilmez. Bunun yerine VTYS' leri bu dosyaları yönetir.

Daha önce de dephinildiği gibi bir dosya pek çok kayıttan oluşur. Her dosya belli bir amaca yönelik kayıtları saklar. Örneğin öğrenciler isimli bir dosyada öğrencilerle ilgili veriler saklanacaktır. Bu veriler öğrencilerin numaraları, adları, soyadları, doğum tarihleri, cinsiyetleri, ev adresleri ve telefon numaraları olabilir. Dosyalar kayıtlardan meydana gelir ve her bir kayıt bir varlığı temsil etmektedir. Doğal olarak varlık aslında çoğu zaman bir nesnedir. Örneğin öğrenciler dosyasında yer alan her bir kayıt bir öğrenciye karşılık gelmektedir. *Satışlar* dosyası söz konusu olsaydı bu durumda her bir kayıt şüphesiz yapılan bir satışa karşılık gelecekti.

Bilindiği gibi her bir varlığın çeşitli *özellikleri* vardır. Hatta kendi aralarında da aynı tür varlıklar bu özelliklerine göre ayrırlar. Bir insanla bir köpeği birbirinden ayırmak son derece kolaydır. Çünkü bunlar farklı tür varlıklardır. Ancak iki insanı birbirinden ayırmak için bu kez söz konusu insanların özelliklerine bakmak gerekecektir. Örneğin saç rengi, boy uzunluğu kilo vb. insan varlığının özelliklerindendir. Bu durumda öğrenciler dosyasında her bir kayıt bir varlık olduğuna göre her bir varlığın özellikleri de numara, adı, soyadı, doğum tarihi, cinsiyet, ev adresi ve telefon numarası olmak üzere yedi adet olacaktır.

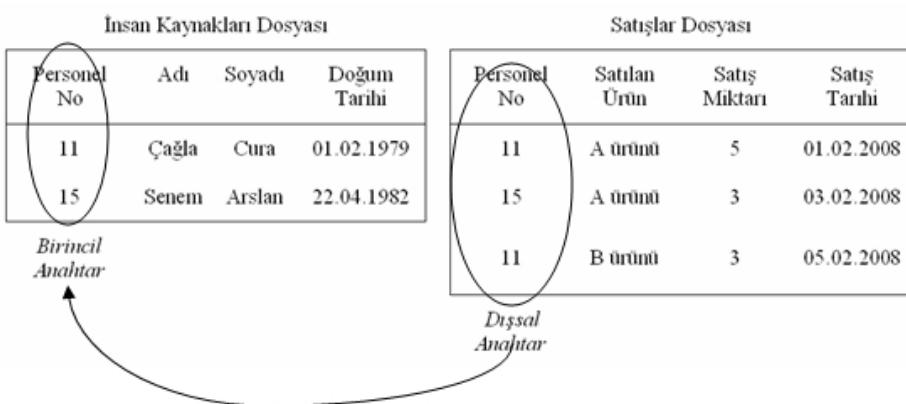
İki adet anahtardan bahsetmek mümkündür: *birincil anahtar* ve *dışsal anahtar*. Bunlardan birincil anahtar varlıkları birbirinden ayırt eden nitelikte bir veya birden fazla özelliği göstermektedir. Başka bir ifadeyle bu anahtar söz konusu varlığı benzersiz kılacak nitelikteki bir özelliği işaret etmektedir. Örneğin öğrenciler tablosunda *öğrenci numarası* özelliği, her bir öğrenci (varlık) için kuşkusuz farklı olacaktır. Bunun dışındaki tüm özellikler farklı öğrencilerde aynı olabilir. Buna göre bir özellik eğer birincil anahtar olarak tanımlanırsa;

- Dosyadaki diğer varlıklarda (kayıtlarda) o özelliğin aynı değer almaması, tek olması gereklidir.
- Söz konusu özellik hiçbir zaman boş değer alamaz.

Boş değer almak aslında boş bırakmak, yani hiçbir değer girmemek anlamını taşımaktadır. Buna rağmen pek çok orijinal kaynakta “null” veya “nil” değer atamak biçiminde kullanılabildiğinden [bkz. Connoly ve ark.-1999] bu çalışmada da “boş değer atamak” biçiminde kullanılmıştır. Hemen hatırlatmak gereklidir ki bazen boş bırakmak, sıfır değerinin girilmesi veya boşluk tuşuna basılması ile karıştırılmaktadır. Gerek sıfır değeri, gerekse boşluk karakteri bir veri olarak saklanmaktadır. Oysa boş değer almak, söz konusu varlığın ilgili özelliği için hiçbir veri saklanmaması anlamını taşımaktadır. Aslında bir ilişkinin fazla “null” değere sahip olması istenen bir durum değildir. Zira “null” değere sahip olan bir özelliğin değeri akla yatkın pek çok yorumda neden olabilir. En temel iki yorum ise değerin olduğu ancak bilinmediği veya değerin olmadığı biçimindedir. Bazı yazarlar ilişki içinde yer alan “null”ların gerçek değerlerini hesaplamaya çalışmışlardır. Bunlar çeşitli yöntemlerle boş değer içeren satırları, benzer satırlarla kıyaslayarak boş özelliğin değerini hesaplamaktadırlar. Benzer satırlar ise diğer özellik değerlerinin karşılaştırılmasıyla bulunmaktadır [bkz. Chen ve Huang-2008].

Uygulamada bazen herhangi bir özelliğin birincil anahtar olarak tanımlanması mümkün olamayabilir. Bu tür durumlarda fazladan bir özellik ilave edilerek eşsiz olan rastlantısal sayılar verilir. Böylece o özelliğin birincil anahtar olması sağlanır. Örneğin öğrenciler dosyasında eğer öğrencilerin numaraları özelliği olmasaydı, bu durumda ilave bir x özelliği eklenerek buna rastlantısal eşsiz sayılar verilebilirdi. Böylece x birincil anahtar olarak tanımlanabilirdi. Genellikle VTYS’leri bu tür rastlantısal eşsiz sayıları kendiliğinden üretebilmektedirler.

Dışsal anahtar da bir özelliği niteler. Ancak adından da anlaşılacağı gibi söz konusu özellik dışarıdaki bir dosyanın özelliğine bağlı olmaktadır. Bu özellik ise dışarıdaki dosyada birincil anahtar olmaktadır. Yukarıdaki örneğe bağlı kalınacak olursa, *insan kaynakları* dosyasında *personel numarası* özelliği kuşkusuz birincil anahtar olacaktır. Bununla beraber *satışlar* dosyasında artık satışı yapan personelin adı ve soyadı değil yalnızca numarasının bulunması yeterlidir. Şekil 9’da görüldüğü gibi *insan kaynakları* dosyasında varlığı benzersiz kılan *personel no* özelliği ve böylece birincil anahtar olmaktadır. Ancak satışı yapan personeli temsil eden personel numarası *satışlar* dosyasında dışsal anahtar olmaktadır. Zira söz konusu personelin diğer verileri *insan kaynakları* dosyasında bulunmaktadır. Görüldüğü gibi dışsal anahtar olan bir özellik, bağımlı olduğu birincil anahtar özelliğin sahip olduğu değerlerden başka değer alamamaktadır. Bununla beraber şekil 6.1’de görüldüğü gibi bir personel birden fazla satış yapabileceği için, dışsal anahtar olan özellikte veri tekrarları olabilir.



Şekil 9 Birincil ve dışsal anahtarlar

Bu anlatılanlar ışığında bir veritabanının hiyerarşik yapısı da şekillenmektedir. Buna göre bir veritabanı dosyalardan meydana gelir. Dosyalar ise kayıtlardan, başka bir ifadeyle varlıklardan meydana gelmektedir. Her bir varlık ise özelliklere sahiptir. Özellikler bir veritabanında bölünebilen en ufak parçalar olarak kabul edilir. Her ne kadar bir özelliğin sahip olduğu değer bit ve byte' lardan meydana geliyor olsa da özelliği parçalara ayırmayan genellikle bir anlamı yoktur.

9.2. Veritabanı Yaklaşımının Yararları

Daha önce veri saklamak ve ihtiyaç duyulduğunda bu veriler üzerinde çeşitli işlemler yapabilmek için en ilkel yöntemden geleneksel dosya yaklaşımı olduğuna deðinilmiştir. Geleneksel dosya yaklaşımının yerini günümüzde artık veritabanı yaklaşımı almıştır. Böyle bir yaklaşım geleneksel yaklaşımla kıyaslandığında önemli yararlar sağladığını görülmektedir. Bu yararlar kısaca şu şekilde sayılabilir [Yarımaðan-2000]:

- Veri tekrarının azalması,
- Veri bütünlüğünün sağlanması,
- Veri paylaşımının sağlanması,
- Kullanımının yaygınlaşması,
- Verilerin gizliliðinin ve güvenliğinin sağlanması,
- Standartların uygulanabilmesi.

9.2.1. Veri Tekrarının Azalması

Veri tekrarlarının azaltılmasına zaten deðinilmiştir. Hatırlanacağı gibi geleneksel yaklaşımda her alt-sistem için veriler ayrı ayrı tutulduğundan her biri için aynı verilerin tekrar etmesi söz konusu olmaktadır. Örneðin insan kaynakları için çalışanların personel bilgileri tutulurken, bunların yaptığı satış kayıtlarında isim ve soy isimleri tekrar edecktir. Böyle bir durum bellek miktari sorununun yanı sıra yapılabilecek hatalar yüzünden tutarsızlıklara yol açabilir. Bu hatalar bazen fark edilemeyecek düzeyde olabilir. Örneðin insan kaynakları dosyasında ismi "Çağla" olarak yazılan personelin satışlar dosyasında ismi "Çagla" olarak yazılsrsa, bunlar uygulama yazılımında farklı kişilermiş gibi algılanabilir.

9.2.2. Veri Bütünlüğünün (Data Integrity) Sağlanması

Veri bütünlüğü, verinin doğru ve tutarlı olması anlamını taþımaktadır. Herhangi bir verinin doğru ve tutarlılığından bahsedebilmek için bazı denetimlerin yapılması mecburdur. Çok basit olarak personelin *doðum tarihi* özelliði dikkate alınacak olursa "01.01.1000" gibi bir değer almaması gerekliliði açıkça ortadadır. Bununla beraber bir çalışanın baðlı olduğu bölümde göre belirli ürünleri satabilmesi mümkün olabilir. Bu durumda veri giriþi yapıldıken söz konusu satışının bölümune bakarak ilgili satışı yapıp yapamayacağı denetlenmelidir. Geleneksel yaklaşımda bütün bunlar uygulama yazılımında programcı tarafından kodlanarak denetlenmesi gerekirken, veritabanı yaklaşımıyla bu iş çok basit bazı ayarlarla halledilebilmektedir.

9.2.3. Veri Paylaşımının Sağlanması

Geleneksel yaklaşımında uygulama yazılımıyla veri dosyaları bir bütündür. Başka bir ifadeyle, dosyalardaki verilere ulaşabilmek için uygulama yazılımına ihtiyaç duyulmaktadır. Bu durum bir MS Excel çalışma dosyasının hemen her bilgisayarda bulunan basit “not defteri” uygulama yazılımıyla açılmasına benzetilebilir. Böyle bir durumda dosyanın pek çok anlamsız karakterden olduğu görülmektedir. Bu dosya ancak MS Excel yazılımıyla birlikte anlamlı olmaktadır. Oysa günümüzde veritabanı ayrı, uygulama yazılımı ayrı olduğundan aynı anda farklı ortamlardan veritabanına erişim mümkün olmaktadır.

9.2.4. Kullanımının Yaygınlaşması

Veritabanı yaklaşımının veri paylaşımını sağlayabilme özelliği beraberinde verinin kullanımının yaygınlaşmasını da sağlamıştır. Geleneksel yaklaşımda yalnızca bir uzmanın veriye ulaşması mümkündü. Sıradan kullanıcılar ise yalnızca uzmanların geliştirmiş oldukları uygulama yazılımı aracılığıyla veriye erişebilmekteydi. Oysa veritabanı yaklaşımı sayesinde verinin kullanım alanı çok genişlemiştir.

9.2.5. Verilerin Gizliliğinin ve Güvenliğinin Sağlanması

Veritabanı yaklaşımında veri yapısıyla birlikte veriye erişim hakkı bulunan kullanıcılar ve bunların yetkileri belirlenir. Daha sonra VTYS, kullanıcıların yetkilerine göre istedikleri veriye erişmelerine izin verir ya da vermez. Kuşkusuz ki kullanıcılarla birlikte şifreleri de oluşturulmaktadır. Örneğin *insan kaynakları* dosyasını insan kaynakları bölümünün dışındaki çalışanların görmemesi istenebilir. Bunun da ötesinde bir dosyadaki bazı özelliklerin görüntülenip, bazlarının saklanması istenebilir. Hatta dosyalar üzerindeki işlemler dahi yetkiye göre kısıtlanabilir. Örneğin bir dosyadaki veriler herhangi bir kullanıcı tarafından incelenebilir. Ancak aynı kullanıcının veriler üzerinde değişiklikler yapması yasaklanabilir. Tüm bu denetim VTYS tarafından gerçekleştirilmektedir.

9.2.6. Standartların Uygulanabilmesi

Veritabanı yaklaşımına göre veritabanından sorumlu olan ve *veritabanı yöneticisi* olarak adlandırılan tek bir kişi vardır. Böylece veriler üzerinde merkezi bir denetim sağlanır. Çünkü veritabanının tasarılanmasından işletimine kadar bu kişi sorumludur. Bazen kişi yerine bir ekip olduğu da görülmektedir. Gerek VTYS' nin yapısı, gerekse de bir sorumlunun olması uygulamalarla ve verilerle ilgili her türlü standardın benimsenmesi ve uygulanmasını olanaklı kılar. Söz konusu standartlar verinin yapısı, gösterimi, adlandırılması gibi değişik konularla ilgili; kurum içi, kurumlar arası, ulusal veya uluslararası konulmuş olabilir.

9.3. Veritabanı Modelleri

Günümüz VTYS' leri varlık, özellik ve ilişkilerin bağlantılarını takip edebilmek için farklı modeller kullanmaktadır. Her modelin kendine göre bazı avantaj ve dezavantajları olsa da bunlar içerisinde özellikle ilişkisel modeli bir kenara almak gerekmektedir. Zira günümüz VTYS' lerinin büyük çoğunluğu bu modeli kullanmaktadır.

Bilindiği gibi geleneksel dosya yaklaşımında her bir alt sistem için aynı veriler defalarca tekrar edebilmektedir. Böyle bir duruma karşı veritabanı yaklaşımının getirdiği en önemli fayda, veriler arasında çeşitli bağlantılar kurarak, aynı veri eğer bir alt sistemde yer alıysorsa, bir diğerinde bunun tekrarlanmasını önlemesidir. Söz konusu bağlantı biçimleri kısaca veritabanı modeli olarak özetlenebilir.

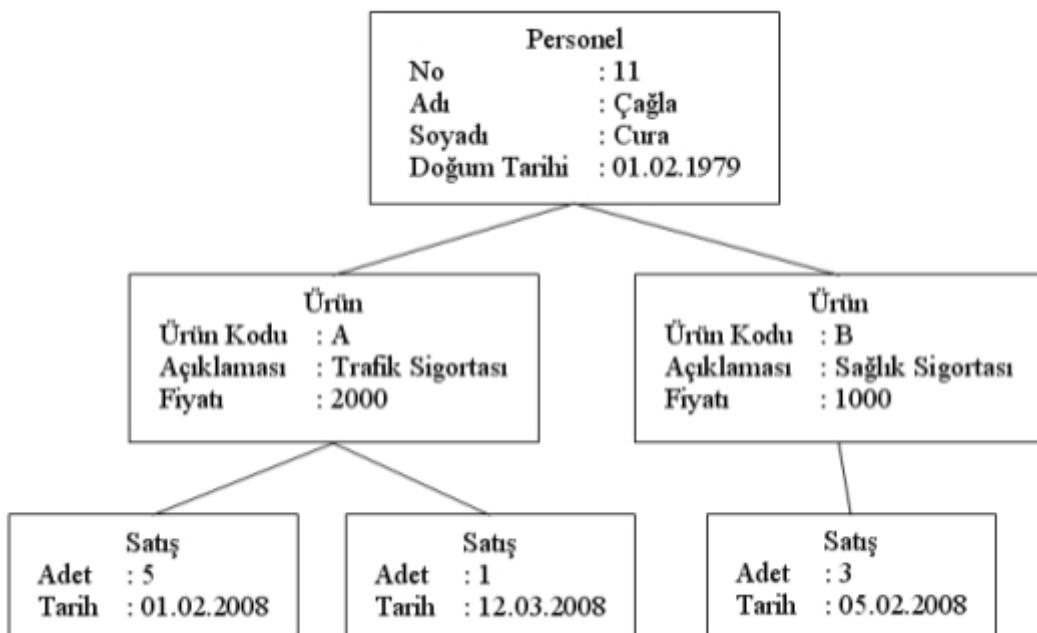
9.3.1. Hiyerarşik (Ağaç) Model

Verilerin, adeta bir organizasyon şeması gibi yukarıdan aşağıya doğru hiperarşik olarak organize edildiği bir modeldir [Gökçen-2005]. Böyle bir yapı şeklinde bir ağaç yapıya benzediğinden, ağaç olarak ta adlandırılmaktadır. Buna göre ağaçın en üstünde en fazla tekrarlaması beklenen veri yer almaktadır. Bunun tek olması gerekmektedir. Dolayısıyla bir hiperarşik modele sahip veritabanında birden fazla ağaç olabilmesi mümkündür. Buna göre bir ağaç modelde *bire çok* (one to many veya 1:N) ilişkiler söz konusudur. Bire çok ilişki hiperarşik modelin belirleyici özelliğidir. Bu ilişki biçimine göre her bir kayıt bir *ebeveyne* (parent) sahip iken, kendinden sonra gelen birden fazla *cocuğa* (child) sahip olabilmektedir. Şekil 10' deki örnekten hareketle bir satışının farklı ürünleri satabildiği görülmektedir. Buna göre geleneksel yaklaşım için her satış

kaydında, satıcının bilgileri ve ardından ürünün bilgileri tekrar etmek durumundayken, şekil 6.2' de görüldüğü gibi tekrarlar önlenmiştir. Böylece 11 numaralı personel kodu A olan üründen iki, B olan üründen ise bir satış yapmaktadır. Ancak görüldüğü gibi bunlar birbirine bağlandığından, her bir satış kaydında ürünle ve personelle ilgili bilgilerin tekrar girilmesine gerek kalmamaktadır.

Bu modelde ilişkiler arasındaki bağlantılar, kayıtlardaki adres veya işaretçilerle takip edilir. Böylece, her kayıt ilk çocuğunun ve varsa kendisiyle aynı seviyedeki diğer kaydın adresini bilir. Buna göre kodu A olan ürünün kaydı ilk çocuğu olan 01.02.2008 tarihindeki 5 adetlik satış kaydıyla, kendisiyle aynı seviyedeki kodu B olan ürün kaydının adreslerini bilmektedir. Arama da bu esasa göre gerçekleşir. Başka bir ifadeyle, 11 numaralı personelin, kodu B olan üründen 05.02.2008 tarihinde yaptığı satışı bulmak için VTYS aramayı söyle yapacaktır:

- Personel kaydı, ilk çocuğu olan A kodlu ürün kaydına gider.
- Aradığı bu olmadığı için A' nin aynı seviyedeki komşusu B' ye gider.
- Aradığı bu ürünü ve bunun ilk çocuğuna (örnekte zaten başka çocuğu yok) gider.



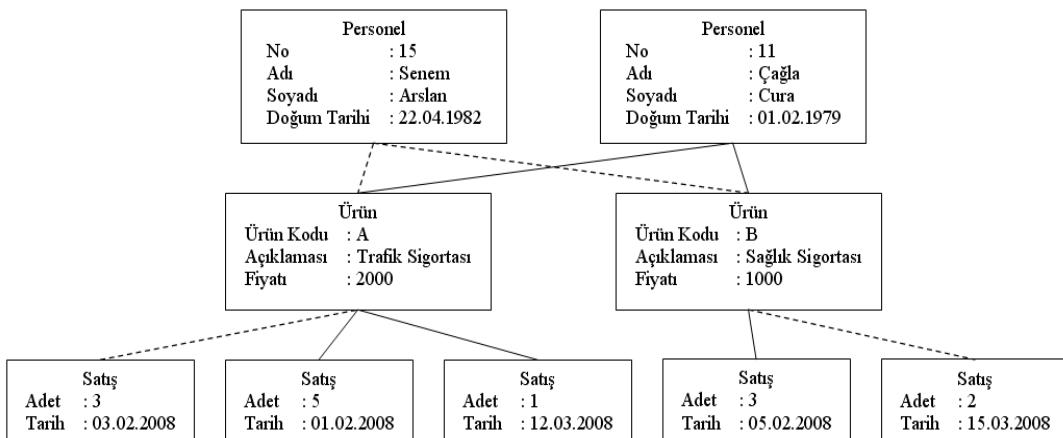
Arama işlemlerini hızlı yapsalar da bu modelin en önemli dezavantajı esnek olmayıdır. Başka bir ifadeyle, en başta yapılmış olan veritabanı tasarımlı daha sonra kolayca değiştirilemez. Bununla birlikte yalnızca bire çoklu ilişkilere izin verip, *çoğa çoklu* (many to many veya N:N) ilişkilere izin vermez. Bu sebeple gereksiz veri tekrarları yapılabilir. IBM IMS, FOCUS, RAMIS II gibi bazı VTYS' leri bu modeli kullanmaktadır [Gökçen-2005].

9.3.2. Ağ Model

Ağ model, hiperarşik model ile hemen aynı dönemlerde ortaya atılmış bir veritabanı modelidir. Bunda hiperarşik modelin sıkılıkla ihtiyaç duyduğu çoğa çoklu ilişki sağlanmıştır. Çoğa çoklu ilişki kısaca, bir kaydın birden fazla çocuğunun olabileceği gibi birden fazla da ebeveyninin olması olarak özetlenebilir. Böyle bir yaklaşım özellikle gereksiz veri tekrarını önlemektedir. İlk olarak 1971 yılındaki Veri Sistemleri Dilleri Konferansı' nda (Conference on Data Systems Language veya CODASYL) ortaya çıktıği için CODALYS veritabanları olarak ta adlandırılmaktadır.

Bu kısımda daha önce verilen örnekten hareket edilecek olursa; Şekil 11' de görüldüğü gibi ağ modele göre ikinci satıcılarının kayıtlarını tutabilmek için ikinci bir ağaç yapısına ihtiyaç duyulmamaktadır. Başka bir ifadeyle, satışı yapılan ürünlerin kayıtları tekrarlanmamıştır. Her iki ürün kaydının da ikişer ebeveyne vardır. Eğer hiperarşik model kullanılsaydı, bu model iki ebeveyne izin vermediğinden, ikinci satıcı için ürün kayıtlarının tekrar oluşturulması gerekecekti.

Unutulmamalıdır ki ağ modelde adresleme yöntemleri farklıdır. Bu sebeple Şekil 11' de farklı ebeveynler ile çocukların arasındaki bağlantılar farklı çizilmiştir. Buna rağmen, adresleme yöntemlerinden birisi hiyerarşik modeldekine oldukça benzerdir. Böyle bir adreslemeye göre her bir personel kaydı kendisine bağlı ilk ürün kaydının adresini tutmakta, her ürün de aynı personele bağlı bir sonraki ürünün adresini tutmaktadır [Gökçen-2005].



Şekil 11 Ağ veritabanı modeli

9.3.3. İlişkisel Model

Diğer modellerde sıkılıkla kullanılan dosya kavramı ilişkisel modelle birlikte yerini *tablolara* bırakmıştır. Başka bir ifadeyle, her ne kadar fiziksel olarak kayıtlar hala dosyalarda tutuluyor olsa da, kavramsal olarak ilişkisel modelin tablolardan meydana geldiği söylenebilir. Bu model ilk kez Codd [1998] tarafından 1970 yılında ortaya atılmıştır. Her ne kadar ilişkisel modele göre tasarım yapabilmek için bilmek gerekmese de, temeli *ilişkisel cebire* (relational algebra) dayanmaktadır. Günümüzde birçok veritabanı uzmanı, ilişkisel cebir terminolojisini kullanmaktadır. Buna göre bir kayıt *tuple*, bir kolon *özellik* ve tablo da *ilişki* olarak isimlendirilebilmektedir. Unutmamak gerekdir ki her tablo ve her kolon kendisini diğerlerinden ayıran bir isme sahiptir. Daha önce tam olarak dephinmemiş olsa da Şekil 9' da yer alan tablolar aslında ilişkisel model örneğidir. Böylece, “insan kaynakları dosyası” ilişkisel modele göre “insan kaynakları tablosu” olarak isimlendirilmelidir. Hatırlanacağı gibi tüm kolonların zaten bir ismi vardır. Bunlara ilaveten, sıkılıkla kolon için *alan* (field) teriminin kullanıldığına da dephinmek gereklidir.

İlişkisel modelde hem bire çoklu, hem de çoğa çoklu bağlantı mümkündür. Bir veri tabanında her iki bağlantı türüne de ihtiyaç duyulabilmektedir. Örneğin Şekil 12' de gösterilen veri tabanına göre bir bölümde birden fazla çalışanın olması bire çoklu, bir ürünün birden fazla satıcı tarafından satılabilmesi ve bir satıcının birden fazla ürünü satabilmesi çoşa çoklu bağlantı olmaktadır. Bu durum birden fazla tablodaki kayıtların birbirine bağlanmasıyla sağlanmaktadır. Daha önce dephinilen birincil ve dışsal anahtarlar, ilişkisel modelde tablolar arası bağlantılarında önemli rol oynamaktadır. Ancak farklı tablolarda bulunan kayıtların bir araya getirilmesi için kesinlikle anahtar gerekmektedir. Böylece, ortak en az bir kolonu olan tablolar istenildiği gibi bir araya getirilebilir [Laudon ve Laudon-2006]. Oysa veriler bağımsız olarak farklı tablolarda tutulmaktadır.

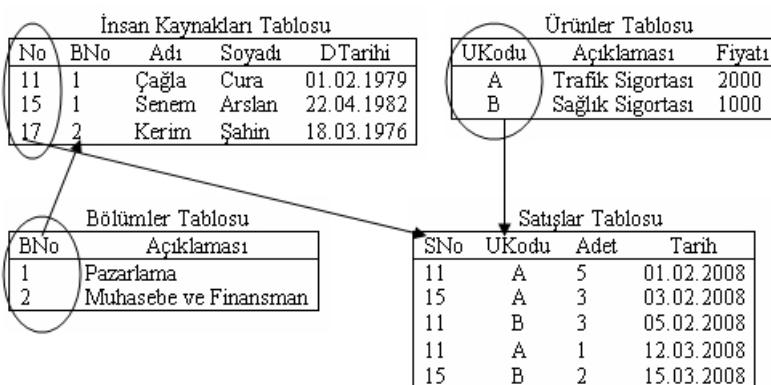
Kuşkusuz tabloların bir araya getirilmesi, veri tabanında kalıcı değişiklikler anlamına gelmeyecektir. Herhangi bir sorgu ya da rapor doğrultusunda ilişkili tablolar bir araya getirilir. Öyle ki, bu bir araya getirilişte bazı kayıtların (satırların) veya kolonların gösterilmemesi istenebilir. Örneğin Şekil 12' de gösterilen ve farklı tablolarda tutulan veriler bir yönetici için anlamsız olacaktır. Buna göre “05.02.2008” tarihi ve bu tarihten sonra yapılan satışlar için; satıcı yapanın adı, soyadı, bağlı olduğu bölümü, satıcı yapılan ürünün açıklaması, kaç adet satış yapıldığı ve bu satışın tarihini gösterecek şekilde sorgulanmak istediği düşünülsün. Görüldüğü gibi veriler farklı tablolardan alınarak bir araya getirilmek zorundadır. Zira veri tabanında bu verilerin hepsini birden barındıran bir tablo yoktur. İşte istege göre anlık olarak oluşturulan bu tablolara bazı yazarlar *birleşme tablo* (join table) demektedirler [bkz. Oz-2006]. Tablo 3' de görüldüğü gibi bunlarda bazı veriler defalarca tekrar edebilmektedir. Böylece tablo daha anlaşılır olmaktadır. Unutulmamalıdır ki birleşme tablolar veri tabanında tutulmamakta, istediği an oluşturulmaktadır. Dolayısıyla bunlara bazı özel teknikler haricinde veri girişi de yapılamaz. Bazen sanal tablolar

oluşturulabilir. Sanal tablolar birleşme tablo gibidirler. Ancak bunlar sanal olarak VTYS tarafından görüntülenir ve neyin neye karşılık geldiği bilinmektedir. Bir veri girişinde VTYS verileri gerekli tablolara kaydeder. Bu sanal tablolara bazı VTYS' lerinde *görüntü* (view) ismi verilmektedir [bkz. Çubukçu-1999]. Örneğin Tablo 3' deki gibi görünen bir görüntüye yeni bir kayıt girildiği düşünüsün: "Senem", "Arslan", "Pazarlama", "Trafik Sigortası", "2", "26.03.2008". Satışlar tablosu haricinde diğer tablolara veri girişi gerekmeyeceğinden yalnızca satışlar tablosuna "15", "B", "2", "26.03.2008" verilerini barındıran kayıt girilir. Hemen deşinmek gerekir ki kayıt girişi yapılacak bir görüntüde ilgili tabloların birincil anahtarları görüntülenmelidir. Örneğin insan kaynakları tablosuna kaydedilecek yeni bir personelin Tablo 3' dekine benzer bir görüntüde numarası bulunmamaktadır. Veri girişinde bu hatalara neden olacaktır.

Tablo 3 Bir birleşme tablo örneği

| Adı | Soyadı | Bölüm | Açıklaması | Ürün | Açıklaması | Adet | Tarih |
|-------|--------|-----------|------------|------------------|------------|------------|-------|
| Çağla | Cura | Pazarlama | | Sağlık Sigortası | 3 | 05.02.2008 | |
| Çağla | Cura | Pazarlama | | Trafik Sigortası | 1 | 12.03.2008 | |
| Senem | Arslan | Pazarlama | | Sağlık Sigortası | 2 | 15.03.2008 | |

İlişkisel modelde Şekil 12' de de görüldüğü gibi bağlantılar genellikle anahtarlar aracılığıyla yapılmaktadır. Yuvarlak içerisinde alınmış alanlar bulundukları tabloların birincil anahtarlarıdır. Bazı durumlarda tek bir alan birincil anahtar olarak tanımlanamaz. Bu tür durumlarda birden fazla alanın birlikte birincil anahtar olması sağlanabilir. Örneğin tüm üniversitelere yönelik merkezi bir veritabanı düşünüsün. Böyle bir durumda fakülte adı tek başına birincil anahtar olamaz. Zira pek çok mühendislik fakültesi, işletme fakültesi, tıp fakültesi vb. olabilir. Fakat üniversite adıyla beraber bunlar tek olacaktır. Böylece iki alan bir arada kullanılarak oluşturulan bu anahtara *birleşik anahtar* (composite key) denir.



Şekil 12 İlişkisel veritabanı modeli

9.3.4. Nesneye Dayalı Model

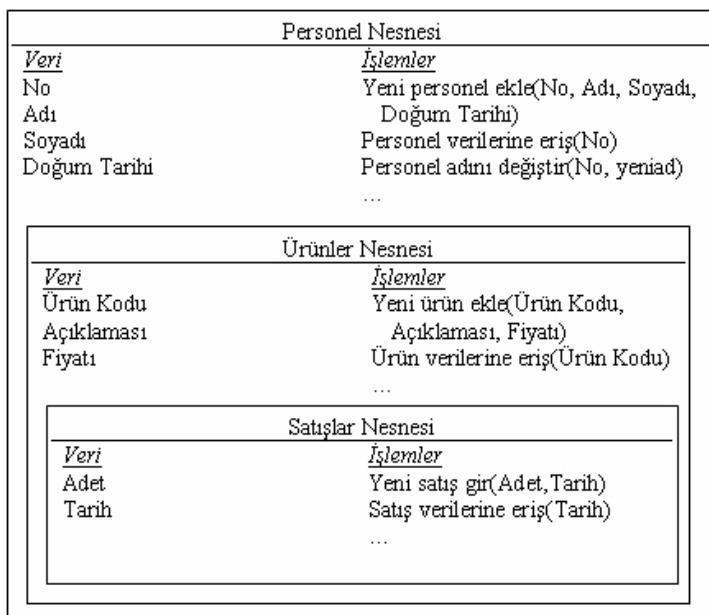
Nesneye dayalı veritabanı modeli dördüncü kısımda incelenmiş olan nesneye dayalı programlamayı temel alan bir modeldir. Bilindiği gibi bir nesne hem veriyi hem de veri üzerindeki işlemleri içerir. Daha da ötesi bir nesne başka nesneleri içinde barındırabilme yeteneğine de sahiptir. Bir veritabanında hem veri, hem veri üzerindeki işlemlerin bir arada muhafaza edilmesi *sarma* (encapsulation) olarak adlandırılmaktadır [Oz-2006]. İlk nesneye dayalı veritabanı (NDV) modelleri 1980' lerin sonlarına doğru ortaya çıkmıştır [bkz. Martin-1993].

Bu veri modeli özellikle artmakta olan farklı tipteki verilerin veritabanında saklanabilmesi için gerekmektedir. Örneğin görüntü, resim, ses gibi çoklu ortam veri tiplerinin ilişkisel modeldeki gibi satır ve kolonlar biçiminde ifade edilmesi oldukça güç olmaktadır. Herhangi bir görüntü verisi sayılar veya karakterler gibi değildir. Bunları görüntüleyebilmek için farklı programlar gerekecektir. Başka bir ifadeyle söz konusu verinin çeşitli işlemlere tabi tutulması gerekecektir. Veritabanı yaklaşımı bilindiği gibi belirli bir bilgisayardan bağımsız olmayı, merkezi bir noktadan veriyi paylaşmayı sağlamaktadır. Oysa görüntü verisinin bu anlatılanlara göre bahsi geçen işlemleri yapan bir program olmaksızın herhangi bir istemci bilgisayarda görüntülenebilmesi mümkün olmayacağından söz konusunu işlemleri ve veriyi bir arada

barındıran nesne, bilgisayardan bağımsız olarak veri üzerinde işlemleri gerçekleştirecek ve onu görüntüleyecektir.

Bir veritabanının önemli işlevlerinden birisi de sorgulamadır. Nesneye dayalı veritabanlarında farklı formattaki verilerin sorgulanması mümkün olmaktadır. Örneğin *Erozel ve ark.* [2008] konuşma dilinin kullanılarak görüntü verilerini sorgulanabilecegi bir yöntem önermişlerdir. Başka bir ifadeyle böyle bir veritabanında “içinde araba olan görüntüleri liste” benzeri bir sorgu oluşturmak mümkündür.

Şuna da deinmek gerekir ki bir NDV modeli söz konusu olduğunda nesneler arası bağlantıların sağlanabilmesi için herhangi bir dışsal anahtara da gerek kalmamaktadır. Bilindiği gibi bir nesne kendi içinde başka nesneler barındırabilir [Oz-2006]. Böylece Şekil 13’ te görülen iç içe geçmiş bir yapı söz konusu olmaktadır. Hemen hatırlatmak gerekir ki her ne kadar bu biçimde çizilmiş olsa da aslında bir nesne diğer nesnenin adresini kapsamaktadır. Başka bir ifadeyle şekilde görüldüğü gibi nesnenin tamamı, bir diğerinin içine gömülü olamamaktadır. Böylece birbirinden bağımsız iki nesne bir başka nesneyi kapsayabilmektedir.



Şekil 13 Nesneye dayalı veritabanı modeli

Dikkat edilirse NDV modelinde işlemler, VTYS tarafından veya istemci tarafında çalışan bir uygulama yazılımı tarafından gerçekleştirilmemektedir. Bunun yerine veriler kendi işlemleriyle istemciye iletilmektedir. Ardından istenen işlemler yerine getirilmektedir. Bu durum nesneye dayalı bir veritabanında yapılacak değişikliğin uygulama yazılımında da değişiklik gerektirmesine yol açmaktadır. Oysa ilişkisel modelde veritabanı üzerindeki değişiklikler nadiren uygulama yazılımlarında da değişiklik gerektirirler. Bu sebeple bazı melez (ilişkisel nesneye dayalı) VTYS’ leri görmektedir.

Bölüm Özeti

Bu bölümde e-ticaret için çok önemli bir konu olan veritabanı yaklaşımına girilmiştir. Öncelikle varlık, özellik ve anahtar gibi kavramlar ele alınmıştır. Ayrıca veritabanı yaklaşımının ortaya çıkışının sebepleri ve böyle bir yaklaşımın işletmelere sağladığı avantajlar ele alınmıştır. Daha sonra en temel veritabanı modelleri ele alınmıştır. Bunlardan özellikle ilişkisel model ve nesneye dayalı model günümüzde en fazla tercih edilen yaklaşım olmuştur.

Kaynakça

Chen S. M., Huang C. M. “A new approach to generate weighted fuzzy rules using genetic algorithms for estimating null values”, Expert Systems with Applications 35, 905–917, 2008.

Codd E. F., "A relational model of data for large shared data banks (Reprinted from Communications of the ACM, June, pg 377-87, 1970)", MD Computing 15, 162-168, 1998.

Connoly T., Begg C., Strachan A., Database Systems: A practical approach to design, implementation and management, Addison Wesley, İkinci baskı, 1999.

Çubukçu F., Microsoft SQL Server 7.0: Sistem Yönetimi uygulama geliştirme, Alfa Yayınları, 1999.

Erozel G., Çiçekli N. K., Çiçekli İ., "Natural language querying for video databases", Information Sciences 178, 2534–2552, 2008.

Gökçen H., Yönetim Bilgi Sistemleri, EPİ Yayınları, 2005.

Laudon K. C., Laudon J. P., Management Information Systems, Prentice Hall, Dokuzuncu baskı, 2006.

Martin J., Principles of Object-Oriented Analysis and Design, Prentice Hall, 1993.

Oz E., Management Information Systems, Thomson, Beşinci baskı, 2006.

Yarımaga Ü., Veri Tabanı Sistemleri, Akademi Yayınları, 2000.

10. VERİ TABANI YÖNETİM SİSTEMLERİ II

Birlikte Düşünelim

1. Normalizasyon nedir?
2. Varlık-ilişki diyagramlarının önemi nedir?
3. Veritabanlarının dağıtılması ne demektir?
4. Veri kalitesi nasıl sağlanır?
5. Veritabanı yönetim sistemlerinin ana bileşenleri nelerdir?

Başlamadan Önce

Veritabanı yönetim sistemlerinde kavramsal tasarım esasen bir veritabanı yöneticisinin veya uzmanının yaptığı en önemli işlerden birisidir. Zira verilerin fiziki olarak depolanması günümüzde veritabanı yönetim sistemleri geliştiricileri tarafından sağlanmaktadır. Kavramsal tasarım denilince de akla ilk olarak ilişkisel model ve dolayısıyla da normal biçimlerin öğrenilmesi gelmektedir. Ayrıca bu bölümde başarılı kavramsal tasarımın sağlanması için ihtiyaç duyulan ve veritabanı yapısının temsilinde kullanılan diyagramlar da ele alınacaktır. Günümüzde artık veritabanlarında çok büyük miktarda veri tutulmaktadır. Bu da son derece yoğun kullandıkları için veritabanı yönetim sistemini barındıran sunucuların performans sıkıntısınamasına neden olmaktadır. Dolayısıyla veritabanlarında yük dağıtıması veya birden fazla bölgeye veritabanının dağıtılmış konuları da büyük önem kazanmıştır. Veritabanlarında maalesef zaman içerisinde eksik ve hatalı verilerin girildiği çok sık yaşanan ve veritabanı üzerinde sağlıklı analizleri engelleyen bir durumdur. Bu nedenle veritabanlarında kalitenin sağlanmasına yönelik genel bilgiler verilecektir. Son olarak bir veritabanının ana bileşenleri genel hatlarıyla inceleneciktir.

Giriş

Bir veritabanının tasarımını iki açıdan ele alınır: kavramsal tasarım ve fiziksel tasarım. Bunlardan fiziksel tasarım verinin fiziksel olarak dosyalarda ve depolama aygıtlarında nasıl tutulacağını kapsamaktadır. Bu sebeple söz konusu tasarım kullanılan VTYS tarafından zaten sağlanmaktadır. Ancak kavramsal veya mantıksal tasarım ise işletmenin ihtiyaçlarına bağlı olarak oluşturulan soyut modeldir. Böylece kavramsal

tasarım için işletmede ihtiyaç duyulan enformasyonun detaylı olarak ortaya konulması gerekmektedir [Laudon ve Laudon-2006].

10.1. Veri Tabanın Tasarlanması

Kavramsal tasarım bir veri tabanındaki veri öğelerinin nasıl gruplanması gerektiğini de kapsamaktadır. Veri ögesi, daha önce ele alınmış olan özelliğe karşılık gelmektedir. Hatırlanacağı gibi bir veri tabanı birden fazla ilişkiden (tablodan) meydana gelebilmekteydi. Ancak bu kesinlikle mecburi değildir. Bu sebeple tek ilişkisi olan bir veri tabanı tasarlanabilir. Böylece tüm özellikler bir grup içinde toplanmış olacaktır. Bu durum daha önce debynildiği gibi pek çok gereksiz veri tekrarına neden olmaktadır. Görüldüğü gibi özelliklerin gruplanması aslında bir ilişkinin bölünerek birden fazla ilişki meydana getirmesidir. İşte bu gereksiz veri tekrarını en azı indirme ve birbiriyile bağıntılı veri öğelerini bir araya getirme gayreti *normalizasyon* olarak adlandırılır. Buna göre ilişkiler için dört normal biçim vardır. Bunlar sırasıyla *birinci normal biçim* (1NF), *ikinci normal biçim* (2NF), *üçüncü normal biçim* (3NF) ve *Boyce Codd normal biçim*dir (BCNF) [bkz. Yarımağan-2000].

10.1.1. Birinci Normal Biçim (1NF)

Birinci normal biçim ilişkiaslında bir ilişkisel modelde olması gereken ilişki biçimini ifade eder. Başka bir değişle en temel biçimdir. Bunun dışında bir biçim tamamen hatalı bir tasarım anlamına gelmektedir. Bu biçim, bir ilişkide her özelliğin tek değer alması gerektiğini ve aynı özelliklerin tekrarlamaması gerektiğini söylemektedir. Bir özelliğin tek değer alması, onun matris veya vektör gibi birden fazla değer içeremeyeceği anlamını taşımaktadır. Başka bir ifadeyle bir ilişkideki özellik kesinlikle bir alt ilişki olamaz. Şekil 14' de görüldüğü gibi *bölüm*ler ilişkisinin *personel* özelliği bir matris (başka bir ilişki) biçimindedir. Ayrıca *bölüm*ler ilişkisindeki *personel* matrisinin her bir kolonunun doğrudan alınması yeterli olmayacağıdır. Çünkü aynı özellikten iki tane olacaktır (*BNo* özelliği). Böylece 1NF' ye uymamakta olan *bölüm*ler ilişkisi, *BölümPers* ilişkisine dönüştürüldüğünde 1NF' ye uymaktadır.

| | | Bölüm | | | | BölümPers | | | | |
|-----|-----------------------|----------|-----|-------|--------|-----------|-----------------------|----|-------|--------|
| BNo | Açıklaması | Personel | | | | BNo | Açıklaması | No | Adı | Soyadı |
| | | No | BNo | Adı | Soyadı | | | | | |
| 1 | Pazarlama | 11 | 1 | Çağla | Cura | 1 | Pazarlama | 11 | Çağla | Cura |
| | | 15 | 1 | Senem | Arslan | 1 | Pazarlama | 15 | Senem | Arslan |
| 2 | Muhasebe ve Finansman | 17 | 2 | Kerim | Şahin | 2 | Muhasebe ve Finansman | 17 | Kerim | Şahin |

Şekil 14 1NF olmayan bir ilişkinin 1NF' ye dönüştürülmesi

10.1.2. İkinci Normal Biçim (2NF)

Bir ilişkinin ikinci normal biçimde olduğunu söyleyebilmek için bunun birinci şartı söz konusu ilişkinin öncelikle birinci normal biçimde olmasıdır. Buna göre, birinci normal biçimde olan bir ilişkide yer alan birincil anahtar olmayan tüm özelliklerin, fonksiyonel olarak birincil anahtara bağlılı olması, bu ilişkinin 2NF olduğu anlamını taşıyacaktır. Hatırlanacağı üzere bazen birincil anahtar birden fazla özelliğin bir araya gelmesiyle de oluşabilmektedir. Bu tür durumlarda ikinci normal biçimden bahsedebilmek için, ilişkideki tüm birincil anahtar olmayan özelliklerin birincil anahtarın tümüne bağlılı olması gerekmektedir.

Bu durumu izah edebilmek için Şekil 14' de görülmekte olan *BölümPers* ilişkisi ele alınabilir. Bilindiği gibi bu ilişki birinci normal biçimdedir. Söz konusu ilişkinin birincil anahtarını, *BNo* ve *No* özelliklerinin birlikte oluşturduğu varsayılsın. Bunun anlamı farklı bölgelerde aynı numaraya sahip farklı çalışanlar olabileceğidir. Yalnızca *No* özelliğe bağlılığı giri görünse de gerek *Adı* gereksiz de *Soyadı* özellikleri birincil anahtara tamamen bağlılıdır. Zira aynı numaradan başka bir bölümün çalışanına da verilmiş olabilir. *Açıklaması* özelliğinde yer alan bölüm açıklaması ise yalnızca *BNo* özelliğe bağlılı olup, birincil anahtarın tamamente değildir. Bu durumda *BölümPers* ilişkisinde *Açıklaması* özelliğinin kaldırılmalı ve ayrı bir ilişkide tutulmalıdır. Şekil 15' de görüldüğü gibi ikinci bir ilişki oluşturularak (*Bölüm*ler), her ikisinin ikinci normal biçimde olması sağlanmıştır.

BNo No Adı Soyadı BNo Açıklaması

- | | | | |
|---|-----------------|---|-----------------------|
| 1 | 11 Çağla Cura | 1 | Pazarlama |
| 1 | 15 Senem Arslan | 2 | Muhasebe ve Finansman |
| 2 | 17 Kerim Şahin | | |

Şekil 15 2NF olmayan bir ilişkinin 2NF' ye dönüştürülmesi

10.1.3. Üçüncü Normal Biçim (3NF)

Bir ilişkinin üçüncü normal biçimde olabilmesi için öncelikle 2NF olması gereklidir. Böylece ikinci normal biçimde olan bir ilişkide birincil anahtar dışında kalan tüm özelliklerin yalnızca birincil anahtarla bağlantılarının olması, bunun yanı sıra birbirleri arasında bağlantılarının olmaması gerekmektedir. Örneğin daha önce Şekil 12' de gösterilen *İnsankaynakları*[3] ilişkisinde ufak değişiklik yapılsın ve çalışanın bölüm bilgisi çıkarılıp, yerine oturduğu bölge bilgileri Şekil 16' da görüldüğü gibi eklenmiş olsun.

| İnsankaynakları | | | | | | |
|-----------------|-------|--------|------------|----------|----------|-----------|
| No | Adı | Soyadı | DTarihi | Şehir | İlçe | Postakodu |
| 11 | Çağla | Cura | 01.02.1979 | İstanbul | Bakırköy | 34142 |
| 15 | Senem | Arslan | 22.04.1982 | İstanbul | Avcılar | 34310 |
| 17 | Kerim | Şahin | 18.03.1976 | İstanbul | Bakırköy | 34142 |

↓

| İnsankaynakları | | | | |
|-----------------|-------|--------|------------|-----------|
| No | Adı | Soyadı | DTarihi | Postakodu |
| 11 | Çağla | Cura | 01.02.1979 | 34142 |
| 15 | Senem | Arslan | 22.04.1982 | 34310 |
| 17 | Kerim | Şahin | 18.03.1976 | 34142 |

→

| Adresler | | |
|-----------|----------|----------|
| Postakodu | Şehir | İlçe |
| 34142 | İstanbul | Bakırköy |
| 34310 | İstanbul | Avcılar |

Şekil 16 3NF olmayan bir ilişkinin 3NF' ye dönüştürülmesi

Böylece *İnsankaynakları* ilişkisinde *No* özelliği birincil anahtar olmaktadır. Diğer tüm özellikler ise birincil anahtara bağlılıdır. Bu durumda ilişki ikinci normal biçimdedir. Ancak çalışanın adresiyle ilgili verileri içeren *Şehir*, *İlçe* ve *Postakodu* özellikleri birbirile bağlılıdır. Bu nedenle *İnsankaynakları* ilişkisindeki birbiriyle bağlılı bu özellikler başka bir ilişkiye şekil 6.8' te görüldüğü gibi aktarılarak üçüncü normal biçim sağlanmıştır.

10.1.4. Boyce Codd Normal Biçimi (BCNF)

Boyce Codd normal biçimini açıklamak için kısaca belirleyicinin ne olduğuna degeinmek gerekir. Bilindiği gibi birincil anahtar bir varlığın eşsiz belirleyicisidir. Buradan belirleyicinin birincil anahtar olduğu çıkarımı yapılabilir. Bu kısmen doğrudur. Zira her zaman belirleyiciler aynı zamanda birincil anahtar olmamaktadır. Bu durumda *belirleyici* için bir varlığı temsil etme yeteneğine sahip olan özellik tanımını yapmak yanlış olmayacağından. Böylece bir ilişkide tüm belirleyiciler aynı zamanda birincil anahtar ise o ilişki için Boyce Codd normal biçimindedir denilir. Sıklıkla üçüncü normal biçimdeki ilişkiler aynı zamanda Boyce Codd normal biçimindedir. Bu normal biçim ancak bir ilişkide birden fazla özelliğin birincil anahtar olabilmesi durumunda söz konusu olmaktadır.

Örnek olarak işletme çalışanlarının farklı eğitimlerce, birbirinden farklı eğitimlere tabi tutulduğu düşünülsün. Bu durumda bir eğitim birden fazla çalışan tarafından alınmakta olsun. Aynı şekilde bir eğitim birden fazla eğitim tarafından verilebilmekte olsun. Ancak bir eğitmenin birden fazla eğitim veremediği varsayılsın.

Tablo 4 BCNF olmayan EğitimProg ilişkisi

| ÇalışanNo | Eğitim | Eğitmen |
|-----------|--------|-----------|
| 15 | NLP | Ayhan Sel |

| | |
|----|----------------------------|
| 11 | Risk Yönetimi Veli Nar |
| 15 | Risk Yönetimi Nazan Zengin |
| 11 | NLP Ayhan Sel |
| 17 | NLP Cengiz Seymen |

Bu durumda söz konusu veriler Tablo 4' de görülen *EğitimProg* isimli ilişkide yer alacak olursa, birincil anahtar olarak *ÇalışanNo* ve *Eğitim* özellikleri bir arada tanımlanabilir. Çünkü bir çalışan aynı eğitimi iki kez alamaz. Ancak bu durumda anahtar olmayan *Eğitmen* özelliği de belirleyici niteliktedir. Zira *Eğitim* ve *Eğitmen* özellikleri bir arada birincil anahtar olarak tanımlanabilirdi. Her iki durumda da *EğitimProg* ilişkisi üçüncü normal biçimde olmasına rağmen BCNF değildir. Eğer Şekil 17' da görüldüğü gibi iki ilişkiye bölünürse, her iki ilişki de Boyce Codd normal biçiminde olacaktır.

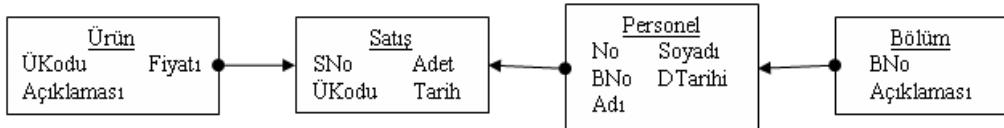
| PersEğitmen | EğitmenEğitim |
|------------------|---------------------------------|
| No Eğitmen | Eğitmen Eğitim |
| 15 Ayhan Sel | Ayhan Sel NLP |
| 11 Veli Nar | Veli Nar Risk Yönetimi |
| 15 Nazan Zengin | Nazan Zengin Risk Yönetimi |
| 11 Ayhan Sel | Cengiz Seymen NLP |
| 17 Cengiz Seymen | |

Şekil 17 BCNF olan iki ilişki

10.2. Varlık-İlişki Diyagramları

Veritabanı tasarımcıları, veri modellerini varlık-ilişki diyagramlarını kullanarak kâğıt üzerine aktarır ve bu şekilde saklarlar. Bunlar tasarım anında kullanıldığı gibi, sonraları başka veritabanı uzmanlarının değişiklik yapmayı düşündüklerinde de yararlanılan evraklar olmaktadır. Daha önce Şekil 12' de gösterilmiş olan veri tabanı için örnek bir varlık-ilişki diyagramı Şekil 18' de görülmektedir.

Varlık-ilişki modellerinde her varlık bir kutu ile temsil edilmektedir. Bu kutuda varlığın adı ve özellikleri yer alır. Şekil 18' e göre varlık ismi altı çizili olarak verilmiştir. Kutular arasındaki bağlantılar ise varlıklar arasındaki ilişkiyi göstermektedir. Ayrıca burada yer alan örneğe göre bağlantıların başı ve sonunda kullanılan şekiller vardır ki bunlar "ok" ve "daire"dir. Buna göre daire bir, ok ise çok anlamına gelmektedir. Böylece *bölüm* ile *personel* varlıklarını arasındaki ilişki bire çokludur. Görüldüğü gibi bağlantının iki ucu daire olursa bire bir, her ikisi de ok olursa çoga çoklu ilişki söz konusu olacaktır.



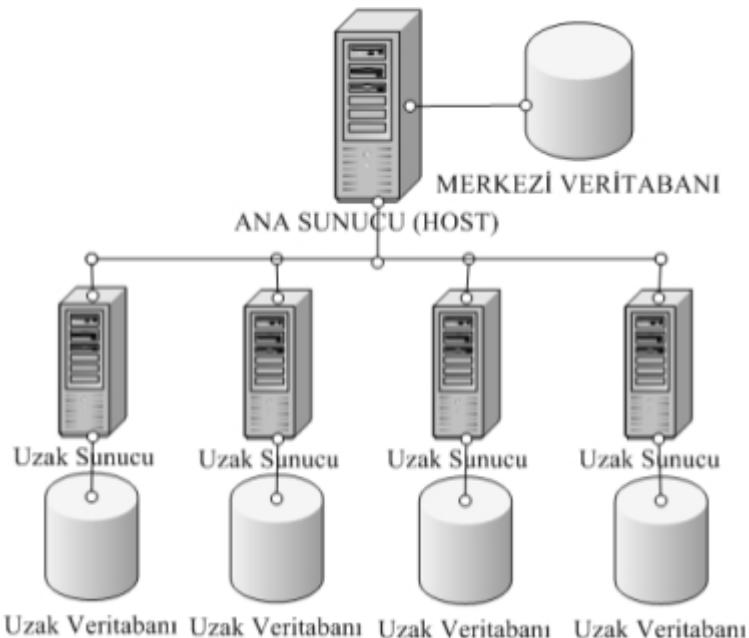
Şekil 18 Bir varlık-ilişki diyagramı

Bazı varlık-ilişki diyagramlarında varlıklar tablo (ilişki) olarak kullanılır. Böylece varlık ismi tablo ismiyle aynı olmaktadır. Bu nedenle birincil anahtarlar da diyagramda gösterilir. Bununla beraber bağlantınlarda ilişki türünü göstermek için farklı işaretler kullanılabilir [bkz. Laudon ve Laudon-2006].

Her ne kadar varlık-ilişki diyagramı ilişkisel modeller için kullanılıyor olsa da bazı araştırmacılar ilişkisel modelin dışında kalan veritabanı modellerine de söz konusu diyagramı uyarlamışlardır [bkz. Yeh ve ark.-2008].

10.3. Veritabanlarının Dağıtılması

Bir veritabanı ortamının oluşturulmasında verinin nasıl dağıtılaceği da tasarımcı tarafından dikkate alınmak zorundadır. Günümüzde artık birçok organizasyonun veritabanları, farklı ülkeler hatta kıtalarda elde edilen verileri saklamak mecburiyetinde kalabilmektedir. Böyle bir durumda veriye hızlı ve güvenli erişimi sağlamak için veritabanının dağıtılması söz konusu olur. Başka bir ifadeyle, merkezi bir sunucu üzerinde hizmet veren tek bir veritabanı yerine, birden fazla sunucu üzerinde hizmet veren, birden fazla veritabanı kullanılmaktadır. Ancak her ne kadar birden fazla veritabanı olsa da merkezi bir veritabanı olur ve diğerlerindeki güncellemeler zaman içinde merkezi veritabanına yansıtılır.



Şekil 19 Bir dağıtık veritabanı

Veri tabanlarının Şekil 19’da görüldüğü gibi dağıtılmışla oluşan tasarım *dağıtık (distributed) veritabanı* olarak adlandırılır. Hemen belirtmek gerekmektedir ki dağıtık veritabanları arasındaki uzaklık hatları veri iletim miktara bağlı olarak yüksek kaliteli olmalıdır. Genellikle öyle olsa da her zaman dağıtık veritabanlarının, çok uluslu organizasyonların büyük veritabanları gibi algılanmaması gerekmektedir. Zira *Artail ve ark. [2008]* cep bilgisayarları gibi mobil cihazlar üzerinde Bluetooth kablosuz ağ protokolüyle dağıtık veritabanı oluşturabilmişlerdir.

İki dağıtık veritabanı türü vardır. Bunlardan birincisi *bölümleme (partitioned) veritabanı* olarak bilinir. Bölümleme veritabanında, adından da anlaşılacağı gibi her bir uzak sunucuya bağlı veritabanı, kendi yerel ağında ihtiyaç duyulan verileri saklar. Başka bir veri bunlarda bulunmaz. Ancak genellikle akşam saatlerinde uzak sistemdeki tüm veriler merkezi veritabanına aktarılır. Kuşkusuz bu bir kopyalama biçiminde değildir. Her bir uzak veritabanındaki yalnızca güncellemeler yansıtılır. Diğer tür ise *çoğaltılmış (duplicate) veritabanı* olarak bilinir. Burada uzak veritabanlarının birbiriniyle haberleşmesi ve güncellemeleri birbirlerine aktarmaları söz konusudur. Özellikle herhangi bir uzak veritabanında yapılan güncellemenin, çok daha çabuk diğer uzak veritabanlarına yansıtılması istendiğinde tercih edilen bir türdür. Herhangi bir çalışma saatinde tüm güncellemeler merkezi veritabanına yansıtılır [bkz. Laudon ve Laudon-2006].

10.4. Veri Kalitesinin Sağlanması

Bir organizasyonda iyi bir veritabanı tasarımı maalesef ihtiyaç duyulan enformasyonun elde edilebilmesini tek başına sağlayamaz. İhtiyaç duyulan enformasyon bilindiği gibi veritabanındaki verilerden üretilecektir. Söz konusu veriler hatalı ve eksik olduğunda enformasyon da hatalı olacaktır. Özellikle geleneksel dosya yaklaşımıyla saklanan verilerde hata oranı veritabanı yaklaşımına kıyasla oldukça fazladır. Bunun en önemli nedeni, geleneksel dosya yaklaşımında aynı verinin birçok farklı dosyada tekrar etmesi ve böylece hata ihtimalinin artmasıdır. Oysa veritabanı yaklaşımı veri tekrarının en aza indirilmesini amaçlar. Buna rağmen veritabanı yaklaşımı da hatalı veriyi engelleyememektedir. Özellikle gelişen Web üzerinden tüketicilerin doğrudan veri girmesi sonucu hata ihtimali artmaktadır.

Hatalı verilerin engellenmesi veya düzeltilmesi için başvurulan bazı yöntemler vardır. Bunlardan ilki hatalı verilerin tespit edilmesi için kullanılan *veri kalitesi denetimi*dir (data quality audit). Bu yöntemde bazı istatistik tekniklerle seçilen örnek verilerin kalitesi denetlenir. Verinin kalitesini anlayabilmek için işe yönelik kurallar belirlenir. Örneğin “eğer hesap tipi borçsa, hesap süresi 30 günden fazlaysa ve hesap yekün dengesi sıfırdan büyükse o zaman hesap faiz ücreti sıfırdan büyük olmalıdır” gibi bir kural ile hesap verilerinin doğruluğu denetlenebilir.

Bir diğer yöntem ise hatalı verilerin yalnızca denetlenmesi değil düzeltmesini hedef alan *veri arıtmadır* (data cleansing veya data scrubbing). Ancak bu yöntem daha çok eksik veya fazla verilerin ve kelime hatalarının denetlenerek düzeltmesini amaçlar. En basitinden bir çalışan verileri inceleyerek hatalı girişleri denetleyebilir. Ancak bunun yerine çeşitli yazılımlar bu işi yerine getirmektedir. Bu tür yazılımlar daha çok kelime hatalarının düzeltmesi, birden fazla kopyası olan verilerin ve belli bir süre boyunca güncellenmeyen verilerin silinmesi gibi işleri yerine getirirler.

10.5. Veritabanı Yönetim Sistemlerinin Bileşenleri

Bir VTYS üç ana bileşenden meydana gelmektedir:

- Veri tanımlama dili,
- Veri işleme (manipulation) dili.
- Veri sözlüğü.

Veri tanımlama dili adından da anlaşıldığı gibi veritabanında saklanacak verilerin yapılarının tanımlanması için kullanılan bir dildir. Bu dil sayesinde veritabanı ve içinde yer alan tablolar belirlenir. Başka bir ifadeyle veritabanının adı, tabloların adı, tablolardaki alanlar (kolonlar), bu alanların özellikleri (sayısal veya karakter değer alması gibi) ve anahtarlar böyle bir dil yardımıyla tanımlanır.

Veri işleme dili ise bir VTYS' nin en sık kullandığı dildir. Bu dil yardımıyla yeni veri girişi, mevcut verilerin güncellenmesi veya silinmesi ve veriler üzerinde sorgulamalar yapılır. Başka bir ifadeyle veri tanımlama dili başta veritabanı yapısını tanımladıktan sonra veritabanının işletilmesi veri işleme diliyle gerçekleşir.

Günümüzde her iki dil için en sık kullanılan *yapısal sorgu dili*dir (structured query language veya SQL). Bu dil İngilizce konuşma diline benzer dördüncü nesil çok yüksek seviyeli bir dildir.

Günümüzde depolama cihazlarının maliyetlerinin düşmesi nedeniyle artık veritabanları oldukça fazla veri saklayabilmektedir. Bu nedenle özellikle fazla kullanıcısı olan veritabanları için sorgulama işlemleri süre olarak uzun sürmektedir. Böylece sorguların özet verilerde yapılabilmesi gündemde gelmiştir. Özet veri, kısaca özelliklerin değerlerinin gruplara ayrılmasıdır. Başka bir ifadeyle bir ilişkide *gelir* özelliği varsa ve değerleri para birimi cinsinden ise bunlar düşük, orta ve yüksek olarak özetlenebilir. Ancak bu tür durumlarda gruplama kıstasları kişiden kişiye değişebildiği gibi kullanılan terimlerin de farklı olması gerekebilir. *Ughetto ve ark.* [2008] bunun için her kullanıcının söz konusu terimleri kendisinin belirleyeceği bulanık mantık tekniğinden yararlanan bir çözüm önermişlerdir. Bulanık mantık tekniği özetle grupların eşik değerlerinin belirlenmesinde esneklik sağlamaktadır. Örneğin gelir seviyesi 750 birimin altında olanlar düşük grubunda yer alabilir. Ancak geliri 749 birim olan kişi düşük iken 750 birim olan orta olmaktadır. İşte bulanık mantık tekniği bu eşiği yumoşatmaktadır. Ayrıca çalışma, bahsi geçen düşük, orta veya yüksek gibi terimleri kullanıcının belirleyebilmesine izin vermektedir.

Veri sözlüğü, veri tanımlama diliyle yapılmış tanımların saklandığı bir dosyadır. Daha da ötesinde verilerle ilişkili tablolar, kullanıcılar ve programlar gibi daha detaylı bilgiler bu sözlükte mevcuttur. Böylece herhangi bir değişiklik yapılacağıda veritabanı yöneticisi bu sözlükten yararlanır. Bazen sözlük otomatik olarak düzenlemelerin yapılmasına yardımcı olabilmektedir. Örneğin beş karakterlik yer ayrılmış olan *Adı* alanı için 10 karakterlik yer yapılması gerektiği düşünülsün. Bu düzeltme veri tanımlama diliyle yapılabilir. Ancak düzeltme yalnızca veritabanını kapsamaktadır. Oysa veri sözlüğü yardımıyla bu veriyi kullanan tüm programlara da düzeltme yansıtılabilir [bkz. Gökçen-2005].

Kaynakça

Artail H., Shihab M., Safa H., "A distributed mobile database implementation on Pocket PC mobile devices communicating over Bluetooth", Journal of Network and Computer Applications, doi:10.1016/j.jnca.2008.04.003.

Gökçen H., Yönetim Bilgi Sistemleri, EPİ Yayıncıları, 2005.

Laudon K. C., Laudon J. P., Management Information Systems, Prentice Hall, Dokuzuncu baskısı, 2006.

Ughetto L., Voglozin W. A., Mouaddib N., "Database querying with personalized vocabulary using data summaries", Fuzzy Sets and Systems 159, 2030 – 2046, 2008.

Yarımaga Ü., Veri Tabanı Sistemleri, Akademi Yayıncıları, 2000.

Yeh D., Li Y., Chu W., "Extracting entity-relationship diagram from a table-based legacy database", The Journal of Systems and Software 81, 764–771, 2008.

Bölüm Özeti

Bu bölümde e-ticaret için çok önemli bir konu olan veritabanı yaklaşımına devam edilmiştir. Özellikle veritabanlarının kavramsal tasarımyla ilgili kavramlar ele alınmıştır. Öncelikle veritabanı tasarımında normalizasyon konusu ele alınmıştır. Daha sonra veritabanı kavramsal tasarımında çok ihtiyaç duyulan varlık-ilişki diyagramları incelenmiştir. Veritabanlarının ana bileşenleri ve birden fazla sunucuya dağıtılması konuları üzerinde durulmuştur.

11. VERİ TABANI YÖNETİM SİSTEMLERİ III

Birlikte Düşünelim

1. İlişkisel işlemler ve ilişkisel cebir nedir?
2. Web üzerindeki veritabanları nasıl kullanılır?
3. Çok boyutlu veri analizi nedir?
4. Veri madenciliği nedir?

Başlamadan Önce

Bir veritabanı altyapısını teorik olarak anlamayı sağlayan ilişkisel işlemler ve ilişkisel cebirdir. Bu sayede veritabanı üzerinde tasarım yapılabileceği gibi performans optimizasyonuna dayalı bazı modeller de geliştirilebilir. Daha ziyade kaynaklar da ileri düzey veritabanı konularına dahil edilse de esasen ortaokul matematik bilgisine dayandığından bu bölümde ele alınması uygun bulunmuştur. Ayrıca bu bölümde ilişkisel veritabanlarında sorgulamalar da incelenecektir. Günümüzde veritabanlarının en önemli olanları Web üzerinde yer almaktadır. Ayrıca E-ticaret denilince şüphesiz ki akla ilk olarak web üzerindeki veritabanları gelmektedir. Son yıllarda bilgisayar donanımındaki hızlı gelişmeyle birlikte klasik tablo biçiminde veri analizi yerini daha karmaşık küp şeklindeki analizlere bırakmaktadır. Kısaca OLAP olarak isimlendirilen bu yöntem de bu bölümün konuları arasındadır. Son olarak Büyük Veri konusuna giriş yapılarak büyük veri kütülelerinden anlamlı sonuçların nasıl çıkarıldığına degeinilecektir.

Giriş

İlişkisel bir VTYS' ni kullanabilmek için ilişkisel işlemleri yerine getirmek gerekmektedir. İlişkisel işlemler aslında VTYS bileşenlerinden veri işleme diliyle yapılabilen işlemlerin bir kısmını kapsamaktadır. Buna göre ilişkisel işlemler geçici tablolar oluştururlar. Bu tablolar gerçek tablo veya tabloların alt kümeleridir. Söz

konusu geçici tablolar ile kullanıcı bazı koşulları sağlayan kayıtlara yönelik raporlar veya bir varlığın yalnızca belirli özelliklerini içeren listeler oluşturabilir [Oz-2006].

Günümüzde Web' in sunduğu geniş erişim imkânı kuşkusuz ki veritabanı bağlantıları olmaksızın anlamsız olacaktır. Web o kadar gelişmiştir ki hemen her tür ürün çevrimiçi olarak satılmaktadır. Bunun ötesinde çoğu banka işlemi de Web üzerinden yapılmaktadır. Bir kullanıcı herhangi bir ürünü satın almadan önce ürünle ilgili stoklarda olup olmadığı, fiyatı, varsa teknik özellikleri, ağırlığı, rengi gibi bazı verileri öğrenmek ister. Tüm bu verilerin bulunduğu yer tabiatıyla veri tabanlarıdır. Banka işlemlerini yaparken kullanıcının hesap verilerine erişilmesi gereklidir. Bunlar da banka veri tabanındadır.

11.1. İlişkisel İşlemler ve İlişkisel Cebir

Bu amaçla kullanılan ilişkisel işlemler aslında ilişkisel cebir işlemlerinin kısaltmasıdır. İlişkisel cebir daha önce de dephinildiği gibi ilişkisel modelin temelidir. İlişkisel işlemleri açıklamadan önce bir ilişkinin matematiksel olarak nasıl temsil edildiğine dephinmek gerekmektedir. Buna göre bir ilişki bir kümeye karşılık gelmektedir. Her bir kayıt (satır) ise bir alt kümedir. Böylece Şekil 17' de gösterilmekte olan *PersEğitmen* ve *EğitmenEğitim* ilişkileri denklem' deki gibi temsil edilir.

$$PersEğitmen = \left\{ \begin{array}{l} \{\text{No : 15, Egitmen: Ayhan Sel}\} \\ \{\text{No : 11, Egitmen: Veli Nar}\} \\ \{\text{No : 15, Egitmen: Nazan Zengin}\} \\ \{\text{No : 11, Egitmen: Ayhan Sel}\} \\ \{\text{No : 17, Egitmen: Cengiz Seymen}\} \end{array} \right\}$$

$$EğitmenEğitim = \left\{ \begin{array}{l} \{\text{Egitmen: Ayhan Sel, Egitim: NLP}\} \\ \{\text{Egitmen: Veli Nar, Egitim: Risk Yönetimi}\} \\ \{\text{Egitmen: Nazan Zengin, Egitim: Risk Yönetimi}\} \\ \{\text{Egitmen: Cengiz Seymen, Egitim: NLP}\} \end{array} \right\}_1$$

Cok daha fazla ilişkisel cebir işlemi olsa da hepsi bu çalışmanın kapsamına girmediğinden yalnızca dördüne dephinilecektir [bkz. Connoly-1999 ve Yarimağan-2000].

11.1.1. Seçme (Selection) İşlemi

Seçme işlemi bir kümedeki alt-kümelerden (satır), belli bir koşulu sağlayanlarının seçilerek yeni bir kümeye oluşturulması için kullanılır. Denklem 2' deki notasyona göre *ilişki adı* ile belirlenmiş kümeden *koşulu* sağlayan alt-kümeler seçilecektir.

$$\sigma_{\text{kosul}} (\text{ilişki adı}) \quad 2$$

Buna göre,

$$R = \sigma_{\text{egitmen} \neq \text{AyhanSel}} (PersEğitmen) \Rightarrow R = \{\{\text{15, AyhanSel}\}, \{\text{11, AyhanSel}\}\}_3$$

olur.

11.1.2. İzdüşüm (Projection) İşlemi

Bu işlem bir kümedeki alt-kümelerin hepsinin ve alt-kümelerdeki belirli elemanların yeni bir kümeye oluşturulmasını sağlar. Bir başka değişle bir ilişkinin belirli özelliklerinin değerleri listelenmektedir. Denklem 4' teki notasyona göre *ilişki adı* ilişkisinden *özellik listesi*nde yer alan özelliklerin değeri listelenir. Birden fazla özellik ismi virgülle ayrılarak listelenir.

$$\pi_{\text{ozelliklistesi}} (\text{ilişkiadı}) \quad 4$$

Buna göre,

$$R = \pi_{\text{Egitmen}}(\text{PersEgitmen}) \Rightarrow R = \left\{ \begin{array}{l} \text{Ayhan Sel}, \\ \text{Veli Nar}, \\ \text{Nazan Zengin}, \\ \text{Ayhan Sel}, \\ \text{Cengiz Seymen} \end{array} \right\}$$

5

olur.

11.1.3. Kartezyen Çarpım (Cartesian Product) İşlemi

Birden fazla küme üzerinde yapılan bir işlemidir. Bu anlamda diğer iki işlemden farklıdır. Bu işlem küme çarpımı olarak da bilinmektedir. Çarpmaya dahil olan tüm kümelerin her bir alt-kümesiyle diğerinin her bir alt-kümesi bir araya gelerek yeni kümenin alt-kümelerini oluşturur. Böylece birden fazla ilişkiden tek bir ilişki oluşturmak mümkün olmaktadır. Denklem 6' daki notasyona göre n adet ilişkinin çarpımı R ilişkisini oluşturmaktadır ve R üzerinde diğer işlemler yapılabilir.

$$R = R_1 \times R_2 \times \dots \times R_n$$

6

Buna göre,

$$R^+ = \text{PersEgitmen} \times \text{EgitmenEgitmen}$$

↓

$$R^+ = \left\{ \begin{array}{l} \{(15, \text{Ayhan Sel}), (\text{Ayhan Sel}, \text{NLP})\}, \\ \{(15, \text{Ayhan Sel}), (\text{Veli Nar}, \text{Risk Yönetimi})\}, \\ \{(15, \text{Ayhan Sel}), (\text{Nazan Zengin}, \text{Risk Yönetimi})\}, \\ \{(15, \text{Ayhan Sel}), (\text{Cengiz Seymen}, \text{NLP})\}, \\ \{(11, \text{Veli Nar}), (\text{Ayhan Sel}, \text{NLP})\}, \\ \vdots \\ \{(17, \text{Cengiz Seymen}), (\text{Ayhan Sel}, \text{NLP})\}, \\ \{(17, \text{Cengiz Seymen}), (\text{Veli Nar}, \text{Risk Yönetimi})\}, \\ \{(17, \text{Cengiz Seymen}), (\text{Nazan Zengin}, \text{Risk Yönetimi})\}, \\ \{(17, \text{Cengiz Seymen}), (\text{Cengiz Seymen}, \text{NLP})\} \end{array} \right\}$$

7

olur. Dikkat edilirse farklı kümelerden gelen alt-kümeler birleşerek yeni alt kümeler olmaktadır. Ayırt edilebilmesi için farklı kümelerden gelen alt-kümeler parantezler içine alınmıştır. Bununla birlikte,

$$R = \pi_{\text{no, PersEgitmen, Egitmen, Egitmen}}(\sigma_{\text{PersEgitmen} \times \text{EgitmenEgitmen}}(R^+))$$

↓

$$R = \left\{ \begin{array}{l} \{15, \text{Ayhan Sel}, \text{NLP}\}, \\ \{11, \text{Veli Nar}, \text{Risk Yönetimi}\}, \\ \{15, \text{Nazan Zengin}, \text{Risk Yönetimi}\}, \\ \{11, \text{Ayhan Sel}, \text{NLP}\}, \\ \{17, \text{Cengiz Seymen}, \text{NLP}\} \end{array} \right\}$$

8

olur. Burada nokta “.” işlevi “in” anlamında kullanılmıştır. Zira hem PersEgitmen hem de EgitmenEgitmen ilişkilerinde aynı isimli özellik olduğundan (Egitim) hangisinin kastedildiği önem kazanmaktadır. Buna göre $R.x$ gibi bir notasyonun anlamı “ R ilişkisinin x özelliği”dir.

11.1.4. Birleştirme (Join) İşlemi

Birden fazla ilişkinin bir araya getirilmesiyle birleşme ilişki (veya tablo) elde edildiğine daha önce de感恩しました. Bu işlemin yapılması kartezyen çarpımıyla giriş yapılmıştır. Buna göre kartezyen çarpımı özetle en az iki farklı ilişkideki kayıtların her birini diğeryle yan yana getirmektedir. Denklem 7' de bu durum gösterilmiştir. Ancak bu birleşmeyle elde edilen ilişki çoğu zaman anlamsızdır. Çünkü her kayıt diğeryle eşleşmektedir. Bunu engellemek için ortak özelliklerin eşit olduğu kayıtlar denklem 8' de görüldüğü gibi seçilmiştir. Yapılan bu işlemin daha kısa bir notasyonla gösterilmesi denklem 9' da sağlanmaktadır. Burada yer alan θ , “=”, “<”, “>”, “≤” veya “≥” işaretlerinden birisini temsil etmektedir ki çok büyük ekseriyetle “=” olmaktadır.

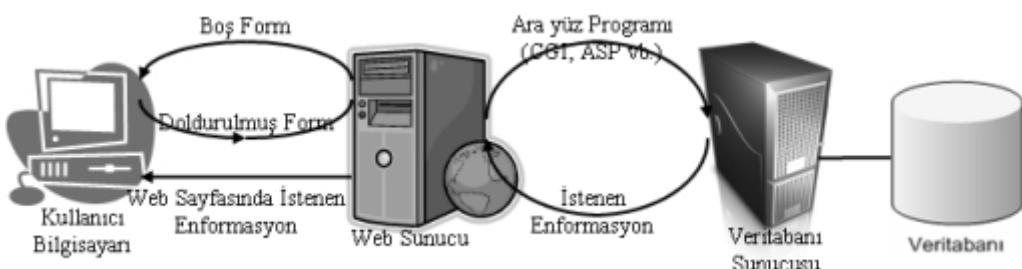
$$R_1 \text{ ilişkisi} \bowtie R_2 \text{ ilişkisi} = \sigma_{R_1 \text{inocelligi} \theta R_2 \text{inocelligi}} (R_1 \text{ ilişkisi} \times R_2 \text{ ilişkisi})$$

9

11.2. Web Üzerindeki Veritabanları

Günümüzde Web' in sunduğu geniş erişim imkânı kuşkusuz ki veri tabanı bağlantıları olmaksızın anlamsız olacaktır. Web o kadar gelişmiştir ki hemen her tür ürün çevrimiçi olarak satılmaktadır. Bunun ötesinde çoğu banka işlemi de Web üzerinden yapılmaktadır. Bir kullanıcı herhangi bir ürünü satın almadan önce ürünle ilgili stoklarda olup olmadığı, fiyatı, varsa teknik özellikleri, ağırlığı, rengi gibi bazı verileri öğrenmek ister. Tüm bu verilerin bulunduğu yer tabiatıyla veritabanlarıdır. Banka işlemlerini yaparken kullanıcının hesap verilerine erişilmesi gereklidir. Bunlar da banka veri tabanındadır.

Veri tabanlarının Web erişimine açık olmasının getirdiği önemli faydaların yanı sıra bazı riskler de vardır. Bilindiği gibi veri tabanları işletmeler için hayatı öneme sahiptir. Bunlar önemle korunmalıdır. Dışarıdan veri çalınması veya mevcut verilerin bozulmasına kesinlikle izin verilmemelidir. Bu sebeple de Web kullanıcılarının doğrudan veri tabanı sunucusuna bağlanarak işlemlerini gerçekleştirmelerine izin verilmelidir. Böylece ortak geçit ara yüzü (common gateway interface veya CGI), Java servlet, aktif sunucu sayfaları (active server pages veya ASP) ve PHP[4] gibi bazı ara yüz programlarıyla veri tabanına erişilir. Bu tür ara yüz programları Şekil 20' de görüldüğü gibi Web sunucusu üzerinde çalışırlar [Oz-2006]. Buna göre Web sunucusu boş bir formu kullanıcıya gönderir ve isteklerini bu formda belirtmesini ister. Kullanıcıdan alınan forma göre veritabanı üzerinde gerekli işlemleri yapar. Elde edilen sonuçlar Web sunucusunda Web sayfası biçiminde hazırlanarak tekrar kullanıcıya gönderilir. Böylece kullanıcı asla doğrudan veri tabanı ile bağlantı kurmamaktadır.

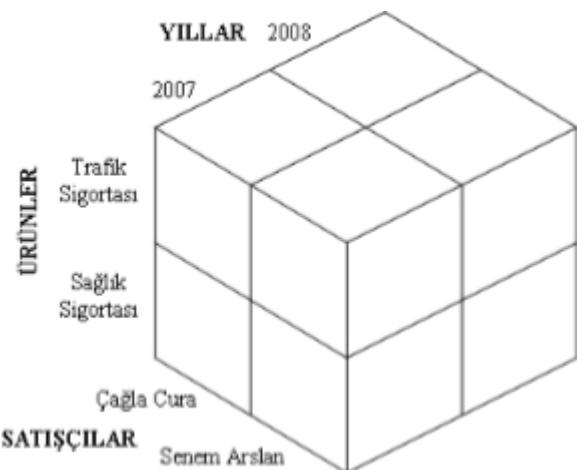


Şekil 20 Web üzerinden veritabanı bağlantısı

11.3. Çok Boyutlu Veri Analizi

Bazen yöneticiler öyle veri analizlerine ihtiyaç duyarlar ki buraya kadar anlatılmış alışlageldik VTYS yaklaşımlarıyla bunların karşılanması mümkün olmaz. Çok boyutlu veri analizi de bunlardan birisidir. Bu analiz türü adından da anlaşılacağı gibi birçok açıdan verilerin karşılaştırılmasına dayalıdır. Örneğin satırlarının ve kolonlarının isimlendirilmiş olduğu bir tablo düşünülsün. Kolon isimleri satışçılar, satır isimleri ise ürünler olsun. Bu durumda hücrelerde de satış miktarları olacaktır. Böylece herhangi bir hücredeki satış miktarı belirli bir satışçının, belirli bir üründen ne kadar sattığını gösterecektir. Bilindiği gibi buraya kadar tartışılan VTYS yaklaşımları veriyi böyle bir biçimde saklamamaktadır. Boyut sayıları

arttırılabilir. Örneğin yıllara göre ürün satışları veya yıllara göre satışçıların yaptığı satışları da incelemek gerekebilir.



Şekil 21 Çok boyutlu veri modeli

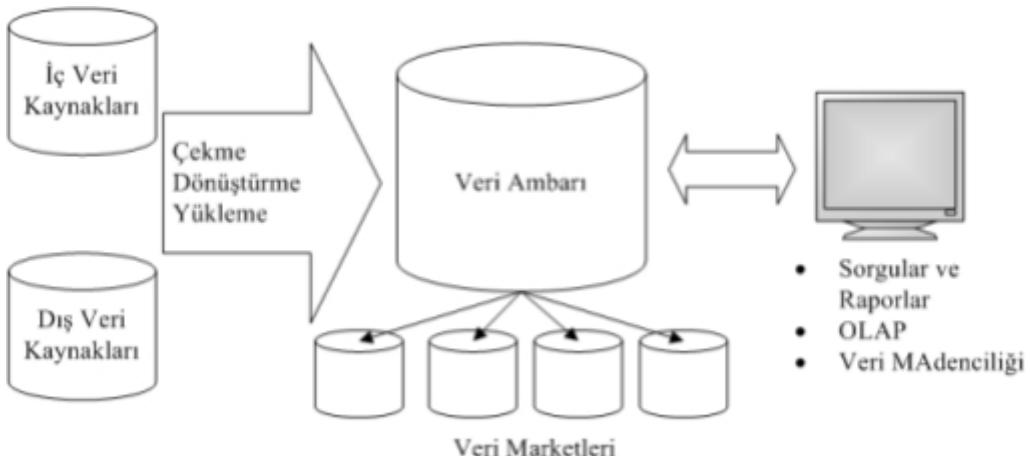
Çok boyutlu veri modeline yukarıda verilen örnek, Şekil 21’ de görülmektedir [Laudon ve Laudon-2006]. Görüldüğü gibi küp şeklindeki bir yapı söz konusudur. Bunlar gibi birçoğu karmaşık analizler yapmak için büyük bir küpün içine yerleştirilebilir. Ayrıca bir küpten, döndürülen yön'e göre farklı enformasyon elde edilebilmektedir. Örneğin Şekil 21’ deki küp 90 derece döndürüldüğünde yıllara göre ürün satışları, 180 derece döndürüldüğünde ise yıllara göre satışçıların yaptığı satışlar analiz edilebilir. Kuşkusuz bilinen VTYS yaklaşımları bu biçimde veriyi saklamadıklarından, ya enformasyonu bu biçimde sunacak bir araç veya çok boyutlu özel bir VTYS kullanılmalıdır. Çok boyutlu veri analizi, *çevrimiçi analitik işleme* (online analytical processing veya OLAP) olarak ta adlandırılmaktadır.

11.4. Veri Madenciliği

Veri madenciliğini açıklamak için öncelikle veri ambarlarına değinmek gerekir. Karar vericiler mevcut işlerle, eğilimlerle ve değişimlerle ilgili kısa, öz ve güvenilir enformasyona ihtiyaç duyarlar. Bu enformasyon sıkılıkla mevcut veri tabanından sağlanamaz. Bir veri tabanında saklanan veriler daha günceldir. Bunların daha eskiye dayananları erişimi daha yavaş, fakat daha ucuz olan depolama aygıtlarında saklanır. Başka bir ifadeyle, mevcut veri tabanında eskiyen bilgiler farklı ortamlara yedeklenir. Daha da ötesi bunlardan iyice eskimiş olanlar tamamen silinebilir. Oysa günümüz işletmelerinde sağlıklı analizlerle karar vericiye yardımcı olmak için ihtiyaç duyulan veriler oldukça eskiye dayanmaktadır. Bu nedenle analizde kullanılacak veri miktarı da ciddi ölçüde artmaktadır. Örneğin bir günde satın alma, satma ve ödeme gibi defalarca gerçekleşen işlemler daha çabuk yedeklenir. Böylece, bir işletme mümkün olduğunda fazla satış kaydını inceleyerek analiz yapmak istediğiinde bazı sıkıntılar yaşayacaktır. Çünkü yedeklenmiş verinin geri çağrıması maliyetli bir süreçtir. Daha kötüsü verinin bir kısmı tamamen silinmiş ve analiz yapmak için mevcut veriler yetersiz olabilir. Bu sebeple analiz amaçlı kullanılacak verilerin ya aynen ya da özetlenmiş olarak saklandığı büyük veri tabanları, *veri ambarı* olarak adlandırılır. Bununla birlikte bir *veri marketi* (data mart), bir veri ambarının alt-kümeleridir. Bunlar belirli bir grup kullanıcı için özetlenmiş ve üzerine odaklanmış bir kısım veriyi barındırırlar. Örneğin yukarıda verilen örnekte olduğu gibi yalnızca satışlara ilişkin veriler veri marketinde saklanır [bkz. Oz-2006].

Bir veri ambarının oluşturulması Şekil 22’ de görüldüğü gibi üç aşamadan meydana gelmektedir. Bunlar, işletmenin kendi veri tabanından ya da dışarıdan satın alma gibi çeşitli yöntemlerle elde ettiği verilerin bir biçimde veri ambarı içerisinde yer almasını sağlar. Böylece söz konusu aşamalar veri çekme, veri dönüştürme ve veri yükleme olarak isimlendirilir. *Veri çekme* aşamasında veri tabanındaki verilerden gerekli olanlar alınıp veri ambarına yerleştirilmek üzere dosyalanırlar. Bir veri ambarında verilerin standart olması, çeşitli kelime hatalarının elenmesi gerekmektedir. Bu sebeple *dönüştürme* aşamasında çekilmiş veriler arıtılır [bkz. Ayrı 10.4]. Veri standartlaşmasına örnek olarak “Türkiye Cumhuriyeti” için “TC”, “T.C.” veya “Türkiye Cumhuriyeti” değerlerinin üçü de karışık olarak girilmiş olduğu durum düşünülsün. Bu durumda tüm veriler yalnızca birine dönüştürülmeli dir. Böylece doğru analizler yapılabilir. Son olarak *yükleme*

aşamasında, dönüştürülmüş veriler veri ambarına aktarılır ve hata olup olmadığını denetlemek için veri tabanındaki kilerle karşılaştırılırlar.



Şekil 22 Bir veri ambarının bileşenleri

Veri ambarları analizler yapabilmek için çeşitli araçlar sunarlar. Bunlar analitik, sorgulama ve grafik raporlama gibi araçlar olmakla birlikte OLAP ve veri madenciliği araçlarını da barındırırlar. *Veri madenciliği* büyük bir veri havuzunda çeşitli teknikleri kullanarak saklı örüntülerini ve ilişkileri ortaya çıkarır ve geleceğe yönelik tahminler yapabilmek için bunlardan bazı kurallar üretir [Fayyad ve ark.-2002 ile Hirji-2001]. Örneğin geçmişte belli bir ürünü satın alan tüketicilerin aynı satın alma işleminde başka hangi ürünleri satın aldıkları analiz edilebilir. Böylece herhangi iki ürünün satışları arasında ilişki kurulabilir ve geleceğe yönelik pazarlama stratejileri belirlenebilir. Benzer biçimde tüketicilerin özellikleri ile satın aldıkları ürünler arasında ilişki olup olmadığı analiz edilerek, belirli özellikteki bir müşterinin hangi ürünün tüketicisi olabileceği tahmin edilebilir [bkz. Laudon ve Laudon-2006].

Bölüm Özeti

Bu bölümde e-ticaret için çok önemli bir konu olan veritabanı yaklaşımının ileri düzey konuları ele alınmıştır. Öncelikle ilişkisel veritabanlarını teorik olarak anlamayı sağlayan ilişkisel işlemler ve sorgulamaları teorik olarak anlamayı sağlayan ilişkisel cebir incelenmiştir. Daha sonra günümüzde önemli bir konu olan Web üzerindeki veri tabanları ele alınmıştır. Son olarak çok önemli analizler yapılmasını sağlayan çok boyutlu veri analizi ve veri madenciliği tartışılmıştır.

Kaynakça

Connolly T., Begg C., Strachan A., Database Systems: A practical approach to design, implementation and management, Addison Wesley, İkinci baskı, 1999.

Fayyad U., Ramakrishnan R., Srikant R., "Evolving Data Mining into Solutions for Insights", Communications of the ACM 45, 2002.

Hirji K. K., "Exploring Data Mining Implementation", Communications of the ACM 44, 2001.

Laudon K. C., Laudon J. P., Management Information Systems, Prentice Hall, Dokuzuncu baskı, 2006.

Oz E., Management Information Systems, Thomson, Beşinci baskı, 2006.

Yarimağan Ü., Veri Tabanı Sistemleri, Akademi Yayınları, 2000.

12. JSF VE WEB UYGULAMALARI

Birlikte Düşünelim

1. JSF Ne amaçla kullanılır?
2. JSF kurulumu için temel gereksinimler nelerdir?
3. JSF mimarisini kısaca anlatınız.
4. JSF yaşam döngüsü hangi adımlardan oluşur?

Başlamadan Önce

Web uygulama geliştirme e-ticaret içerisinde bilinmesi gereken konulardandır. Her zaman bir yöneticinin uzman bir yazılımcı olması şüphesi ki beklenemez. Fakat bir e-ticaret işletmesini yönten kişi konu hakkında yeterli bilgiye de sahip olmalıdır. Bu nedenle web uygulama geliştirmede sıkça kullanılan JSF konusu hakkında bu bölümde bilgi verilecektir. Öncelikle JSF' in kullanım amaçları ve katkıları konusunda genel bir inceleme yapılacaktır. Ardından JSF' in uygulanabilmesi için örnek olarak bir bilgisayara kurulum adımları gösterilecektir. Son olarak bu bölümde JSF' in temel mimarisi anlaşılacaktır. Böylece JSF' in çalışma prensibi hakkında gerekli bilgi sağlanmış olacaktır.

Giriş

Java Server Faces (JSF), web tabanlı kullanıcı arayüzlerinin geliştirme entegrasyonunu basitleştirmeyi amaçlayan Java tabanlı bir web uygulama iskeletidir. Java Server Faces, Java Topluluğu Süreci ile bir spesifikasyonda resmîleştirilmiş standart bir ekran teknolojisidir.

Bu bölüm size temel JSF kavramlarını öğretecek ve aynı zamanda JSF iskeleti ile ilgili çeşitli ileri kavramlar boyunca sizi yönlendirecektir.

Bu bölüm, yeni başlayanların temel JSF programlamasını anlamalarına yardımcı olacak şekilde hazırlanmıştır. Bu dersler tamamladıktan sonra, kendinizi bir sonraki seviyelere götürebileceğiniz JSF programlamada orta derecede bir uzmanlığa yaklaşmış olacaksınız.

12.1. JSF Nedir?

Java Server Faces (JSF), bir sayfada yeniden kullanılabilir Kullanıcı Arabirimleri (UI) bileşenlerini kullanan sunucu tabanlı uygulamalar için UI yapımını kolaylaştıran bir MVC (Model Görüntü Kontrolör) web iskeletidir. JSF, kullanıcı arabirimini bileşenlerini veri kaynaklarına ve sunucu tarafı olay işleyicisine bağlamak için bir olanak sağlar. JSF spesifikasyonu bir dizi standart UI bileşeni tanımlar ve bileşen geliştirmek için bir Uygulama Programlama Arayüzü (API) sağlar. JSF, mevcut standart UI bileşenlerinin yeniden kullanılmasını ve genişletilmesine izin verir.

JSF, bir Java uygulama sunucusunda çalışacak ve uygulama kullanıcı arabirimini hedef bir istemciye aktaracak uygulamalar oluşturma ve sürdürme çabalarını azaltır. JSF, aşağıdakileri sayesinde Web uygulaması geliştirmeyi kolaylaştırır.

- Yeniden kullanılabilir UI bileşenleri sağlama
- UI bileşenleri arasında kolay veri aktarımı yapmak
- Birden çok sunucu isteğinde UI durumunu yönetme
- Özel bileşenlerin uygulanmasını sağlama
- Sunucu tarafı uygulama koduna istemci tarafı olayı gönderme

JSF, geliştiricilere, birden fazla istemci türü için (örneğin- HTML tarayıcısı, kablosuz veya WAP cihazı) farklı şekillerde işleyebilecek UI bileşenleri koleksiyonlarından Web uygulaması oluşturma olanağı sağlar.

JSF aşağıdakileri sağlar:

- Çekirdek kütüphane.
- Temel kullanıcı arabirimini bileşenleri kümesi- standart HTML giriş öğeleri
- Temel kullanıcı arabirimini bileşenlerinin ek kullanıcı arabirimini bileşen kitaplıklarını oluşturmak veya mevcut bileşenleri zenginleştirmek için genişletilmesi
- İstemci türlerine bağlı olarak JSF UI bileşenlerinin kendilerini farklı hale getirmelerini sağlayan çoklu oluşturma yetenekleri

12.2. JSF Kurulumu

Bu bölüm size JSF Framework ile çalışmaya başlamak için bir geliştirme ortamı hazırlama konusunda rehberlik edecektir. JSF Framework kurulumunu yapmadan önce JDK, Eclipse, Maven ve Tomcat makinelerini nasıl kuracağınızı öğreneceksiniz.

JSF, JDK 1.5 veya daha üst bir sürümü gerektirir, bu nedenle ilk gereksinim, makinenizde JDK yüklü olmasıdır. JSF uygulama geliştirmeye başlamak üzere ortamınızı ayarlamak için aşağıda verilen adımları takip edilmesi gerekmektedir.

Adım 1: Makinenizde Java kurulumunu kontrol edin.

Konsolu açın ve aşağıdaki Java komutunu yürütün.

Tablo 5 Java kurulumunu doğrulama

| İşletim Sistemi | Görev | Komut | Üretilen Çıktı |
|------------------------|--------------------------------|-----------------------------------|---|
| Windows | Komut konsolunu açın (cmd.exe) | c:\> java -version | java version "1.8.0_131"

Java(TM) SE Runtime Environment (build 1.8.0_131-b11)

Java HotSpot(TM) Client VM (build 25.131-b11, mixed mode, sharing)
java version "1.8.0_131" |
| Linux | Komut terminalini açın | \$ java -version | Java(TM) SE Runtime Environment (build 1.8.0_131-b11)

Java HotSpot(TM) Client VM (build 25.131-b11, mixed mode, sharing)
java version "1.8.0_131" |
| Mac | Terminali açın | machine:~ tunchan\$ java -version | Java(TM) SE Runtime Environment (build 1.8.0_131-b11)

Java HotSpot(TM)64-Bit Server VM (build 25.131-b11, mixed mode, sharing) |

Adım 2: Java Geliştirme Setini (JDK) Ayarlayın.

Eğer Java kurulu değilse, Java Yazılım Geliştirme Kitini (SDK) Oracle'in Java sitesinde Java SE İndirmeleri kısmından yükleyebilirsiniz. JDK yükleme talimatlarını sitede mevcuttur, kurulumu yapmak ve yapılandırmak için verilen talimatları izleyin. Son olarak, PATH ve JAVA_HOME ortam değişkenlerini, sırasıyla java_install_dir/bin ve java_install_dir olmak üzere java ve javac içeren dizinleri gösterecek biçimde ayarlayın.

Tablo 6 JAVA_HOME ortam değişkeninin ayarlanması

İşletim Sistemi Çıktı

| | |
|---------|---|
| Windows | JAVA_HOME ortam değişkenini C:\Program Files\Java\jdk1.8.0_131 olarak ayarlayın |
| Linux | Export JAVA_HOME=/usr/local/java-current |
| Mac | Export JAVA_HOME=/Library/Java/Home |

Adım 3: Eclipse IDE Ayarlayın.

Bu derslerdeki tüm örnekler Eclipse IDE kullanılarak verilmiştir. Bu nedenle, işletim sisteminize bağlı olarak, makinenize Eclipse'in en son sürümünün yüklü olması önerilir.

Eclipse IDE'yi kurmak için, <http://www.eclipse.org/downloads/> adresinden WTP destekli en son Eclipse ikili dosyalarını indirin. Kurulumu indirdikten sonra, ikili dağıtımları uygun bir yere paketinden çıkarın. Örneğin, Windows'ta C:\eclipse veya Linux - Unix'te /usr/local/eclipse içinde ve sonunda PATH değişkenini uygun şekilde ayarlayın.

Not: Aşağıdaki eclipse yazılımı güncelleme sitesini kullanarak m2eclipse eklentisini eclipse' e yükleyin.

m2eclipse eklentisi: <http://m2eclipse.sonatype.org/update/>

Bu eklenti, geliştiricilerin, eclipse içindeki maven komutlarını çalıştırmasını sağlar.

Maven 3.6.1'i (son sürüm) <http://maven.apache.org/download.html> adresinden indirin. Arşivi, kurmak istediğiniz dizine çıkarın. Apache-maven-3.6.1 alt dizini arşivden oluşturulacaktır.

Ortam değişkenlerine M2_HOME, M2, MAVEN_OPTS eklenir.

Tablo 7 Maven için ortam değişkenlerinin düzenlenmesi

İşletim Sistemi Yapılacak İşlem

Sistem Değişkenlerine aşağıdakiler eklenir

| | |
|-------------|---|
| Windows | M2_HOME=C:\Program Files\Apache Software Foundation\apache-maven-3.6.1
M2=%M2_HOME%\bin

MAVEN_OPTS=-Xms256m -Xmx512m
Komut terminali açılır ve ortam değişkenleri ayarlanır. |
| Linux / Mac | export M2_HOME=/usr/local/apache-maven/apache-maven-3.6.1
export M2=%M2_HOME%\bin

export MAVEN_OPTS=-Xms256m -Xmx512m |

Daha sonra M2 değişkeni Sistem Yolu'na (System Path) eklenir. Bunun için Windows işletim sisteminde %M2% dizgisi Path değişkeninin sonuna eklenir. Linux veya Mac işletim sistemlerinde ise *export PATH = \$M2:\$PATH* komutu çalıştırılır.

Adım 4: Apache Tomcat'i kurun.

Tomcat'ın en yeni sürümünü <http://tomcat.apache.org/> adresinden indirebilirsiniz. Kurulumu indirdikten sonra, ikili (binary) dağıtımları uygun bir yere paketinden çıkarın. Örneğin, Windows'ta C:\apache-tomcat-9.0.21'de veya Linux - Unix'te /usr/local/apache-tomcat-9.0.21'de kurulum konumlarına işaret eden CATALINA_HOME ortam değişkenini ayarlayın.

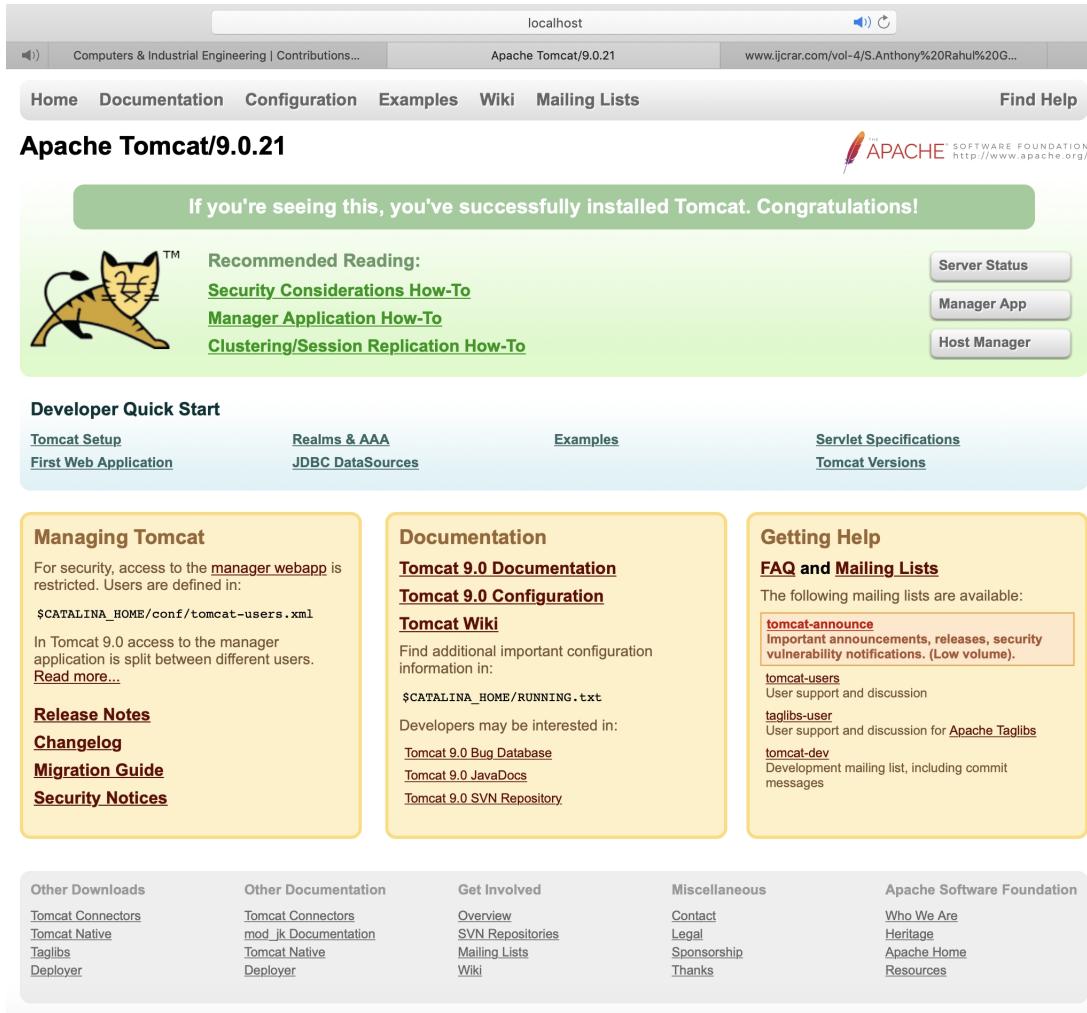
Tomcat, Windows makinesinde aşağıdaki komutları çalıştırarak başlatılabilir veya startup.bat dosyasına çift tıklayabilirsiniz.

%CATALINA_HOME%\bin\startup.bat veya yukarıda gösterildiği gibi çıkartıldığa C:\apache-tomcat-9.0.21\bin\startup.bat

Tomcat, Unix (Solaris, Linux, vb.) makinelerde ise startup.sh komutu çalıştırarak başlatılabilir.

%CATALINA_HOME%\bin\startup.sh veya usr/local/apache-tomcat-9.0.21/bin/startup.sh

Başarılı bir başlangıçtan sonra, Tomcat ile birlikte gelen varsayılan web uygulamaları http://localhost:8080/ adresini ziyaret ederek kullanılabilir. Her şey yolundaysa, aşağıdaki gibi bir sonuç gösterecektir.



Şekil 23 Başarılı Tomcat kurulumu.

Tomcat'ı yapılandırması ve çalıştırması hakkında daha fazla bilgi, Tomcat web sitesinde bulunabilir: <http://tomcat.apache.org>.

Tomcat, bir Windows bilgisayarda aşağıdaki komutları çalıştırarak durdurulabilir:

%CATALINA_HOME%\bin\shutdown.bat veya C:\apache-tomcat-9.0.21\bin\shutdown.bat

Tomcat, Unix tabanlı (Solaris, Linux, vb.) makinelerde aşağıdaki komutları çalıştırarak durdurulabilir:

%CATALINA_HOME%\bin\shutdown.sh veya usr/local/apache-tomcat-9.0.21/bin/shutdown.sh

12.3. JSF Mimarisi

JSF teknolojisi, sunucu tarafı Kullanıcı Arabirimini Bileşenleri geliştirmek, oluşturmak ve bunları bir web uygulamasında kullanmak için bir iskelettir. JSF teknolojisi, algoritma mantığını sunumdan ayırmak için Model Görünüm Kontrolörü (MVC) mimarisine dayanmaktadır.

MVC tasarım yapısı, üç ayrı modül kullanarak bir uygulama tasarlamayı sağlar:

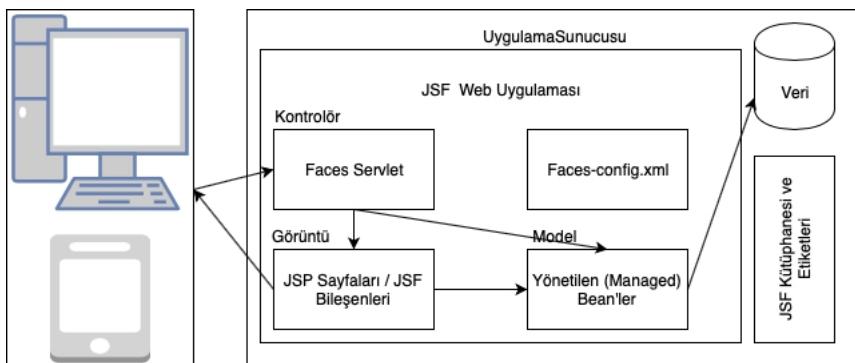
- Model, proje içerisinde kullanılacak olan nesnelerin oluşturulduğu kısımdır. Günlük hayatın somut nesnelerin, bilgisayar ortamında modellenmesi anlamına gelir. Örneğin bir kütüphane otomasyonu yapmak istediğimizde kitap bilgilerinin tutulacağı bir modele ihtiyaç duyulacaktır. Bu modelde kitabı seri numarası, adı, yazarı, yayinevi gibi bilgileri yer alacaktır.
- Görünüm (View), proje tamamlandığında kullanıcının gördüğü ara yüzüdür. Bu bir web sayfası, masaüstü uygulaması ara yüzü veya mobil bir tasarım olabilir. Projenin yapısına göre bu tasarım farklı şekillerde oluşturulabilir.
- Kontrolör (Controller), projedeki tüm işlemlerin (veri tabanı işlemleri, hesaplamalar, veri aktarımı vb) yapıldığı kontrol bölümündür. Controller ayrıca model ve görünüm arasındaki veri akışını da kontrol eder.

MVC tasarım modelinin amacı, geliştiricilerin temel becerilerine odaklanmalarını ve daha net bir şekilde iş birliği yapmalarını sağlayan model ve sunumları ayırmaktır.

Web tasarımcılarının model ve denetleyici katmanından ziyade yalnızca görünüm katmanına odaklanması gereklidir. Geliştiriciler model kodunu değiştirebilir ve genellikle görünüm katmanını değiştirmeleri gerekmeyez. Denetleyiciler, kullanıcı eylemlerini işlemek için kullanılır. Bu süreçte, katman modeli ve görüntüler değiştirilebilir.

JSF uygulaması, diğer Java teknolojisi tabanlı web uygulamalarına benzer; bir Java servlet teknolojisi altında çalışır ve aşağıdakileri içerir:

- JavaBeans, uygulamaya özgü işlevsellik ve veri içeren modeller olarak kullanılan bileşenler.
- Olay işleyicileri ve doğrulayıcıları temsil etmek için özel bir etiket kitaplığı.
- UI (Kullanıcı Arayüzü) bileşenlerini oluşturmak için özel bir etiket kitaplığı.
- UI bileşenleri, sunucuda durum bilgisi nesneler olarak temsil edilir.
- Sunucu tarafı bazı yardımcı sınıfları.
- Hata kontrolörleri, olay işleyicileri ve gezinti işleyicileri.
- Uygulama kaynaklarını yapılandırmak için uygulama yapılandırma kaynak dosyası.



Şekil 24 JSF Mimarisi

Kullanıcı eylemlerini gerçekleştirmek için kullanılabilen kontrolörler vardır. UI, web sayfası yazarları tarafından oluşturulabilir ve iş mantığı yönetilen bean (nesne) tarafından kullanılabilir.

JSF, tek bir bileşeni oluşturmak için çeşitli mekanizmalar sağlar. İstenilen temsili seçmek web sayfası tasarımcısına bağlıdır ve uygulama geliştiricisinin bir JSF UI bileşenini oluşturmak için hangi mekanizmanın kullanıldığını bilmesi gerekmeyez.

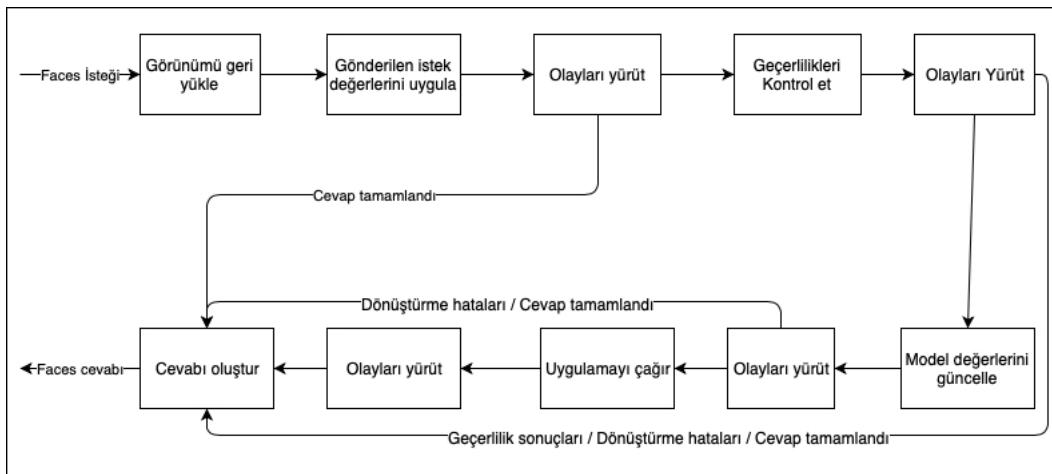
12.4. JSF Yaşam Döngüsü

JSF uygulama yaşam döngüsü aşağıdaki gibi altı aşamadan oluşur:

- Görünümün geri yüklenmesi aşaması.
- İstek değerlerinin alınması aşaması; olaylar yürütülür.

- Süreç doğrulamaları aşaması; olaylar yürütülür.
- Model değerlerinin güncellenmesi aşaması; olaylar yürütülür.
- Uygulama aşaması çağrılır; olaylar yürütülür.
- Cevabı oluşturma aşaması.

Yukarıdaki altı faz, JSF'nin bir formu işlediği sırayı gösterir.



JSF, bir link veya bir düğmeye tıklanır tıklanmaz ve JSF bir istek alır almadır geri yükleme görünümü aşamasına başlar. Bu aşamada JSF görünümü oluşturur, olay işleyicileri ve doğrulayıcıları UI bileşenlerine bağlar. Doğrulayıcılar istemci tarafında girilen verilerin kurallara uyup uymadığını denetleme görevini üstlenir. Örneğin boş bırakılmaması gereken telefon numarası boş bırakılırsa doğrulayıcı tanımlı uyarıyı yapacaktır.

Bileşen ağacı oluşturulduktan veya geri yüklenildikten sonra, bileşen ağacındaki her bileşen, yeni değerini istek parametrelerinden ayıklamak için kod çözme yöntemini kullanır. Bileşen bu değeri saklar. İstek parametreleri istemcinin girdiği değerlerin sunucuya iletildiği biçimdir. Örneğin kullanıcının doğum tarihi alanına girdiği değer, sunucuya istek parametrelerinden birisi olarak gönderilecektir.

Süreç doğrulama aşamasında JSF, bileşen ağacında kayıtlı tüm doğrulayıcıları yürütür. Doğrulama için bileşen için tanımlı kuralları inceler ve bu kuralları bileşen için depolanan yerel değerle karşılaştırır.

JSF verinin geçerli olduğunu kontrol ettikten sonra, bileşen ağacının üzerinde yürütür ve karşılık gelen sunucu tarafı nesne özelliklerini istemci tarafından bileşenlerin yerel değerlerine ayarlar. JSF, bileşeninin değerine karşılık gelen bean özelliklerini güncelleyecektir.

Uygulamanın çağırılması aşamasında, JSF form gönderme / başka bir sayfaya bağlantı verme gibi uygulama düzeyinde olayları yürütür.

Son olarak cevap oluşturma aşamasında JSF, sayfayı oluşturmak için uygulama sunucusundan ister. İlk istek için (sayfaya ilk kez giriliyorsa), sayfada gösterilen bileşenler, bileşen ağacına eklenecektir. Bu bir ilk istek değilse, bileşen ağacı zaten oluşturulmuştur, bu nedenle bileşenlerin tekrar eklenmesine gerek yoktur. Her iki durumda da sunucu sayfadaki etiketleri ve bileşenleri işleyecektir. Ardından cevap istemciye gönderilecektir.

Bölüm Özeti

Bu bölümde e-ticaret uygulama konularından ve Web sitelerinin geliştirilmesi için sıkılıkla tercih edilen teknolojilerden birisi olan Java Server Faces ele alınmıştır. Öncelikle ne amaçla kullanıldığı inceleminmiştir. Daha sonra Java Server Faces ile Web sitesi geliştirebilmek için ihtiyaç duyulan kurulumların yapılması anlatılmıştır. Son olarak Java Server Faces' in mimarisi ve istemci – sunucu arasındaki iletişimde yaşam döngüsünün ne şekilde gerçekleştiği anlatılmıştır.

Kaynakça

ECLIPSE Foundation, <https://www.eclipse.org>, 2020.

Mahmoud Q. H., <https://www.oracle.com/technetwork/articles/java/javaserverfaces-135231.html>, 2019

ORACLE, <https://javaee.github.io/javaserverfaces-spec/>, 2020.

13. JSF İLE UYGULAMA GELİŞTİRME

Bu Bölümde Öğrenecekleriniz

13.1. İlk JSF uygulaması

13.2. Eclipse Projesinin hazırlanması

13.3. Projenin Eclipse IDE' ye aktarılması

Birlikte Düşünelim

1. Bir JSF uygulaması nasıl oluşturulur?
2. Eclipse ile JSF projeleri nasıl geliştirilir?
3. Servlet nedir?
4. Bir yönetilen Bean nedir?
5. WAR dosyası nedir?

Başlamadan Önce

Teorik konu anlatımının yanı sıra uygulamaya dönük örneklerin verilmesi özellikle e-ticaret gibi eğitimlerde hayatı önem taşımaktadır. Bu nedenle JSF kullanılarak giriş niteliğindeki bazı uygulamaların nasıl oluşturulduğu ile ilgili öğrencinin hazırlanması bu bölümde hedef alınmıştır. Böylece basit bir JSF uygulaması maven kullanılarak geliştirilecektir. Bu tür projelerin geliştirilmesinde en fazla kullanılan araçlardan birisi olan Eclipse ile bir projenin geliştirilebilmesi için gereken adımlar bu bölümde ele alınacaktır. Proje geliştirme esnasında, JSF projesinin Eclipse IDE bünyesinde geliştirilebilmesi için yapılması gerekenlerin incelenmesi de gerekmektedir. Servlet ile JSF arasındaki ilişkinin anlaşılması teorik olarak uygulama geliştiriciler açısından önem taşıyan bir başka konudur. Böylece Bir yönetilen Bean' in ne olduğu, kullanıcı ara yüzlerinin oluşturulması için yapılması gerekenler ve bir bütün olarak projenin son halinin verilmesi bu bölümde ele alınacak diğer konulardır. Son olarak gerçek bir projenin canlıya alınabilmesi için zorunlu olan sunucu üzerinde çalıştırılabilir dosyaların oluşturulması ve dağıtılması gösterilerek bu bölüm sonlandırılacaktır.

Giriş

JavaServer Faces (JSF), sunucu tarafı uygulamalara Kullanıcı Arabirimleri (UI) oluşturmak için standart bir özelliktir. JavaServer Faces'ten önce, web uygulamaları oluşturan geliştiriciler genellikle HTML uygulama arayüzü bileşenlerini servletlerle veya JavaServer Pages (JSP sayfaları) ile oluşturmaktaydı. Bunun nedeni, HTML kullanıcı arayüzü bileşenlerinin, web tarayıcılarının desteklediği en düşük ortak payda olmasıdır. Standart HTML arayzlarına dayalı web uygulamalarının, bağımsız istemciler ile karşılaşıldığında zengin kullanıcı arayüzleri olmadığı ve bu nedenle daha az işlevsellik ve / veya düşük kullanılabilirliğe sahip olduğu anlamına gelmektedir. Masaüstü uygulamalar zengin kullanıcı arayüzleri geliştirmek için kullanılabilirken, web uygulama geliştiricileri her zaman hangi müşterilerin uygulamaya erişeceğini bilmez ve / veya istemci cihaza erişimi olmayabilir.

Ayrıca, büyük ölçekli bir web sistemi geliştiriliyorsa, veritabanı sorguları oluşturmak için tablo görüntüleyici gibi özel bileşenlerin nasıl uygulanacağı gibi teknik zorluklarla karşılaşılabilir. Bu tür özel bileşenleri oluşturmak, yeni kütüphaneleri oluşturmak ve test etmek için uzmanlık ve önemli miktarda zaman gerektirir. İdeal bir ortamda, geliştiriciler, uygulama geliştirme ortamlarına entegre etmek için önceden oluşturulmuş, test edilmiş ve son derece yapılandırılabilir bileşenleri kullanabilirler.

JSF, zengin kullanıcı arayüzlerine sahip web uygulamaları geliştirmek için sunucu tarafında bir teknolojidir. JSF ile, özel kullanıcı arayüzü bileşenleri oluşturmak gibi teknik zorlukları çözebilirsiniz. Bunun nedeni, JSF teknolojisinin, sunucu tarafında çalışan ve kullanıcı arabirimini istemciye geri yükleyen Java tabanlı web uygulamaları oluşturmak için bir kullanıcı arabirimini iskeleti olmasıdır. Kullanıcı arayüzü kodu, istemcide oluşturululan olaylara cevap vererek şekilde sunucu üzerinde çalışır [Mahmoud – 2019].

13.1. İlk JSF Uygulaması

Basit bir JSF uygulaması oluşturmak için maven-archetype-webapp eklentisini kullanacağız. Aşağıdaki örnekte, C:\JSF klasöründe maven tabanlı bir web uygulaması projesi oluşturacağız.

Komut konsolunu açın, C: \> JSF dizinine gidin ve aşağıdaki mvn komutunu uygulayın:

```
C:\JSF>mvn archetype:create -DgroupId=com.ilkdgers.test -DartifactId=helloworld -DarchetypeArtifactId=maven-archetype-webapp
```

Maven işlemeye başlayacak ve eksiksiz bir java web uygulaması proje yapısı oluşturacaktır. Artık C:/JSF dizinine gidilebilir. Helloworld adlı (artifactId'de belirtildiği gibi) oluşturulan bir Java web uygulaması projesi göreceksiniz. Maven, standart bir dizin düzenini kullanır. Buna göre anahtar kavramlar Tablo 8' deki gibi açıklanır.

Tablo 8 Maven dizin düzeni

| Dizin Yapısı | Açıklama |
|--------------------|--|
| helloworld | src klasörü ve pom.xml içerir. |
| src/main/webapp | WEB-INF klasörü ve index.jsp sayfasını içerir. |
| src/main/resources | Görüntü/ozellik dosyalarını içerir (Genellikle bu yapıyı elle oluşturmamız gereklidir) |

Aşağıdaki JSF bağıllıkları pom.xml' e eklenir.

```
<dependencies>
<dependency>
<groupId>com.sun.faces</groupId>
<artifactId>jsf-api</artifactId>
<version>2.2.19</version>
</dependency>
<dependency>
<groupId>com.sun.faces</groupId>
<artifactId>jsf-impl</artifactId>
<version>2.2.19</version>
</dependency>
```

```
</dependencies>
```

Böylece pom.xml' in tamamı aşağıdakine benzer biçimde (sürüm sayılarında farklılıklar olabilir) olacaktır:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/maven-v4_0_0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.tutorialspoint.test</groupId>
  <artifactId>helloworld</artifactId>
  <packaging>war</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>helloworld Maven Webapp</name>
  <url>http://maven.apache.org</url>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>com.sun.faces</groupId>
      <artifactId>jsf-api</artifactId>
      <version>2.2.19</version>
    </dependency>
    <dependency>
      <groupId>com.sun.faces</groupId>
      <artifactId>jsf-impl</artifactId>
      <version>2.2.19</version>
    </dependency>
  </dependencies>
```

```
<build>
<finalName>helloworld</finalName>
<plugins>
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-compiler-plugin</artifactId>
<version>3.8.1</version>
<configuration>
<source>1.6</source>
<target>1.6</target>
</configuration>
</plugin>
</plugins>
</build>
</project>
```

13.2. Eclipse Projesinin Hazırlanması

Artık komut konsolunu açılarak C:\>JSF>helloworld dizinine gidilerek aşağıdaki mvn komutu yürütülebilir.

```
C:\JSF\helloworld>mvn eclipse:eclipse -Dwtpversion=2.0
```

Maven işlemeye başlayacak, Eclipse hazır projesini oluşturacak ve buna wtp yeteneği ekleyecektir.

13.3. Projenin Eclipse IDE' ye Aktarılması

Aşağıdaki adımlar izlenerek maven ile oluşturulan proje Eclipse IDE' ye aktarılabilir:

- Alma sihirbazını kullanarak projeyi Eclipse aktarın.
- “Dosya->İçe Aktar ...->Mevcut proje çalışma alanı” na gidin.
- Helloworld için kök dizini seçin.
- Kopyalanacak projeleri kontrol edilecek çalışma alanına kaydedin.
- Son düğmesine tıklayın.
- Eclipse, C:\>Projects->Data->WorkSpace çalışma alanına projeyi aktaracak ve kopyalayacaktır.

13.4. Servlet Yapılandırması

Bir servlet, belirli bir ağ isteğine (genellikle bir HTTP isteği) cevap veren bir java sınıfıdır. Temel olarak servlet'ler genellikle web uygulamalarını yürütmek için kullanılır Ancak, servlet'lerin sağladığı "bir HTTP isteği, bu HTTP isteğine yanıt yaz" gibi bir seviyeden daha yüksek düzeyde bir soyutlama vermek için servlet'lerin üzerinde çalışan çeşitli iskeletler de vardır.

Servlet'ler ağ tarafını idare eden bir servlet kabında çalışır (örneğin, bir HTTP isteği ayırtırma, bağlantı işleme vb.). En iyi bilinen açık kaynaklı servlet kaplarından biri Apache Tomcat' tır.

Kısaca bir servlet sunucu tarafından belirli uygulamalar için çalışan ve gelen istekleri dinleyen nesnelerdir. Tomcat tarafından idare edilir. İstemciden gelen istekleri anlamlandırmır ve istemciye HTML biçiminde uygun cevaplar oluşturup gönderir.

Bir Java servletinin bir tarayıcıdan erişilebilir olması için, servlet kabına hangi sunucuların konuşlandırılacağını ve hangi URL'lerin sunuculara eşleneceğini belirtmeniz gereklidir. Bu, Java web uygulamanızın web.xml dosyasında yapılır.

İlk önce sunucu uygulamasını yapılandırılır. Bu <servlet> elemanı kullanılarak yapılır. Burada servlet' e bir ad verirsiniz ve servlet' in sınıf adını yazarsınız.

İkincisi, servlet' i bir URL veya URL modeline eşlersiniz. Bu <servlet-mapping> ögesinde yapılır. Aşağıdaki örnekte .xhtml, .jsf, .faces ile biten tüm URL'ler servlet' e gönderilir.

<servlet> ögesi, servlet kabının servlet' i ne zaman yükleyeceğini denetlemek için kullanabileceğiniz <load-on-startup> adlı bir alt ögeye sahiptir. Bir <load-on-startup> ögesi belirtmezseniz, servlet kabı tipik olarak servlet' i ilk istek geldiğinde yükler. Burada yüklemek ile kastedilen sunucuda belleğe alınmasıdır.

<load-on-startup> 1 </load-on-startup> ögesinin içindeki sayı, servlet kabına servlet' lerin hangi sırayla yüklenmesi gerektiğini söyler. Önce düşük sayılar yüklenir. Değer negatifse veya belirtilmemişse, servlet kabı servlet' i herhangi bir zamanda yükleyebilir.

Webapp->WEB-INF klasöründe web.xml dosyasını bulun ve aşağıda gösterildiği gibi güncelleyin.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  id="WebApp_ID" version="2.5">
  <welcome-file-list>
    <welcome-file>faces/home.xhtml</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
```

```

<url-pattern>/faces/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>Faces Servlet</servlet-name>
<url-pattern>*.jsf</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>Faces Servlet</servlet-name>
<url-pattern>*.faces</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>Faces Servlet</servlet-name>
<url-pattern>*.xhtml</url-pattern>
</servlet-mapping>
</web-app>

```

13.5. Bir Yönetilen Bean Oluşturma

Java Bean' ler birçok nesneyi tek bir nesne (bean) içine alan sınıflardır. Aşağıdaki kuralları takip etmesi gereken bir java sınıfıdır:

1. Serileştirilebilir (içerdiği tüm verilerle diske kaydedilebilir) olmalıdır.
2. Genel (her yerden erişilebilen) argümansız bir kurucuya sahip olmalıdır.
3. Java Bean' deki tüm özellikler özel (dışarıdan erişilemez) olmalıdır. Ancak dışarıdan erişmek için getter ve setter olarak bilinen genel metodlara sahip olmalıdır.

Yönetilen Bean, JSF'ye kayıtlı düzenli bir Java Bean sınıfıdır. Başka bir deyişle, Yönetilen Bean, JSF iskeleti tarafından yönetilen bir Java Bean' dir. Yönetilen Bean, getter ve setter yöntemleri, iş mantığını ve hatta bir destek Bean' i içerebilir.

src->main->java altında com->ilkders->test olarak bir paket yapısı oluşturun. Bu pakette HelloWorld.java sınıfı oluşturun. HelloWorld.java kodunu aşağıda gösterildiği gibi güncelleyin.

```

1 package com.ilkders.test;
2
3 import javax.faces.bean.ManagedBean;
4
5 @ManagedBean(name = "helloWorld")
6 public class HelloWorld {
7     public HelloWorld() {

```

```

8 System.out.println("HelloWorld started!");
9 }
10
11 public String getMessage() {
12 return "Hello World!";
13 }
14 }
```

Küme parantezleri bir bloğu temsil etmektedir. Blok içi blok mümkündür. Burada 6. satırda sınıf tanımlanmaya başlamıştır. Açılan küme parantezi 14. Satırda kapandığından 6 – 14. satırlar arasında kalanlar *HelloWorld* isimli Bean' in içeriğidir. 5. satırda bunun bir Yönetilen Bean olduğu ve erişim isminin *helloWorld* olduğu belirtilmiştir. 7. satırda sınıfla aynı isme sahip bir metod görülmektedir. Bu metod oluşturucu metoddur ve herhangi bir argüman almamıştır. Oluşturucu metod söz konusu sınıftan bir nesne oluşturulduğu an otomatik olarak yürütülen metottur. Bu durumda bu sınıftan bir nesne oluşturulduğunda, sunucu tarafında terminalde “Hello World started!” yazısının görülmesi beklenir. 11 – 13 satırları arasında da çağrıldığı yere “Hello World!” değerine sahip metin (String) türü sonuç geri döndüren bir metod görülmektedir.

13.6. Bir JSF Sayfasının Oluşturulması

Webapp klasörü altında home.xhtml sayfasını oluşturun. home.xhtml kodunu aşağıda gösterildiği gibi güncelleyin.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>JSF Örneği!</title>
</head>
<body>
#{helloWorld.getMessage()}
</body>
</html>
```

Daha önce html etiketlerinin anlamları ele alınmıştı. Bu kısımda ilk defa görülen `#{} etiketi` ve içine aldığı komutlardır. `#{} ertelenen ifadeler` içindir (sayfanın yaşam döngüsüne bağlı olarak çözümlenir) ve bir Bean' den okumak, ona yazmak veya bir yöntem çağrıları yapmak için kullanılabilir. Bir önceki kısımda yer alan Yürüten Bean' e erişim ismi olarak *helloWorld* verilmiştir. Bu nedenle *helloWorld.getMessage()* söz konusu sınıftan oluşturulmuş nesnenin *getMessage()* metodunu çağıracaktır. Bu metodun geriye “Hello World!” metnini döndürdüğünü biliyoruz. O zaman sunucu tarafında sayfa oluşturulurken `#{} etiketi` içerisindeki komutlar yürütülecek ve sonuçlar ilgili kısma yerleştirilerek istemciye gönderilecektir.

13.7. Projenin İnşa Edilmesi

Proje inşa edilmesi aşağıdaki adımların takip edilmesiyle sağlanır:

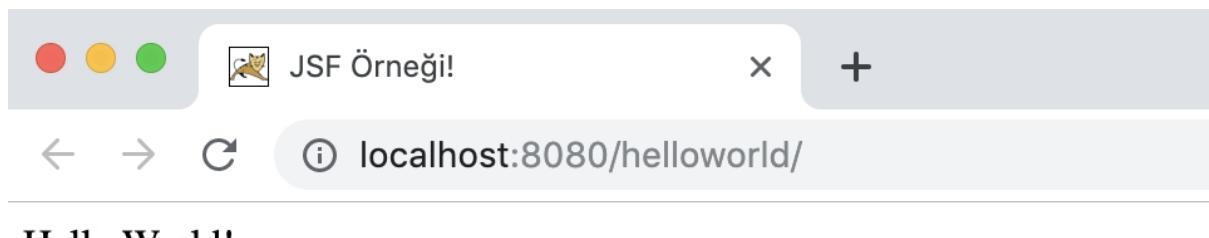
- Eclipse içinde helloworld projesini seçin
- Farklı Çalıştır sihirbazını kullan
- Farklı Çalıştır -> Maven Paketi'ni seçin.
- Maven projeyi oluşturmaya başlayacak ve "C:\Projects->Data->WorkSpace->helloworld->target" klasörü altında helloworld.war oluşturacaktır.

13.8. WAR Dosyasının Sunucuda Dağıtılması

Web sunucusu olarak kullanılan bilgisayara oluşturulan war dosyası dağıtıldığında artık sunucuya tarayıcı ile erişen herkes oluşturduğumuz sayfalara erişebilecektir. Bunun için aşağıdaki adımların takip edilmesi gerekmektedir:

- Tomcat sunucusunu durdurun.
- Helloworld.war dosyasını tomcat kurulum dizininde -> webapps klasörüne kopyalayın.
- Tomcat sunucusunu başlatın.
- Webapps dizinine bakın, helloworld'ün oluşturduğu bir klasör olmalı.
- Şimdi helloworld.war başarıyla Tomcat Webserver kökünde konuşlandırıldı.

Uygulamayı başlatmak için web tarayıcısına <http://localhost:8080/helloworld/home.jsf> URL' sini girin. Sunucu adı (localhost) ve port (8080) tomcat yapılandırmanızla göre değişebilir.



Şekil 26 İlk JSF uygulaması

Bölüm Özeti

Bu bölümde e-ticaret uygulama konularından birisi olan Java Server Faces uygulaması ele alınmıştır. Öncelikle basit bir uygulama örneği verilmiştir. Daha sonra Eclipse ile Java Server Faces projesi geliştireilmek için ihtiyaç duyulan ayarlamaların yapılması anlatılmıştır. Son olarak Java Server Faces kullanılarak Eclipse ile geliştirilen bir projenin hayatı geçirilmesi gösterilmiştir.

Kaynakça

ECLIPSE Foundation, <https://www.eclipse.org>, 2020.

Mahmoud Q. H., <https://www.oracle.com/technetwork/articles/java/javaserverfaces-135231.html>, 2019

ORACLE, <https://javaee.github.io/javaserverfaces-spec/>, 2020.

14. YÖNETİLEN BEAN'LER VE SAYFA NAVİGASYONU

Birlikte Düşünelim

1. JSF ifade dili nedir?
2. @ManagedBean ek açıklaması ne amaçla kullanılır?
3. JSF ile sayfalar arası navigasyon nasıl sağlanır?

Başlamadan Önce

Başarılı web uygulamaları geliştirebilmek için teknolojiyi doğru ve etkili kullanmak gereklidir. Eski teknoloji araçlarla geliştirilen web sayfaları bu nedenle de fazla kullanıcı çekmeyecektir. Bu bölümün en önemli konusu ve JSF teknolojisinin önemli araçlarından olan yönetilen Bean'lerin nasıl belirtildiğine yönelik yaklaşımların ele alınması bir zorunluluktur. Ayrıca Html ara yüz sayfaları ile yönetilen Bean'lerin dinamik bağlantılar kurabilmesi için ihtiyaç duyulan dile giriş yapılacaktır. Bu dersin konusu yalnız olarak web uygulama geliştirme olmadığından tüm konulara derinlemesine girilmemekte daha çok bazı örneklerle öğrenciye yol gösterilmektedir. Dolayısıyla Yönetilen Bean'lerin JSF uygulamalarında tümü değil fakat en yaygın belirtilme biçimini bu bölümde incelenecektir. Son olarak kullanıcılarımızın Web sayfaları arasında gezinti yapabilmesini sağlamak için alternatif bazı yöntemler gösterilecektir.

Giriş

Yönetilen Bean'ler, UI bileşeni için Model olarak çalışır. Yönetilen Bean' e JSF sayfasından erişilebilir.

JSF 1.2'de, bir yönetilen bean' in faces-config.xml gibi JSF yapılandırma dosyasına kaydedilmesi gerekiyordu. JSF 2.0'dan itibaren, yönetilen bean ek açıklamalar kullanılarak kolayca kaydedilebilir. Bu yaklaşım, bean'leri ve kaydını bir yerde tutar, böylece yönetimi daha kolay hale gelir.

XML Konfigürasyonu aşağıdaki gibidir:

```
<managed-bean>
<managed-bean-name>helloWorld</managed-bean-name>
<managed-bean-class>com.ilkders.test.HelloWorld</managed-bean-class>
<managed-bean-scope>request</managed-bean-scope>
</managed-bean>
<managed-bean>
<managed-bean-name>message</managed-bean-name>
<managed-bean-class> com.ilkders.test.Message</managed-bean-class>
<managed-bean-scope>request</managed-bean-scope>
</managed-bean>
```

Ek açıklama kullanımı ise aşağıdaki gibi olacaktır:

```

@ManagedBean(name = "helloWorld", eager = true)

@RequestScoped

public class HelloWorld {

    @ManagedProperty(value="#{message}")

    private Message message;

    ...

}

```

14.1. JSF İfade Dili (EL)

JSF zengin bir ifade dili sağlar. `#{işlem-ifade}` gösterimini kullanarak normal işlemler yazabiliriz. Aşağıdakiler JSF İfade dilinin avantajlarından bazıları.

- Bean özelliklerini referans alabilir ki burada bean' in istek, oturum veya uygulama kapsamında saklanan bir nesne olabilmesi veya bir yönetilen bean olması söz konusudur.
- Bir list, map veya bir array olarak ifade edilen bir koleksiyonun öğelerine kolay erişim sağlar.
- İstemci tarafı veri ve bilgilere sahip request gibi önceden tanımlanmış nesnelere kolay erişim sağlar.
- Aritmetik, mantıksal ve ilişkisel işlemler ifade dili kullanılarak yapılabilir.
- Otomatik tür dönüşümünü sağlar. Örneğin bir tamsayı hiçbir işlem yapılmaksızın metne dönüsür.
- Eksik değerleri NullPointerException hatası yerine boş olarak gösterir.

14.2. @ManagedBean Ek Açıklaması

`@ManagedBean`, name özelliğinde belirtilen adla yönetilen bir bean' i işaretler. Name niteliği belirtilmemezse, yönetilen bean adı varsayılan olarak tam sınıf adı olur. Ancak baş harf her zaman küçük olacaktır. Bizim örneğimizde, name niteliğine herhangi bir değer verilmemesi halinde varsayılan olarak helloWorld olurdu.

Bir başka önemli özellik eager özelliğidir. Eğer `eager = "true"` ise, ilk defa çağrılmadan önce yönetilen bean yaratılır, aksi takdirde sadece istendiğinde yaratılacak olan "tembel" başlatma kullanılır. Başka bir ifadeyle tembel yöntem, sunucuda belleğin daha etkili ancak performansın nispeten düşük olmasına neden olan bir yöntemdir. Zira istemci arayüze istek yaptığı anda sunucu belleğinde yaratılacak ve istemciye cevap verilecektir. Aksi halde ise istek yapılmadan yaratılacak ve sunucu tarafında belleğin şısmesine neden olacaktır.

14.2.1. Kapsam Ek Açıklamaları

Kapsam ek açıklamaları, yönetilen bean' in yerleştirileceği kapsamı belirler. Kapsam belirtilmemişse, o zaman bean varsayılan olarak request kapsamına girer. Her kapsam Tablo 9' da kısaca açıklanmıştır.

Tablo 9 Kapsamlar ve ömürleri

| Kapsam | Açıklama |
|-----------------------------|---|
| <code>@RequestScoped</code> | Bean, HTTP istek yanıtı devam ettiği sürece yaşar. Bir HTTP isteği üzerine oluşturulur ve HTTP isteği ile ilişkilendirilmiş HTTP yanıtı tamamlandığında yok edilir. |
| <code>@NoneScoped</code> | Bean, tek bir EL değerlendirmesi olduğu sürece yaşar. EL değerlendirmesinde yaratılır ve EL değerlendirmesinden hemen sonra imha edilir. |
| <code>@ViewScoped</code> | Bean, tarayıcı penceresinde / sekmesinde, kullanıcı aynı JSF görünümüyle (bir xhtml arayüzüne bağlı bean) etkileşime girdiği sürece yaşar. Bir HTTP isteği üzerine oluşturulur ve kullanıcı farklı bir görüntüme geri gönderdiğinde yok edilir. |

| | |
|--------------------|---|
| @SessionScoped | Bean, HTTP oturumu yaşadığı sürece yaşar. Bu bean' i oturumda içeren ilk HTTP isteği üzerine yaratılır ve HTTP oturumu sonlandığında imha edilir. Kullanıcı güvenli çıkış seçeneğiyle çıkmadığı sürece, siteden çıkışa bile varsayılan olarak genellikle 30 dakika boyunca oturumu sonlanmaz. |
| @ApplicationScoped | Bean, web uygulaması yaşadığı sürece yaşar. Uygulamada bu bean' i içeren ilk HTTP isteği üzerine yaratılır (veya eager = true niteliği @ManagedBean'da ayarlanmışsa web uygulaması başlatıldığında) ve web uygulaması kapandığında yok olur. |
| @CustomScoped | Bean, bu kapsam için oluşturulan özel Haritaya, bean' in girişi devam ettiği sürece yaşar. |

14.2.2. @ManagedProperty Ek Açıklaması

JSF basit bir statik Bağımlılık Enjeksiyonu (DI) iskeletidir. @ManagedProperty ek açıklamasını kullanarak, bir yönetilen bean özelliği başka bir yönetilen bean' e enjekte edilebilir.

Yönetilen bean' ler ile yukarıdaki ek açıklamaları test etmek için bir test JSF uygulaması oluşturalım.

1. JSF ile uygulama geliştirme bölümünde açıklandığı gibi com.ilkders.test paketi altında *helloworld* adında bir proje oluşturun.
2. HelloWorld.java'yı aşağıda açıklandığı şekilde değiştirin. Dosyaların kalanını değiştirmeden tutun.
3. Aşağıdaki açıklandığı gibi com.ilkders.test paketinin altında Message.java oluşturun.
4. İş mantığının gereksinimlere göre çalıştığından emin olmak için uygulamayı derleyin ve çalıştırın.
5. Son olarak, uygulamayı WAR dosyası biçiminde oluşturun ve Apache Tomcat Web Sunucusu'nda konuşlandırın.
6. Web uygulamanızı, son adımda aşağıda açıklandığı şekilde uygun URL kullanarak başlatın.

HelloWorld.java sınıfı aşağıdaki gibi olacaktır:

```

1 package com.ilkders.test;
2
3 import javax.faces.bean.ManagedBean;
4 import javax.faces.bean.ManagedProperty;
5 import javax.faces.bean.RequestScoped;
6
7 @ManagedBean(name = "helloWorld", eager = true)
8 @RequestScoped
9 public class HelloWorld {
10
11 @ManagedProperty(value="#{message}")
12 private Message messageBean;
13 private String message;
14
15 public HelloWorld() {
16     System.out.println("HelloWorld started!");

```

```
17 }  
18  
19 public String getMessage() {  
20   message = messageBean.getMessage();  
21   return message;  
22 }  
23  
24 public void setMessageBean(Message message) {  
25   this.messageBean = message;  
26 }  
27}
```

HelloWorld yönetilen bean' i 12. satırda görüldüğü gibi *Message* isimli başka bir bean' i *messageBean* değişkeniyle kullanmaktadır. Eğer bu bir yönetilen bean olmasaydı ve *@ManagedProperty* ek açıklamasıyla tanımlanmasaydı aşağıdaki gibi bir satırla öncelikli olarak bu sınıftan bir nesne oluşturamız gerekecekti. Ancak *Message* da bir yönetilen bean' dir ve HelloWorld yönetilen bean' inin *message* özelliğine bağlanmıştır. Bu nedenle arayüzden *message* özelliğine erişildiğinde (*getMessage* ve *setMessage* metotlarıyla kendiliğinden yaratılacaktır.

```
messageBean = new Message();
```

Bu durumda *Message.java* sınıfı aşağıdaki gibi olacaktır:

```
1 package com.ilkders.test;  
2  
3 import javax.faces.bean.ManagedBean;  
4 import javax.faces.bean.RequestScoped;  
5  
6 @ManagedBean(name = "message", eager = true)  
7 @RequestScoped  
8 public class Message {  
9  
10 private String message = "Hello World!";  
11  
12 public String getMessage() {  
13   return message;  
14 }
```

```

16 public void setMessage(String message) {
17   this.message = message;
18 }
19}

```

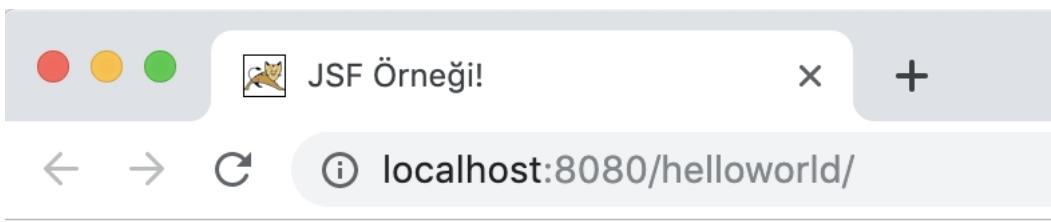
Son olarak home.xhtml arayüzü aşağıdaki gibi güncellenmelidir:

```

<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
<h:head>
  <title>JSF Örneği!</title>
</h:head>
<h:body>
  #{helloWorld.message}
</h:body>
</html>

```

Yapılan tüm değişiklikleri hazır hale getirdiğinizde, uygulamayı JSF ile Uygulama Geliştirme bölümünde yaptığımız gibi derleyelim ve çalıştırıralım. Uygulamanızla ilgili her şey yolundaysa, Şekil 27' deki sonucu verecektir.



Hello World!

Şekil 27 @ManagedProperty ek açıklaması örneği

14.3. Sayfa Navigasyonu

Navigasyon kuralları, bir düğme veya bir bağlantı tıklandığında hangi görünümün gösterileceğini açıklayan JSF iskeleti tarafından sağlanan kurallardır.

Navigasyon kuralları, faces-config.xml adlı JSF yapılandırma dosyasında veya yönetilen bean' lerde tanımlanabilirler.

Navigasyon kuralları, elde edilen görünümün gösterilebileceği koşulları da içerebilir. JSF 2.0, navigasyon kurallarını bu şekilde tanımlamaya gerek olmayan kapalı navigasyon sağlar.

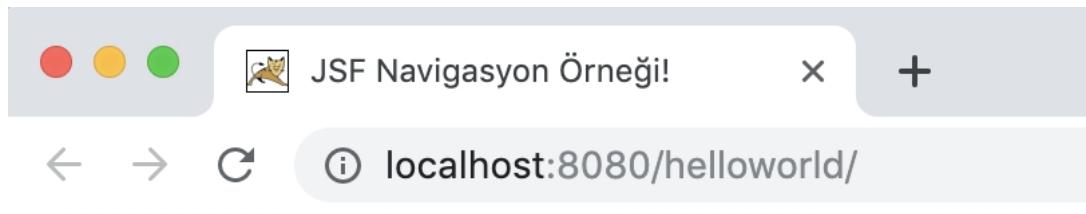
14.3.1. Örtük Navigasyon

JSF 2.0, örtük navigasyon adlı otomatik görüntüleme sayfası çözümleyici mekanizması sağlar. Bu durumda, ara yüz adını yalnızca eylem özniteliğine koymaınız gereklidir; JSF konuşlandırılmış uygulamada otomatik olarak doğru ara yüz sayfasını arayacaktır.

Herhangi bir JSF UI Bileşeninin işlem niteliğinde ara yüz adının aşağıdaki gibi ayarlanması ilgili ara yüz sayfasına geçebilmeyi sağlayacaktır.

```
1 <h:form>
2 <h3>JSF Sonucunu kullanarak</h3>
3 <h:commandButton action="page2" value="Sayfa 2"/>
4 </h:form>
```

Bir ara yüzde herhangi bir etkileşimli bileşenin oluşturulabilmesi için ilgili sayfadaki tüm bileşenlerin *form* etiketi içerisinde alınmış olması gereklidir. Ana sayfaya 3. satırda görüldüğü gibi bir düğme eklenmiştir. Burada, page2 düğmesi tıklatıldığında, JSF, page2 sayfa adını page2.xhtml uzantısı olarak çözer ve page2.xhtml dosyasını görüntüler. Böylece ana sayfadan page2.xhtml' e gezinti (navigasyon) yapılmış olur. page1.xhtml ve page2.xhtml ara yüz dosyalarının Webapp klasörü içerisinde eklendikten ve home.xhtml' de gerekli düzenlemeler de yapıldıktan sonra proje çalıştırıldığında Şekil 28'de görüldüğü gibi iki düğme ile farklı sayfalara navigasyon imkanı sağlanmış olacaktır.



JSF Sonucunu kullanarak

Sayfa 2

Yönetilen Bean kullanarak

Sayfa 1

Şekil 28 JSF navigasyon örneği

Sayfa 2 isimli düğmenin page2.xhtml ara yüzüne navigasyon sağladığı yukarıda gösterilmiştir. Page2.xhtml dosyasının aşağıdaki gibi olduğunu varsayıp olursak. İlgili düğme tıklandığında Şekil 29' da görülen ara yüze ulaşılacaktır.

```
<?xml version='1.0' encoding='UTF-8' ?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml"
```

```
    xmlns:h="http://xmlns.jcp.org/jsf/html">
```

```
<h:head>
```

```
<title>Sayfa 2</title>
```

```
</h:head>
```

```
<h:body>
```

```
Burası ikinci sayfa
```

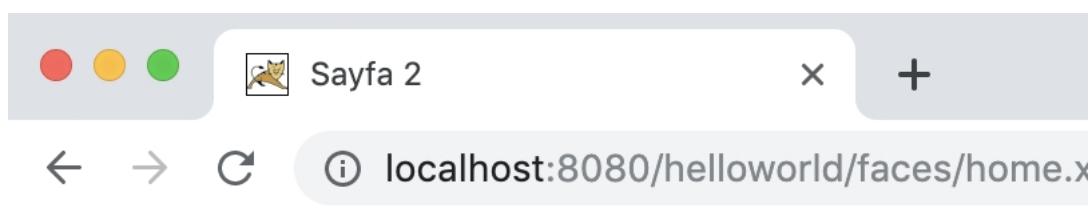
```
<h:form>
```

```
<h:commandButton action="home" value="Ana Sayfaya geri dön"/>
```

```
</h:form>
```

```
</h:body>
```

```
</html>
```



Burası ikinci sayfa

[Ana Sayfaya geri dön](#)

Sekil 29 İkinci sayfaya navigasyon

Bir görünüm adı döndürmek için yönetilen bean metodu aşağıdaki gibi tanımlanabilir. Böylece söz konusu metot bir düğmenin eylemine bağlandığında ve bu düğme tıklandığında metodun geri döndürdüğü ara yüze navigasyon sağlanmış olacaktır.

```
1 @ManagedBean(name = "navigationController", eager = true)
```

```
2 @RequestScoped
```

```
3 public class NavigationController implements Serializable {
```

```
4
```

```
5     public String moveToPage1(){
```

```
6         return "page1";
```

```
7     }
```

```
8 }
```

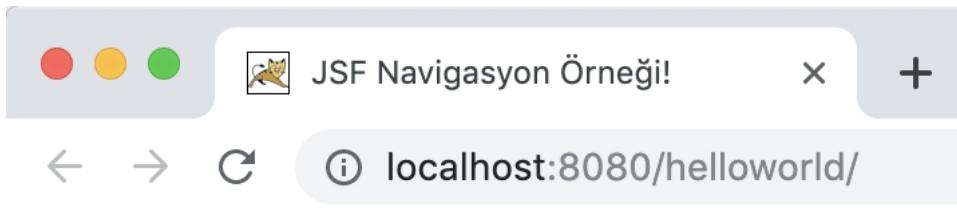
Yönetilen bean kullanarak, herhangi bir JSF UI Bileşeninin action niteliğinde ara yüz adı aşağıdaki gibi alınabilir.

```
1 <h:form>
2 <h3>Yönetilen Bean kullanarak</h3>
3 <h:commandButton action="#{navigationController.moveToPage1}"
4 value="Sayfa 1" />
5 </h:form>
```

Burada, Sayfa 1 düğmesi tıklatıldığında JSF, *moveToPage1* metodunun geri döndürdüğü sayfa1 adını sayfa1.xhtml uzantısı olarak çözer ve ilgili dizinde sayfa1.xhtml isimli dosyayı bulur.

14.3.2. Koşullu Navigasyon

Yönetilen bean kullanarak, navigasyonu kolayca kontrol edebiliriz. Yönetilen bir bean' de yer alan bir metoda parametre vererek bu en kolay biçimde sağlanır. Şekil 30' da home.xhtml ara yüzü üç düğme barındıran bir navigasyon sayfasına dönüştürülmüştür. Her düğme ayrı bir sayfaya yönlendirmektedir. Bunun için hepsi de aşağıda görüldüğü gibi yönetilen bean' de aynı metodu farklı parametrelerle çağrımaktadır.



Şekil 30 Koşullu navigasyon örneği

```
1 @ManagedBean(name = "navigasyonKontrol", eager = true)
2 @RequestScoped
3 public class NavigasyonKontrol implements Serializable {
4
5 public String showPage(int pageIndex) {
6 if (pageIndex == 0) {
7 return "home";
8 }
9 if (pageIndex == 1) {
10 return "page1";
```

```
11 } else if (pageId == 2) {  
12   return "page2";  
13 } else {  
14   return "home";  
15 }  
16 }  
17}
```

Göründüğü gibi *navigasyonKontrol* isimli yönetilen bean 5. Satırda *showPage* isimli bir metot tanımlamıştır. Söz konusu metot *pageId* isminde bir parametreye sahiptir. if – else if (eğer ise – değil ise) koşul bloğu ile metoda gönderilen parametre değerini kontrol etmekte ve buna göre bir geri değer döndürmektedir. Örneğin *pageId* değeri 2 ise metot geriye “page2” metnini döndürmektedir. *home.xhtml* ara yüzünün gövde kısmı aşağıdaki gibi güncellendiğinde navigasyon başarıyla gerçekleşecektir.

```
<h:body>  
<h3>Koşullu Navigasyon</h3>  
<h:form>  
  <h:commandButton action="#{navigasyonKontrol.showPage(1)}"  
  value="Sayfa 1"/>  
  <h:commandButton action="#{navigasyonKontrol.showPage(2)}"  
  value="Sayfa 2"/>  
  <h:commandButton action="#{navigasyonKontrol.showPage(3)}"  
  value="Ana Sayfa"/>  
</h:form>  
</h:body>
```

Burada "Sayfa1" butonuna tıklandığında.

- JSF, *pageId* = 1 parametresiyle bir istek oluşturur
- Ardından JSF, bu parametreyi *navigasyonKontrol* yönetilen bean'ının *showPage* metoduna aktarır.
- Daha sonra *pageId* kontrol edildikten sonra *navigasyonKontrol.showpage(1)* geriye “page1” metnini döndürür.
- JSF, *page1* olan görünüm adını *page1.xhtml* uzantısına dönüştürür
- Geçerli dizinde karşılık gelen görünüm dosyası *page1.xhtml*' i bulur.

Bölüm Özeti

Bu bölümde e-ticaret uygulama konularından birisi olan Java Server Faces uygulanabilmesi için önemli bazı kavramlar ele alınmıştır. Öncelikle Java Server Faces ifade dilinin ne olduğu açıklanmıştır. Daha sonra Java Server Faces için en temel kavramlardan birisi olan *@ManagedBean* ile yönetilen Bean'lerin nasıl belirtildiği anlatılmıştır. Son olarak Web sayfaları arasında gezinti yapma yöntemleri anlatılmıştır.

Kaynakça

ECLIPSE Foundation, <https://www.eclipse.org>, 2020.

Mahmoud Q. H., <https://www.oracle.com/technetwork/articles/java/javaserverfaces-135231.html>, 2019

ORACLE, <https://javaee.github.io/javaserverfaces-spec/>, 2020.

[1] Bazen “tıklama sayısı” olarak kullanıldığı görülebilmektedir.

[2] Gönderilen bir mesajın kabul edilmeme imkânı yoktur.

[3] Bazı VTYS’ leri ilişki ve özellik isimlerinde boşluk bırakılmasına izin vermeyebilir. Hatta bazıları Türkçe karakterlerin kullanımına da izin vermeyebilir. Ancak çok fazla karmaşıklaştmamak için bu çalışmada Türkçe karakterler kullanılmıştır.

[4] Başlarda kişisel ana sayfalar (Personel Home Pages) kısaltması olarak kullanılsa da günümüzde yardımcı metin ön-islemcisi (PHP: Hypertext Preprocessor) olarak bilinir.