

Yüksek Lisans Tezi
Trakya Üniversitesi Fen Bilimleri Enstitüsü

ÖZET

Toplumların gelişiminde eğitimin rolü büyüktür. Medeniyetlere en parlak zamanlarını yaşatan ya da tarih sahnesinden silen belirleyici unsur, büyük ölçüde eğitim olmuştur. Yıllar boyu pek çok farklı yöntem ve yaklaşım eğitimi etkilemiş, öğrenmeyi en üst düzeye çıkarabilecek sistemler üzerindeki çalışmalar aralıksız devam etmiştir ve etmektedir. Eğitimi temelden etkileyen teknolojik yenilikler ve buluşlar, her defasında bir önceki sisteme göre üstünlükler sağlamakta, yeni kavramların ortaya çıkmasına neden olmaktadır.

Uzaktan eğitimin temelini oluşturan sürecin, yazılı kaynakların basılı kaynaklar haline dönüşmesi, bu sayede rahat çoğaltılması, dağıtılabilir ve kolay erişebilir bir hale gelmesi olduğunu söyleyebiliriz. Bu yazıda uzaktan eğitim ile ilgili önemli kavramlara değinilmekle birlikte, sistemin sağladığı temel avantajlar ve özelliklere de yer verilmektedir. Bu kavramlar; çoğaltılabilir olma, dağıtılabilir olma ve kolay erişebilir olma olarak özetlenebilir. Tüm bu avantajlara bilgisayar destekli sistemlerin de entegre edilmesi, çoklu ortam araç ve tekniklerinden yararlanılması, Internet üzerinden içeriğe hızlı ve uygun maliyetler ile ulaşılması, kullanıcı etkileşiminin yeni teknolojiler yardımı ile artırılması, günümüzde uzaktan eğitimin kimi zaman örgün eğitime destek, kimi zaman ise başlı başına bir eğitim tekniği olarak kabul görmesini sağlamıştır.

E-Learning ya da uzaktan eğitim sistemler internetin gelişmesi ve band genişliğinin artması ile daha önem kazanmıştır. Örgün eğitimin maliyetinin yarısına kadar bir maliyeti olan e-Learning sistemler birçok fonksiyonun birleşmesinden oluşan sistemlerdir. Bunların için projeler, sınavlar, değerlendirmeler, ödevler gibi bir çok modül vardır. İşte bu konuda e-Sınav olarak bir modül hazırlamaya karar verdik ve bunu Web Teknolojisi üzerinden ASP ve ASP.Net dilleri kullanarak yazdık.

Tezin tamamı dört bölümden oluşmaktadır.

İlk iki bölümde e-Learning sistemler ve kullandığım Programlama dilleri birde onların birbirlerine üstünlüklerini ele alacağım.

Üçüncü bölümde geliştirilen uygulamaya ayrılmıştır.

Dördüncü bölüm Tartışma ve Sonuç Bölümüdür.

Anahtar Sözcükler:

Master Thesis
Trakya University Graduate School of
Natural and Applied Sciences
Department of Computer Engineering

SUMMARY

Education has an important role in the development of societies. Also it has been the most determining factor for civilizations *to be or not to be* in the scene of history. For many years, several methods and approaches have affected education; studies on systems that may increase learning to the highest level have been carried on and are still being done so. The technological innovations and inventions that affect education radically provide superiorities relative to the former system and led to new concepts.

We can state that the process which forms the basis of distant education includes the change of the written resources into pressed resources and in this way their easily multiplication, distribution and as a result being easily accessible. In this text, besides mentioning significant concepts related to distant education, the basic advantages and features that the system provides are also presented. These concepts can be stated as; being multipliable, being distributable, and being easily accessible. Also the integration of computer-assisted systems into all these advantages, the use of multimedia tools and techniques, the access to content fast and with relevant costs through Internet, the increase in user interaction with the help of new technologies made distant education an important part of formal education. Furthermore, distant education in some cases is seen as an educational technique itself.

E-Learning or distant education systems became more important with the development of Internet and with the increase in the bandwidth. E-Learning systems, which cost half the cost of formal education, are systems that composed of the combination of several functions. In these are several modules such as projects, exams, evaluations, homework. In this point, we decided to prepare a module called e-Exam and we prepared this using ASP and ASP.NET languages through Web Technology.

Thesis includes four parts.

In the first and second part, I talked about e-Learning systems and the programming languages I used and I mentioned the superiorities of each to the other.

Third part is for the application that is developed.

And in the last part is the Discussion and Conclusion.

Keywords:

Year: 2006

Page : 60

ÖZET	i
SUMMARY.....	iii
BÖLÜM 1.....	1
1.1 E-LEARNING MODELİ	3
1.2 WEB TABANLI UZAKTAN EĞİTİM SİSTEMLERİ (WTUES)	5
1.3 E-SINAV SİSTEMİ.....	8
1.3.1 İşlevsel Gereksinimler	8
1.3.2 Veri tabanının hazırlanması	9
1.3.3 Sınav bilgilerinin, soruların ve cevapların girilmesi	9
1.3.4 Sınavın Yapılması.....	10
1.3.5 Sınavın Yapılma şekli:	11
1.3.6 Cevapların notlandırılması.....	12
1.3.7 Sınav notlarının yayımlanması.....	12
1.4 DİĞER GEREKSİNİMLER	13
1.4.1. Güvenlik gereksinimleri.....	13
1.4.2. Performans gereksinimleri.....	13
1.5 SONUÇLAR.....	13
BÖLÜM 2.....	15
2.1 ASP.....	15
2.1.1 ASP'de Kullanılan Nesneler.....	15
2.1.2 Örnek Kodlar	16
2.2 ASP.NET	17
2.3 ASP.NET İLE GELEN YENİLİKLER.....	18
2.3.1 Daha Az Kod	18
2.3.2 Uygulama ve Sunumun Ayrılması	19
2.3.3 Derlenmiş Kod	19
2.3.4 Değişebilen Mimari.....	19
2.3.5 Bileşen Kullanımı	19
2.3.6 Her Kullanıcıda Çalışır.....	20
2.3.7 İstemci Durumunun İdaresi	20
2.3.8 ASP İle Tam Uyum	21
2.3.9 Kullanıcı Verimliliğindeki Artış.....	21
2.3.10 Konfigürasyon ve Mimari.....	21
2.3.11 Konfigürasyon Dosyaları.....	21
2.3.12 HTTP Runtime.....	23
2.3.13 HTTP Runtime'ın Yapısı.....	24
2.3.14 HTTP Handler(tutucu).....	24
2.3.15 Http Module(modül).....	25
2.3.16 Global.asax (uygulama konfigürasyon dosyası).....	25
2.4 KLASİK ASP'DEN ASP.NET'E DÖNÜŞÜM.....	27
BÖLÜM 3.....	34
UYGULAMA	34
3.1 PROGRAM ANALİZİ.....	34

3.2 UYGULAMA İÇİN KULLANILAN PROGRAMLAR	35
3.2.1 Asp	35
3.2.2 Internet Information Services(IIS).....	36
3.2.3 Microsoft Access	39
3.2.3.1 Tablo Oluşturmak	40
3.2.3.2 Veri türleri.....	41
3.2.3.3 Veri Tabanı Tasarımı.....	42
3.3 PROGRAMIN AKIŞ ŞEMALARI	43
3.4 PROGRAMIN ÇALIŞMASI :	45
BÖLÜM 4.....	49
TARTIŞMA VE SONUÇ.....	49
KAYNAKLAR.....	50
TEŞEKKÜR.....	51
ÖZGEÇMİŞ.....	52

BÖLÜM 1

Günümüzde varlığını sürdürebilmek, her alanda başarılı ve öncü olmak ile eş değer bir yaklaşım ortaya koymayı gerektirmektedir. Salt ekonomik, toplumsal, askeri, siyasal ya da kültürel gelişim yeterli olmamaktadır. Tüm bu unsurların birlikte gelişmesi, olgunlaşması ve üretime yansması gerekmektedir. Burada bilgiyi üretebilen toplumlar başarıya en yakın olanlardır. Toplumun bilgiyi üretebilmesi, toplumu oluşturan bireylerin eğitim düzeyleri ile doğrudan ilgilidir. Bilgi düzeyini sürekli artırabilen ve güncel tutabilen bireyler ancak, toplumsal bilgi üretimine katkı sağlayabilirler. Yaşam boyu eğitim kavramı burada kendini göstermekte, temel eğitim süreci ile genelde tamamlandığını düşündüğümüz kişisel gelişimin sürekli olması fikri, bu sayede değer kazanmaktadır.

Yaşam boyu eğitimin birey üzerine getirdiği yük, sorunların da temelini oluşturmaktadır. Hali hazırda tam zamanlı bir işte çalışmakta olan birey, kişisel gelişimine yardımcı olan eğitim süreci için yeterli vakti ayıramamaktadır. Burada örgün eğitim olarak adlandırabileceğimiz, öğrenci – sınıf – öğretmen üçlüsünün aynı fiziksel alanı paylaştığı geleneksel eğitim modelinden farklı bir yaklaşıma ihtiyaç duyulmuştur. Bu ihtiyacı karşılayacak eğitim modelini uzaktan eğitim olarak tanımlamak mümkündür.

Uzaktan eğitimin gelişme sürecinde daha önceden de bahsettiğimiz kitaplar, bunun yanında posta servisleri, radyo ve televizyon gibi araçlar kullanılmış, eğitim kalitesi ve yöntemleri tartışılacak sistemler ortaya konmuştur. Bu sistemlere verilebilecek en güzel örneklerden biri Anadolu Üniversitesinin ön lisans ve lisans eğitimi veren Açık Öğretim Programları'dır. Yine Açık Lise olarak bilinen eğitim programları ülkemizde uzaktan eğitim ile ilgili örneklerden bir kaçıdır.

Uzaktan eğitimin istenilen eğitim kalitesine ulaşabilmesi, yeterli öğrenci – öğretmen, öğrenci – eğitim materyali etkileşiminin sağlanabilmesi, Internet'in ortaya çıkışı ve eğitim platformu olarak da Internet'in kullanılmaya başlanması ile mümkün olmuştur. Bu yeni nesil uzaktan eğitim formatı, Internet'in yaygınlaşması, ucuzlaması ve bant genişliğinin artması ile Internet'e Dayalı Uzaktan Eğitim'in ortaya çıkmasına ve

Internet'in uzaktan eğitim için uygun bir platform olarak kabul görmesine neden olmuştur.

Uzaktan Eğitim İle İlgili Çeşitli Tanımlar

Literatürde uzaktan eğitim ile ilgili yapılmış birçok tanıma rastlamak mümkündür. Genel hatlarıyla benzerlik gösteren bu tanımlamalar kimi zaman farklı yaklaşımlar da içermektedir.

Uzaktan eğitime ilişkin California Distance Learning Project (CDLP 2004) tarafından yapılan tanımda, uzaktan eğitimin öğrenciyle eğitsel kaynaklar arasında bağlantı kurularak gerçekleştirilen bir eğitim sistemi olduğu belirtilmekte, uzaktan eğitim programlarının herhangi bir eğitim kurumuna kayıtlı bulunmayan kimselere de eğitim imkanı sağlamasının son dönemde öğrencilere tanınan eğitim imkanlarını artıran bir durum olduğunun altı çizilmektedir. Uzaktan eğitim programının bir başka yönü de mevcut kaynaklardan yeterince faydalanarak gelişen teknolojiyi de yakından takip etmek zorunda olmasıdır.

United States Distance Learning Association (USDLA 2004)'ın tanımı da şu şekildedir:

"Uzaktan eğitim uydu, video, ses, grafik, bilgisayar, çoklu ortam teknolojisi gibi araçların yardımıyla, eğitimin uzaktaki öğrencilere ulaştırılmasıdır. USDLA, öğretmen ve öğrencinin birbirlerinden coğrafi olarak uzak olduğunu belirterek bu eğitim programında elektronik araçların ya da yazılı materyal ve matbu malzemelerinin kullanılması gerektiğinin altını çizer. Uzaktan eğitim; öğretmenleri içine alan öğretim ile öğrencileri içine alan öğrenim olmak üzere iki temel bölümden oluşmaktadır."

Yukarıda yapılan tanımların sayısını artırmak olanaklı olmakla birlikte, bu noktada öz bir tanım vermek kanımızca yeterli olacaktır. En basit tanımıyla uzaktan eğitim, farklı coğrafyalardaki öğrenci, öğretmen ve öğretim materyallerinin iletişim teknolojileri aracılığıyla bir araya getirildiği bir eğitim şeklidir.

1.1 e-Learning Modeli

Web Tabanlı Uzaktan Eğitim Modeli'nde, Internet'e Dayalı Uzaktan Eğitim adı altında kullanılan farklı tekniklerin hemen hemen tamamından yararlanılmaktadır. İçeriğe erişmek için HTML sayfa yapıları düzenlenmekte, iletişimin sağlanması ve sağlıklı olarak yürütülmesi için elektronik posta listelerinden faydalanılmakta, etkileşimin artırılabilmesi için tartışma listeleri ve sohbet programları kullanılmaktadır. Bu modelin en önemli avantajları arasında sanal bir kampüs yaratılabilmesi ve eşzamansız (asynchronous) eğitime olanak vermesi yer almaktadır. Öğrenciler sistem dahilindeki içeriğe istedikleri zaman ulaşabilmekte ve kaynaklardan istedikleri ölçüde faydalanabilmektedirler. Sağlanan bu esneklik, maliyet avantajları ile birleştiğinde ideal bir model oluşmasına olanak tanımaktadır.

Bu modelin günümüzde kabul görmesinin ve popülaritesinin giderek artmasının en temel nedeni eğitimin zamandan ve mekandan bağımsız oluşudur. E-Learning'in bu esnek ve bağımsız yapısı iş hayatı nedeniyle zaman sıkıntısı çeken ya da eğitimin verildiği yerde fiziki olarak bulunamayan kişiler için önemli bir tercih nedeni oluşturmuştur.

Bu modelin ön plana çıkan zaman ve mekan avantajları diğer önemli üstünlüklerini kimi zaman gölgede bırakmıştır. Bu üstünlükler sistemin tercih edilmesinde önemli roller oynayabilecek niteliktedir. Bu nitelikler kişiselleştirilebilir eğitim, öğrenci merkezli eğitim, öğrenci yönetimli eğitim ve düşük maliyetli eğitim olarak karşımıza çıkmaktadır. Kişiselleştirilebilir eğitimde, verilen eğitimin şirket, bölüm, grup hatta kişiye göre özelleştirilmesi sağlanmaktadır. Öğrenciye verilecek eğitimin, öğreticinin kapasitesine göre değil, öğrencinin gereksinimlerine göre belirlenebilmesini öğrenci merkezli eğitim sağlamaktadır. Öğrenci yönetimli eğitimin getirisi ise öğrencinin çevrimiçi topluluklar oluşturmaya olanak tanıyarak, öğrencinin kendisi için içerik ya da program oluşturabileceği bir ortam hazırlayabilmesidir. Maliyet açısından bakıldığında e-Learning modelinin maliyetinin geleneksel örgün eğitim maliyetinin ortalama yarısı kadar olduğu görülmektedir. Öte yandan e-Learning sistemlerinin etkileşimli eğitim özelliği ve güncel içerik sunması da göz ardı edilmemelidir.

e-Learning ile ilgili örnekleri incelediğimizde temel olarak iki farklı yaklaşım ortaya çıkmaktadır. Bu yaklaşımlar akademik programlar ve sertifika programları olarak ifade edilebilir. Üniversite ve özel sektör bünyesinde verilen çok çeşitli sertifika programları bulunmaktadır. Sayıca üstün olan bu sertifika programlarını lisans, ön lisans ve yüksek lisans programları izlerken, doktora programlarının sayısı diğerlerine göre bir hayli düşüktür.

Bilişimin hemen hemen her alanında olduğu gibi uzaktan eğitim konusunda da ülkemizde içerik sıkıntısı bulunmaktadır. Bant genişliğinin yeni yeni belirli standartlara ulaşabildiği ülkemizde, WTUE konusunda da yeterli ilerlemeler kaydedilememiştir. En büyük sıkıntılardan biri içeriğin yetersiz oluşudur. Yurt dışında oluşturulan zengin içerik, dil sorunu ve müfredat farklılıkları yüzünden kullanılamamaktadır. Tüm bu olumsuz tabloya rağmen sayıları çok az da olsa WTUE konusunda standartları yakalamış özel sektör kuruluşlarına ve üniversitelere rastlanmaktadır.

Türkiye’de Web sayfası aktif olarak çalışan toplam 79 üniversiteden sadece 5’i e-Learning olarak tanımlayabileceğimiz bir programa sahiptir.

Bu üniversiteler ve programları Tablo 1’de yer almaktadır.

Tablo 1. Türkiye’de Uzaktan Eğitim Programına Sahip Üniversiteler

Üniversite	Program Adı	URL
Ahmet Yesevi Üniv.	Türtep	http://www.yesevi.net
Anadolu Üniv.	E - MBA	http://emba.anadolu.edu.tr
Anadolu Üniv. Açık Öğretim Fakültesi	Bilgi Yönetimi Önlisans Programı	http://www.bilgi.aof.edu.tr
İTÜ	UZEM	http://www.uzem.itu.edu.tr
ODTÜ – IDEA	Asynchronous Internet Education	http://idea.metu.edu.tr
ODTÜ – Online	METU Online	http://online.metu.edu.tr
ODTÜ - Informatics	Informatics Online - Master of	http://ion.ii.metu.edu.tr

	Science Program	
ODTÜ – DİL	diL (Distance Interactive Learning)	http://www.dil.metu.edu.tr
İstanbul Bilgi Üniv.	E - MBA	http://www.bilgiemba.net

Üniversiteler haricinde özel sektör Uzaktan Eğitim programlarından bazılarını Tablo 2’de yer verilmektedir.

Tablo 2. Türkiye’de Uzaktan Eğitim Hizmeti Veren Özel Sektör Faaliyetlerinden Bazıları.

Kuruluş	Program Adı	URL
IDEA e-learning solutions	Microsoft Eğitimleri	http://www.ideaegitim.com
Öğretmenler Sitesi	Teknoloji Eğitimleri	http://www.ogretmenlersitesi.com
Enocta	Mesleki Eğitim	http://www.meslekegitimleri.com
Netron	E-LearnCampus	http://www.netron.com.tr

19 Mayıs 2004 tarihinde YÖK’ün web sitesinde yer alan liste dikkate alınmıştır.

1.2 Web Tabanlı Uzaktan Eğitim Sistemleri (WTUES)

WTUE’den bahsedebilmemiz için birçok fonksiyonu içinde barındıran bir eğitim yönetim sisteminin bulunması gerekmektedir. Bu bağlamda sadece ders içeriğinin elektronik ortamda bulunması yeterli değildir. Ders içeriğine Web üzerinden erişilebilmesi WTUE’nin önemli unsurlarından biridir, ancak hiçbir zaman tamamı anlamına da gelmemektedir.

Eğitim Yönetim Sistemi, ders içeriklerinin hazırlanmasından öğrenci kayıtlarının tutulmasına, sistemin kullanım saatleri ve sıklıkları gibi istatistikî bilgilerden, öğrencilerin başarı durumuna kadar birçok bilginin sistem üzerinden alınmasına ya da Web ortamına aktarılmasına olanak sağlamalıdır. Bu açıdan bakıldığında WTUES’lerin sahip olması gereken özellikler sistemin genelini de tanımlar niteliktedir.

Web Tabanlı Uzaktan Eğitim Sistemleri'nin Sahip Olması Gereken Özellikler

WTUES' lerin Web üzerindeki basit eğitim içeriklerinden ayrılabilmesi için sahip olması gereken temel özellikler vardır. Bu özellikler eğitim sisteminin amacına ve hedef kitlesine göre kimi zaman değişiklikler gösterse de genel hatlarıyla aşağıdaki fonksiyonları içermelidir

1. Kullanıcıların tanımlanması ve yönetilmesi: Geniş alan ağları, yerel ağlar ya da Internet üzerinden yayın yapan WTUES' LER genel erişime açık bir yapıya sahip olabilmektedir. Ancak eğitim içeriklerinin herkes tarafından görüntülenmesi istenmeyebilir. Belirli kullanıcı grup ve hakları doğrultusunda sisteme giriş yetkisi verilmek istendiği durumlarda WTUES' lerin kullanıcı tanımlayabilir ve yönetebilir bir yapıda olması gerekmektedir.
2. Ders içeriklerinin hazırlanması: WTUE'nin temelini oluşturan ders içeriklerinin hazırlanması ya da hazırlanmış içeriklerin Web ortamına aktarılması sistem içerisinde yapılabilir. Hazır bir şablon kullanılabileceği gibi, içeriğin oluşturulmasında farklı programları da kullanmak mümkündür.
3. Derslerin yönetilmesi: Öğrenci ders yüklerinin kontrol edilmesi, hangi dönem hangi dersi alması gerektiği ya da hangi dersi aldıkları gibi bilgilerin takip edilebilmesi gerekmektedir. Tüm bu bilgiler ışığında öğrencinin belirli bir programı takip etmesi ve bitirmesi sağlanabilir. Bu sayede sistem genelinde aktif olan derslerin kullanım yoğunluğu da takip edilmiş olmaktadır.
4. Öğrenciye özel programların açılması: WTUE'nin en önemli avantajlarından birinin esneklik olduğundan daha önce bahsetmiştik. Bu esneklik öğrenciye özel programların oluşturulabilmesiyle ön plana çıkan bir özellik haline gelmektedir. Eğitim programı zamandan bağımsız olarak tasarlanabildiğinden, dönemlik, aylık hatta haftalık ders yükleri farklı şekilde belirlenebilir. Seçmeli derslerin sınıf mevcuduna göre açılıp açılmama durumu gibi sorunlar bu sistemde yer almaz.
5. Ödev ve proje verilmesi/teslimi: Öğrencilere ödev ve projelerin verilmesi, bu çalışmalar ile ilgili içerik ve açıklamaların öğrencilere aktarılması, tamamlanan

alışmaların toplanıp deęerlendirilmesi gibi işlemlerin yapılabilmesi gerekmektedir. Tüm bu işlemlerin tek bir merkezden yapılması, sorumlu kişilerin üzerindeki iş yükünü azaltacağı gibi, sürece de hız kazandıracaktır.

6. Sınav ve testlerin hazırlanması ve uygulanması: WTUE uygulamalarında dönem içinde aktarılan bilginin öğrenci tarafından ne derecede alınabildięi ortaya konmalıdır. Bütün eğitim sistemlerinde olduęu gibi WTUE’de de bu çalışma sınav ve testler yoluyla yapılmaktadır. Bu çalışmalarda iki farklı yöntem genel olarak tercih edilmektedir. Bunlardan biri dönem/eğitim sonunda öğrencilerin bir merkezde toplanarak sınava tabi tutulmalarıdır. Bu sistem farklı ülkelerden sisteme dâhil olan kullanıcılar için uygun bir yöntem değildir. Bu durumda çevrimiçi sınavlar devreye girmektedir. Öğrenciler terminaller yardımıyla merkezden gelen soruları yanıtlamaktadırlar. İki yöntemin beraber kullanıldığı sistemler de mevcuttur. Her iki yöntemde de (ya da ikisini de uygulayan sistemlerde) eğitim süresince öğrencinin kendi bilgi düzeyini test etmesi gerekmektedir. Genel deęerlendirmede kullanılacak testlerin yanı sıra, sadece deneme amaçlı olarak testlerin oluşturulabilmesi ve bu testlerin eğitim sistemi üzerinden öğrenciye sunulabilmesi de gerekmektedir.

7. Öğrenci davranışlarının izlenmesi ve incelenmesi: WTUES’ leri başarıya taşıyacak en önemli çalışmalardan biri şüphesiz sistemin ne derece etkin kullanıldığıının gözlenebilmesidir. Bunun yolu kullanıcıların sistem içerisinde davranışlarının izlenebilmesinden geçer. Öğrencilerin günün hangi saatinde sistemden ne ölçüde yararlandıkları, hangi ders içeriklerinde ne kadar vakit geçirdikleri gibi bilgilerin sistem üzerinden takip edilebilmesi gerekmektedir. Elde edilen verilerin belirli istatistikî bilgiler halinde sorumlu kişilere aktarılması yine sistemin sorumluluğunda olmalıdır.

8. Öğrencilerin başarı durumlarının deęerlendirilmesi: Eğitimin sonunda hem sistemin başarısını, hem de öğrencinin başarısını öğrenci başarı durum deęerlendirmesi ortaya koyacaktır. Bu deęerlendirme aynı zamanda, diploma, sertifikasyon ya da başarı belgesine öğrencinin hak sahibi olup olmadığını da belirleyecektir. Başarı durumlarının deęerlendirilmesi eğitim programında daha sonraki aşamalarda ön koşulun yerine getirilip getirilmediğinin de bir göstergesi olacaktır. Tüm bu çalışmalar sistemin sorumlulukları arasında yer almaktadır.

9. Etkileşimli iletişim ortamlarının oluşturulması ve yönetilmesi: WTUE'nin önemli avantajlarından birisi de birçok değişik Internet tabanlı iletişim sistemini kendi bünyesinde barındırıyor olmasıdır. Tartışma grupları, sohbet odaları, akışkan video ve ses aktarımı, Flash gibi kullanıcı etkileşimi sağlayabilecek ara yüz teknolojilerinden en üst düzeyde fayda sağlanması, sistemin sahip olması gereken özelliklerin başında gelmelidir.

Proje teslimi şeklinde değerlendirilecek ders programlarında bu yöntemler uygulanmayabilir

1.3 e-Sınav Sistemi

Eğitimin konularından biri de, öğrencilerin edindikleri bilgi ve becerilerinin ölçülmesidir. Klasik okul/sınıf ortamlarında, öğrencilerin aynı zamanda aynı mekâna toplanmasıyla yapılan sınavlar, e-öğrenme veya uzaktan eğitim söz konusu olunca geçerliliğini ve uygulanabilirliğini yitirmektedir. Internet destekli eğitim alan öğrencilerin klasik sınavlara sokulmasının zorlukları ortadadır. Bu nedenle uzaktan eğitimi destekleyecek bir uzaktan sınav veya bir başka deyişle, e-öğrenmeyi destekleyecek bir e-sınav modeline gerek vardır.

Bu belgede, bir e-sınav sisteminin kapsamı ve gereksinimleri incelenmektedir. Bu gereksinimleri kapsayan ve yazar tarafından ASP ve Access kullanılarak geliştirilmiş bulunan E-Sınav Sistemi tanıtılacaktır.

1.3.1 İşlevsel Gereksinimler

İşlevsel gereksinimler, beş alt başlık altında toplanmıştır:

- Veri tabanının hazırlanması
- Sınav bilgilerinin, soruların ve cevapların girilmesi
- Sınavın yapılması
- Cevapların notlandırılması
- Sınav notlarının yayımlanması

E-Sınav Sistemini (ESS) kullanan tüm kullanıcılar, ister öğretmen, ister öğrenci olsunlar, sisteme girişte mutlaka geçerli bir kullanıcı kodu ve şifresi girmek zorundadırlar.

1.3.2 Veri tabanının hazırlanması

ESS'nin kullanılabilmesi için, veri tabanına şu bilgilerin yerleştirilmesi ve güncellenmesi gereklidir:

- Ders bilgileri: Dersin kodu, adı, kredisi ve diğer gerekli bilgiler.
- Öğretmen bilgileri: Öğretmenin adı, soyadı, sisteme giriş kodu ve şifresi, yetki seviyesi.
- Öğrenci bilgileri: Öğrencinin okul numarası, adı, soyadı, şifresi.
- Ders-Öğretmen ilişkisi: her sömestre için, hangi dersi hangi öğretmen(ler)in verdiği bilgisi.
- Ders-Öğrenci ilişkisi: her sömestre için hangi dersi hangi öğrencilerin aldığı bilgisi.

Bu bilgiler sınav sorularının girilebilmesi ve sınavın yapılması için gerekli bilgilerdir. ESS bu bilgilerin girilmesi ve güncellenmesi için gerekli modülleri içerir. Bu modülleri, yetkisi olan öğretmenler kullanabilirler. Örneğin, ders-öğrenci ilişkisinin kayıtlarını, organizasyonun yapısı ve sorumlulukların dağıtılmasına göre, bölüm başkanı (veya bölüm başkanının görevlendirdiği kişi) veya dersi veren öğretmenler girebilirler. ESS'de, ders-öğrenci ilişki-sinin kayıtlarının, hem bölüm başkanı, hem de öğretmenler tarafından güncellenmesine izin verilmiştir.

1.3.3 Sınav bilgilerinin, soruların ve cevapların girilmesi

Öğretmen, kendisinin verdiği derslerden herhangi biri için sınav oluşturabilir. Sınav bilgileri iki kategoride toplanır:

- Sınav özlük bilgileri
- Sınav soruları

Sınav Özlük Bilgileri:

Ders, sınavın adı, sınavın süresi, sınavın başlangıç zamanı, sınavın bitiş zamanı, kaç puan üzerinden olduğu, soru adedi, sınav ile ilgili açıklamalar.

Sınav soruları:

Sınav soruları üç tipte olabilir: (a) cevabı metin olan sorular, (b) çoktan seçmeli sorular, (c) boşluk doldurmacılı sorular. Diğer soru tipleri, uygulama geliştirme sırasında zorluklar çıkaracağı için, ESS'nin ilk sürümünde kapsam dışında bırakılmıştır. Gerek duyulursa, ESS başka soru tiplerini de kapsayacak şekilde geliştirilebilir.

Her sınav sorusu için aşağıdaki bilgiler öğretmen tarafından girilir: soru numarası, soru tipi, sorunun metni, sorunun puanı, cevabı, soru ile ilgili açıklayıcı bilgiler.

Soruların doğru cevaplarının girilmesi de bu başlık altına alınmıştır, çünkü öğretmen soruları sisteme girerken, her soru için bir kayıt yaratacaktır ve cevapların da aynı kayıt içinde, yani soru kaydında saklanması uygun görülmüştür. Bu cevaplar, öğrencilerin sınavı yaparken girdikleri cevaplar değildir. Öğrencilerin cevaplarının, veritabanında ayrı bir "cevap" tablosunda saklanması zorunludur.

1.3.4 Sınavın Yapılması

Üniversitelerde, bilgisayar laboratuvarlarında yapılan sınavlar, öğrencilerin cevaplarını kalem kullanarak kâğıda yazmaları yerine, genellikle cevaplarını bilgisayarda girip, diskete kaydederek veya bir siteye yükleme yöntemiyle öğretmene iletmeleri şeklinde yapılmaktadır. Özellikle bilgisayar programlama derslerinin laboratuvar sınavlarında, öğrenciler sınavda sorulan sorular için geliştirdikleri programları derleyip çalıştırarak, hatalarını bulup düzeltme fırsatını bulmaktadırlar ki bu da yazılım geliştirme süreci ile paraleldir. Bilgisayar ve ofis paketlerini kullanım sınavlarında, öğrencilerin ilgili yazılımları kullanmasını ne derece öğrendikleri, hazırladıkları dosyalar incelenerek belirlenir. Dolayısıyla, yazılım geliştirme ve

bilgisayar kullanımı ile ilgili sınavlarda, öğrencilerin hazırladıkları dosyalar, öğretmene ulaştırılmak durumundadır.

E-Sınav Sistemi'nin şimdiki sürümü, düz metinden oluşan cevaplar içeren her türlü sınavı yapmayı mümkün kılmaktadır.

Uzaktan sınavın bazı kurallarının olması doğaldır. E-Sınav Sistemi aşağıda tanımlanan 1. ve 2. kurala göre çalışır.

Kural-1: Öğrenciler sınavı sadece sınav kaydında bulunan "sınavın başlangıç zamanı" ile "sınavın bitiş zamanı" arasında alabilirler. Yani, sınavın başlangıç zamanından önce sınava başlayamazlar, sınavın bitiş zamanından sonra cevap kaydedemezler.

Kural-2: Her öğrencinin e-sınava başladığı an ile, en son kaydettiği cevabın kaydediliş anı arasında geçen süre, öğretmen tarafından girilen "sınav süresi"ni aşamaz.

Ayrıca, öğrenci her soruyu cevapladıkça, hangi IP adresinden ve ne zaman (tarih saat dakika saniye) cevabı gönderdiği o öğrencinin o sorunun cevabının kaydedildiği kayıta saklanır. Böylece, eğer aynı bilgisayardan aynı zaman diliminde (sınav süresi kadar bir zaman diliminde), iki öğrencinin cevapları girilmişse, bu durum saptanabilir. Bilgisayar laboratuvarında, farklı IP adresleri olan bilgisayarlar kullanılarak yapılan bir e-sınavda, bir öğrenci eğer kendi cevaplarını kaydettikten sonra, bir arkadaşının kullanıcı kodu ve şifresi ile sisteme girip, arkadaşı adına da cevaplar girmişse, bu durum saptanır. Veya öğrenci sınav süresince, farklı IP adreslerinden soruları cevaplamışsa, bu durum da saptanır.

1.3.5 Sınavın Yapılma şekli:

Öğrenciler bir web tarayıcı kullanarak, ESS'nin giriş (login) modülüne bağlanırlar. Her öğrenci kendi öğrenci numarasını ve şifresini doğru girdiğinde, bir menü görüntülenir. Menüdeki seçeneklerden biri sınav alma seçeneğidir. Öğrenci sınav alma seçeneğini tıkladığında, karşısına o anda alabileceği sınavların listesi gelir. Öğrenci bu sınavlardan birini seçer. Seçtiği sınavın ilk sorusu karşısına gelir.

Soru ekranları görüntülendiğinde, öğrenci sorunun cevabını girebilir veya başka bir soruya geçebilir. Sorunun cevabını girdiğinde, otomatik olarak bir sonraki soru karşısına gelir. Öğrenci, sorular içinde dolaşabilir, yani ekranına sınavın herhangi bir sorusunu getirebilir. Daha önce cevapladığı bir soruyu getirdiğinde, önceki girdiği cevapta ekranında belirir, cevabında değişiklik yapabilir. Cevaplar kaydedilirken, ip adresi ve zaman da kayıta eklenir. Bunun yapılmasının nedeni, öğrenci bir bilgisayardan bazı soruların cevaplarını girerken, bir başkasının, başka bir bilgisayardan öğrencinin numarası ve şifresi ile girip, diğer soruların cevaplarını girmesi durumunun saptanmasının sağlanmasıdır, yani 3. kuralın ihlal edilip edilmediğini bulmaya yöneliktir.

Öğrenci artık yeni cevap girmeyecekse veya cevaplarında değişiklik yapmayacaksa, yani bir başka deyişle, sınavı bitirdiyse, ekranında "sınavı bitir" seçeneğini tıklayarak sınavını tamamlar. Bu andan itibaren, öğrenci sınavı tekrar alıp, cevap giremez veya girdiği cevapları değiştiremez.

1.3.6 Cevapların notlandırılması

Sınavı veren öğretmen, cevapları, "öğrenci+soru" sırasında birer birer görüntüleyip, her sorunun cevabı için not girer.

Öğretmen, bir sınava ait bütün cevapları notlandırdıktan sonra, ESS menüsünde yer alan, "Sınav Notlarını Hesapla" adımını seçerek, her öğrencinin soru notlarının toplanarak sınav notlarının hesaplanmasını ve veri tabanına kaydedilmesini sağlar. Bu modül aynı zamanda, sınavın genel not ortalamasının hesaplanıp saklanmasını da sağlar.

1.3.7 Sınav notlarının yayımlanması

Sınav notları internet üzerinden iki türlü yayımlanabilir:

1. Tüm sınıfın notları listelenebilir. Bu öğretmen veya bölüm başkanı tarafından kullanılacak olan bir işlemdir.
2. Her öğrenci sisteme girip, kendi sınav notlarını görebilir.

Sınav notlarının yayımlanmasında, statik sayfalar üretilmez. Notları görmek isteyen kullanıcı (öğretmen veya öğrenci) sisteme girip notları görme seçeneğini seçtiğinde, o andaki mevcut notların listesini görür.

1.4 Diğer Gereksinimler

1.4.1. Güvenlik gereksinimleri

Kopya çekilmesini engelleyici ve fark edici unsurlar, her sınav sisteminin güvenlik gereksinimlerinin özünü oluşturur. Bir başka güvenlik unsuru da, soruların çalınmasıdır ki, bir e-sınav sisteminde bu durum veri tabanına sızıp, soruları veri tabanından çekmek şeklinde veya öğretmenin kullanıcı kodu ve şifresi ile sisteme girerek gerçekleşebilir.

"Sınavın Yapılması" kısmında belirtildiği gibi, sistemin tüm kullanıcıları geçerli kullanıcı kodu ve şifre girdikten sonra sistemi kullanabilirler.

Sistemin güvenliği aşağıdaki faktörler yoluyla artırılabilir:

- Sınavın sadece belli bir ip-adresi grubundan alınmasının sağlanması,
- Şifrelerin veri tabanında kriptolanarak saklanması,
- Soruların veri tabanında kriptolanarak saklanması, öğrenci sınavı yaparken, soruların kriptolarının çözülerek ekranlarına getirilmesi,
- cevapların girildiği bilgisayarlardan, ESS sisteminin bulunduğu server dışındaki server'lara erişimin engellenmesi, yani internet üzerinden diğer tür haberleşmelerin engellenmesi.

1.4.2. Performans gereksinimleri

Sistemin hızlı çalışması için, grafik ve ses öğelerinden kaçınılmıştır.

1.5 Sonuçlar

E-Sınav Sistemi, internete bağlı yeterli bilgisayarı olan eğitim kurumlarında kullanılabilir.

Bilgisayar ve Internet teknolojilerindeki gelişmeler, gelecekte, gerek eğitimde, gerek eğitimin sınav boyutunda, bilgisayar ve internetin kullanılmasını vazgeçilmez bir duruma getirecektir. Dolayısıyla, burada tanımlanan sisteme benzer, ancak daha gelişmiş sistemlerin kullanılması gelecekte kaçınılmaz olacaktır.

Her alanda olduğu gibi, eğitimde de bilgisayar teknolojilerinin kullanılmasına olabildiğince erken başlamak, ülkemiz eğitim sistemine katkıda bulunacaktır.

Sistemde şu geliştirmeler yapılabilir:

- Soruların grafik (resim, şekil) içermesi mümkün kılınabilir. Sorulara ses dosyaları bağlanıp, öğrenci bir soruyu ekrana getirdiğinde, bir ses dosyası hoparlörden duyurulabilir.
- Öğrenci cevap olarak dosya(lar) yükleyebilir.
- Öğrenci bir sorunun cevabını kaydetmeye çalıştığında, eğer aynı bilgisayardan (aynı ip-adresinden) başka bir öğrenciye ait cevap girilmişse, sistem cevabı kaydetmeyip, öğrenciye bir hata mesajı gönderebilir ve hatalı kullanımı sistem veri tabanına kaydedebilir.
- Cevapların notlandırılması otomatik yapılabilir. Çoktan seçmeli sorular için bu çok kolay olmasına karşın, metin cevaplı sorular ve boşluk doldurmacılı soruların cevaplarını notlandırmak, yapay zekâ uygulamaları sınıfına girmektedir ve çok daha zordur. Sistemin sonraki sürümlerinde, çoktan seçmeli soruların otomatik notlandırılması, ilk eklenecek özelliklerden birisi olacaktır.
- Sınav sırasında, öğretmen o anda ESS sistemini kullanmakta olan ve sınav alan öğrencilerin listesini alabilir.

TOTES sistemi de ESS sistemine paralel olarak onu model alarak tasarlandı.

BÖLÜM 2

2.1 Asp

ASP, ya da Active Server Pages (Etkin Sunucu Sayfaları), Microsoft tarafından geliştirilen internete yönelik sunucu taraflı (serverside) çalışan bir teknolojidir. Sunucunun sadece durağan sayfaları istemciye (ziyaretçi) göndermesi yerine, ziyaretçiden veri kabul edilmesi gerekliliğinin sonucunda ortaya çıkmıştır.

ASP sayfaları çoğunlukla VBScript kullanılarak yazılır, ancak @Language komutuyla JScript, hatta Perl gibi başka bir dil de seçilebilir.

Bir ASP dosyasının içinde, özel nesneler ve VBS, JS, SQL kodları bulunur, bu sayfalar istemci tarafından istendiğinde sunucu öncelikle ASP içindeki kodları icra ederek, istemciye göndereceği bilgiyi oluşturur ve gönderir. Gönderilen bilgi genellikle HTML (ya da SGML) şeklindedir. Fakat sadece bunlarla sınırlı değildir, aynı şekilde bir grafik dosyasıda oluşturulup, istemciye gönderilebilir.

ASP sayfaları HTML kodlarının içine gömülü şekilde oluşturulduğu halde bir kez server tarafından yorumlandığında saf HTML olarak döner. Kaynak kodlara bakıldığında ASP kodları görülmez. Bu kodlamacıların kaynaklarını saklamalarını kolaylaştırır.

ASP'nin ortaya çıkış nedenlerinden birisi de CGI dillerinin Oturum (Session) ve Uygulamaların (Application) başından sonuna kadar izlenmesinin yetersiz oluşundandır.

2.1.1 ASP'de Kullanılan Nesneler

ASP sayfalarında özel altı adet nesne kullanılır.

- Application:
- ObjectContext:
- Request:

- Response:
- Server:
- Session:

[değiştir]

2.1.2 Örnek Kodlar

```
<%
    Response.Write "Merhaba Dünya"
%>
```

Yazdırma işlemidir. HTML ile daha kolay ve daha hızlı yapıldığı halde daha önceden tanımlanmış değişkenleri çalıştırabilir.

```
<%
    degisken = "Merhaba Dünya"
    Response.Write degisken
%>
```

Değişkenler için çift tırnak (") ifadesi kullanılmaz.

Lazım olabilecek ve öğretici bir fonksiyon. Aşağıdaki kodlar yardımıyla yazınızın içerisindeki istemediğiniz kelimelerin görünmesini engelleyebilirsiniz.

```
<%
RESPONSE.REDIRECT "default.asp"
%>
```

Bu kod sayfayı farklı bir sayfaya yönlendirir.

```
<%
RESPONSE.cookies("cookieName")("yetki")="tam"
RESPONSE.cookies("cookieName")("yetki")="yok"
RESPONSE.cookies("cookieName").expires=date()+365
%>
```

bu kod sayesinde bir bilgisayara cookies(çerez) bırakabilirsiniz ama tekrar okurken okuyamazsınız onun için request.cookie kodunu kullanmalısınız. çerez: bir sistemin sizi hatırlaması için bırakmış olduğu txt dosyalarıdır. <%

```
Private Function yassak ( yokla )
    const sakınca = "chtooxlchotoxlchootoxlcthoox"
    dim ara, c
    ara = Split(sakınca, "|")
    For c = 0 to CInt(UBound(ara))
        if InStr(yokla, ara(Int(c))) <> 0 then yokla = Replace(yokla, ara(Int(c)),
            String(Len(ara(Int(c))),"*"), 1, -1, 1)
    next
    yassak = yokla
    '© chotoox
End Function

%>

<%=yassak( "chtoox - yandaki yıldızlar sakıncalı kelimeleri kamufle eder." )%>
```

2.2 Asp.Net

Çıkalı henüz beş-altı yıl olan ASP (Active Server Pages) çok büyük bir beğeni topladı ve Internet uygulamalarının çoğu ASP ile geliştirildi. Bu ilginin sebebi ise ASP ile dinamik Web siteleri ve dağıtık Web uygulamalarının geliştirilebilmesidir. XML'in öneminin artması ve gelişen teknolojiye ayak uydurabilmek için Microsoft ASP'yi geliştirmeye başladı ve bu yeni versiyona ASP+ adı verildi. Microsoft, .NET stratejisine uygun olması için bu adı daha sonradan ASP.NET olarak değiştirdi.

ASP.NET, .NET Framework'ün çok önemli bir kısmını oluşturmaktadır. ASP.NET, dinamik Web uygulamalarının geliştirilebilmesi için gerekli alt yapıyı sağlamaktadır ve Web uygulamalarının çok daha az zamanda geliştirilebilmesini sağlayacak şekilde dizayn edilmiştir.

ASP’de uygulama geliştirirken karşılaşılan büyük bir sorun sadece script dillerinin kullanılabilmesi ve bu dillerden hangisinin kullanılacağını doğru seçilmesidir. İnternet üzerinde çalışacak bir uygulama geliştiriyorsak Jscript veya JavaScript dillerinden birini kullanmak zorundayız, çünkü VBScript sadece İnternet Explorer üzerinde çalışmaktadır. Fakat İntranet üzerinde bir uygulama geliştiriyorsak VBScript kullanarak İnternet Explorer Web gezginini güçlendirmek mümkündür. ASP.NET’in ASP’den en büyük üstünlüğü dinamik Web uygulamaları geliştirirken dilden bağımsız olması ve her platform ve cihaz için çıktı üretebilmesidir. ASP.NET’de birçok dil kullanma şansınız var. Hangi programlama dilini kullanıyor olursanız olun herkes aynı objelere, özelliklere ve metotlara erişebilir. COM objelerinden farkı sadece çağırmakla kalmıyorsunuz aynı zamanda farklı bir programlama dilinde yazılmış bir sınıfı inherit (kalıtım) edebiliyorsunuz.

2.3 ASP.NET İle Gelen Yenilikler

ASP.NET bir çok açıdan eski versiyonu olan ASP’den üstündür. Aşağıda ASP.NET’in bu üstün özellikleri açıklanmıştır:

2.3.1 Daha Az Kod

ASP.NET’in üstün özelliklerinden biri, uygulama geliştirirken daha az kod gerektiriyor olmasıdır. Satır sayısında %40 ile %70 arasında azalma olur. Önceden HTML ve script kullanarak yapabildiğimiz işlerin birçoğunu artık sadece bir sunucu kontrolü koyarak yapabiliriz. Örneğin ASP’de bir metin kutusunun kullanıcı tarafından boş bırakılmaması ve sadece belirli aralıktaki sayısal verilerin girilmesi gerektiği bir durumda bir script yazarak o metin kutusunun içine bir şeyler yazılmış mı yazılmışsa belirli koşulları sağlayıp sağlamadığını kontrol ederdik. Oysa şimdi aynı işi sadece bir RequiredFieldValidator kontrolü koyarak yapabiliriz. Bir Calendar kontrolü yaratarak ekrana çok hoş bir takvim koyabiliriz. ASP.NET, kullanıcının durumunu otomatik olarak tutar ve böylece ASP’deki gibi bir kullanıcı sayfayı yenilediğinde kullanıcının durumunu tutmak için script yazmak zorunda kalmayız.

2.3.2 Uygulama ve Sunumun Ayrılması

ASP sayfaları kolay ve çabuktu fakat HTML ve onu geliştiren sunucu taraflı mantık birbirine karışıyordu. Örneğin:

```
<H2><%= “<font size=3>” & adisoyadi & “</font>” %></H2>
```

“<%=” ve “%>” şeklindeki sunucu taraflı mantık ile HTML kodları içi içe geçtiği için bu tür bir kodu değiştirmek, hatalarını ayıklamak ve devamlılığını sağlamak zordur.

ASP.NET ile sunucu taraflı Web Form ve Web servisi geliştirdiğimizde içerik ve kod birbirinden tamamen ayrılır ve farklı dosyalarda olabilir.

2.3.3 Derlenmiş Kod

ASP sayfaları yorumlanarak çalıştırılıyordu fakat ASP.NET sayfaları yorumlanmaz, derlenir. Bir ASP.NET sayfası ilk kez çalıştırıldığında bir .NET sınıfı içine derlenir. Bu sınıf ön bellekte depolanır ve sayfaya yapılacak olan sonraki isteklerde ön bellekten kullanılır. ASP.NET sayfasında bir değişiklik yapıldığında bu değişiklik fark edilir ve ilk istekte tekrar derlenir ve ön belleğe kaydedilir. Sayfa, her istek için yorumlanmak zorunda kalmadığından performansta büyük bir artışa neden olur.

2.3.4 Değişebilen Mimari

ASP’ de sunulan imkanları programcı beğenmese de aynen kullanmak zorundaydı. ASP.NET bu sorunun üstesinden gelir ve kullanıcıya çeşitli imkanlar sunar. Örneğin oturum durumuna ait ASP.NET’in sağladığı özellikleri beğenmiyorsanız silebilir ve kendi geliştirdiğiniz bir uygulamayı kullanabilirsiniz veya eksik buluyorsanız ASP.NET’in sunduğu imkanları inherit edip yeni özellikler katabilirsiniz. ASP.NET tamamen açıktır ve kullanıcı kendisine uygun olacak şekilde değiştirebilir.

2.3.5 Bileşen Kullanımı

ASP’de çoğu zaman kendi yazdığımız scriptler yetmiyordu ve uygulamalarımıza dışarıdan bileşenler eklememiz gerekiyordu. Bu hazır kodları kullanabilmemiz için de

ekstra bir efor sarf etmemiz gerekiyordu. Öncelikle bu bileşenlerin sisteme kaydının yapılması gerekiyordu ve daha sonra bu bileşenlerin daha yeni sürümlerini kullanmak istediğimizde IIS'i durdurup güncellemeyi yapıp tekrar başlatmamız gerekiyordu. Bir Web sunucusu için bu hiç istenilmeyecek bir durumdur. Bir bileşenin konfigürasyonu da tam bir dertti. ASP.NET ile kayıt, versiyon ve konfigürasyon sorunlarının çoğu ortadan kalktı. ASP.NET , ASP gibi sadece COM objelerinin sunduğu arayüzü kullanmaz. ASP.NET sayfaları COM objeleri ile eskisine göre daha uyumlu çalışırlar. Bir COM dll'inin yeni bir versiyonunu yüklediğinizde xcopy ile ASP.NET, dosyaların güncellendiği hakkında uyarılır. Eski versiyona olan isteklerin cevaplanması bitene kadar yeni versiyon yüklenir ve bundan sonraki istekler yeni versiyona göre yapılır. Önceden çok karmaşık bir yapıda IIS Metabase'i içinde saklanan konfigürasyon bilgileri artık XML formatında saklandığı için insanlar tarafından okunup anlaşılması daha kolay bir hale geldi.

2.3.6 Her Kullanıcıda Çalışır

ASP'nin istemde bulunan her tür Web tarayıcısında çalıştığı söylenirdi fakat Internet Explorer, Netscape ve diğer tarayıcıların her birinin değişik versiyonları için bunun garantisini vermek zordu çünkü, ASP sayfaları HTML kodu üretir ve değişik tarayıcılar bu kodları farklı yorumlarlar. ASP.NET'de sunucu tarafında çalışan birçok bileşen vardır ve bunlar istemde bulunan Web tarayıcısını tanırlar ve onun anlayacağı şekilde çıktı üretirler. Hatta Mobil Internet Toolkit'i kullanırsanız istemde bulunan her tür cihaza göre çıktı üretebilirsiniz.

2.3.7 İstemci Durumunun İdaresi

ASP'nin istemci durumunun idaresi için sağladığı problemlili ve yetersiz imkanları yerini çok daha fazla kullanıcıyı destekleyen, daha performanslı ve multi-threaded imkanlara bırakıyor. Örneğin bir kullanıcının durumunu SQL Server veritabanına kaydedebilmek gibi.

2.3.8 ASP İle Tam Uyum

Eskiden yazdığınız ASP kodlarını değiştirmeden kullanmak zorundaysanız hiç sorun değil, ASP.NET buna izin veriyor. ASP.NET, kullanıcılarına çok yumuşak bir geçiş imkanı tanıyor.

2.3.9 Kullanıcı Verimliliğindeki Artış

ASP.NET tamamen kullanıcının verimliliğini arttıracak şekilde dizayn edilmiştir. Dinamik Web sitelerini hazırlamak artık çok daha az zaman alıyor çünkü ASP.NET bir çok işi kendisi otomatik olarak yapıyor. Karmaşık bir çok işi sadece sürükleyip bırakarak yapabiliyoruz veya bir Web servisine referans tanımlayıp o Web servisinin sunduğu tüm imkanlardan yararlanabiliyoruz.

2.3.10 Konfigürasyon ve Mimari

ASP.NET ile konfigürasyon ve mimari çok daha güçlü bir hale getirilirken aynı zamanda çok da basitleştirilmiştir. Şimdi, bu konuda yapılan değişikliklerin neler olduğuna bakalım.

2.3.11 Konfigürasyon Dosyaları

Konfigürasyon dosyaları XML formatında olduğu için kullanıcılar tarafından okunup değiştirilebilmesi çok basit hale gelmiştir. Bu dosyaların yapıları ve içerikleri hakkında bilmemiz gereken bazı önemli noktalar vardır.

En üst seviyedeki konfigürasyon dosyası “machine.config” dosyasıdır ve <winnt>\Microsoft.NET\Framework\<version> klasörü içinde yer alır. Bir makinede çalışan tüm uygulamalar bu dosyadan etkilenirler. Diğer konfigürasyon dosyaları ise “web.config” dosyası adını taşırlar ve bu dosyalar her uygulamaya ait klasör içinde bulunurlar.

Konfigürasyon dosyalarının içerdikleri kayıtlar konfigürasyon dosyasının çeşidine göre değişir. Tüm makine için geçerli olan kayıtlar sadece “machine.config” dosyası içinde bulunur. Örneğin <processmodel>. Sadece uygulama seviyesinde geçerli olan kayıtlar ise hem <machine.config> hem de <web.config> dosyaları içinde

bulunabilirler. Örneğin <sessionstate> gibi kayıtlar. Bu kayıtlar sadece uygulama seviyesinde geçerli oldukları için daha alt seviyelere etkileri olmaz, sadece o uygulamaya ait klasör ve bu klasörün altındaki klasörler için geçerlidir. Bir uygulamaya ait klasörün alt klasörleri için de konfigürasyon dosyaları yaratılabilir. <security> güvenlik kaydı gibi kayıtlar her iki tür konfigürasyon dosyasında da bulunabilir çünkü bu tür bilgiler dosya ve klasör seviyesinde etkilidirler. Bir konfigürasyon dosyasının içeriğinin nasıl olabileceğine dair bir örnek aşağıda verilmiştir:

```
<configuration>
  <configsections>
    <!--config bölümüne ait alt elemanlar-->
  </configsections>

  <httpmodules>
    <!--http modülüne ait alt elemanlar-->
  </httpmodules>

  <httphandlers>
    <!--http tutucularına ait alt elemanlar-->
  </httphandlers>
  <sessionstate>
    <!--oturum durumuna ait alt elemanlar-->
  </sessionstate>
</configuration>
```

Aşağıda konfigürasyon dosyalarında kullanılabilecek ayarlar kısaca açıklanmıştır:

- • <appsettings> : Data Source Name gibi uygulama bilgilerini tanımlar.
- • <compilation> : ASP.NET için tüm derleme ayarlarını tutar.
- • <browsers> : Bir Web tarayıcısının yeteneklerini ayarlama kullanılır.

- • <configuration> : Tüm konfigürasyon ayarlarını tutan konfigürasyon bölümüdür.
- • <customerrors> : Kullanıcının kendisine ait olan hata mesajlarını tanımlar.
- • <configsections> : Her config bölümü ile ilgili tutucuların bir listesi.
- • <globalization> : Bir uygulama için global anlamda geçerli olan ayarları düzenlemede kullanılır.
- • <location> : Bir klasör veya dosya gibi belirli bir yere ayarları gönderir.
- • <httpmodules> : HTTP modülleri ekleyip siler ve bu modüllerin içeriğini temizler.
- • <security> : Tüm güvenlik ayarlarını tanımlar.
- • <processmodel> : IIS için işlem modeli ayarlarını tutar.
- • <trace> : ASP.NET'teki trace servisini ayarlar.
- • <sessionstate> : Port numarası, çerez bilgileri (cookies) ve zaman sınırı gibi oturum durumunu ilgilendiren bilgileri ayarlama da kullanılır.
- • <webcontrols> : Kontrollerin kullanıcıdaki yerlerini tanımlar.
- • <webservices> : Web servisi ayarlarını kontrol eder.

Konfigürasyon dosyaları XML dokümanları oldukları için bu dosyaları kullanırken iyi düzenlenmiş XML dokümanları için geçerli olan kurallara uymak gerekir.

Konfigürasyon dosyaları hiyerarşik bir yapıya sahiptir.

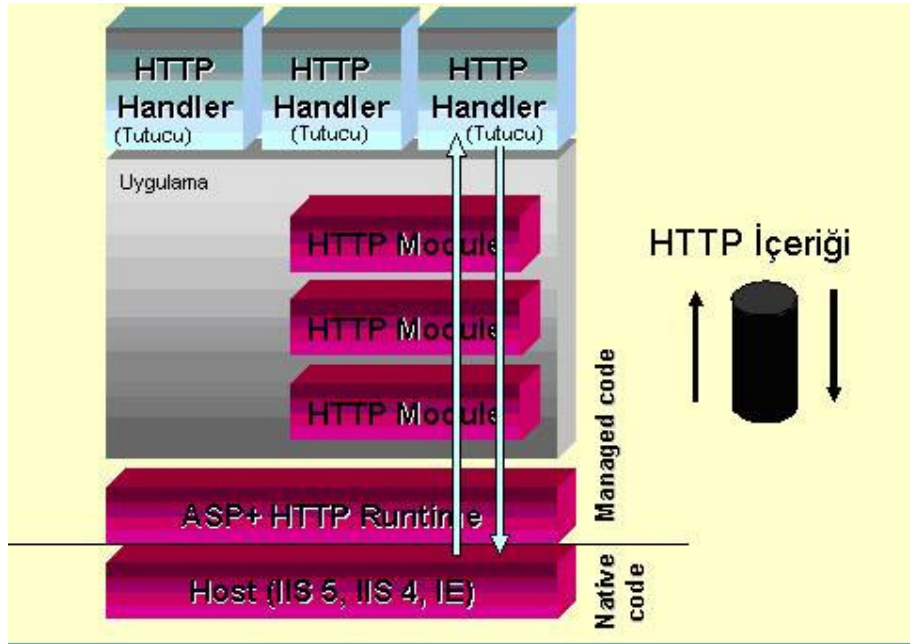
2.3.12 HTTP Runtime

HTTP Runtime (çalışma zamanı) ASP.NET'in üzerine kurulduğu, mesajlaşmayı sağlayan bir araç, bir motordur. Http Runtime, Http istemlerinin ve bu istemelere verilen yanıtların işlenmesini sağlayan alt yapıdır.

Http istemlerinden ve yanıtlarından bahsedebilmemiz için öncelikle bir bilgisayar üzerine kurulu bir hosta gereksinimimiz vardır. Bu genellikle Internet Information Server'dır. IIS, Http istemleri için 80 numaralı portu, Https istemleri için ise 443 numaralı portu dinler. Bir .aspx sayfası için gelen bir istek bir ISAPI dll'i olan XSPISAPI.dll tarafından engellenir. Bu DLL, ASP.NET çalışan işlemi olan XSPWP.exe'yi devreye sokar. Bu andan sonra yapılacak işleri bu uygulama düzenler.

2.3.13 HTTP Runtime'ın Yapısı

Hostun Internet Information Server olması şart değildir, 80 numaralı portu dinleyen başka bir uygulama da olabilir. Örneğin üçüncü parti bir Web sunucusu veya kendi yazdığınız, aynı işi gören bir uygulama veya simüle edilmiş Web isteklerini yanıtlayan bir uygulama da olabilir.



2.3.14 HTTP Handler(tutucu)

Http Handler, mevcut sistemlerde kullanılmakta olan ISAPI'lere karşılık gelmektedir. Web'den gelen bir isteğin işlenmesindeki son duraktır. İstekte bulunan sayfanın uzantısına göre ilgili tutucu o istek için ayrılır. Bir .aspx sayfasına gelen istekle

ilgilenebilecek birçok tutucu olabilir fakat bunlardan sadece bir tanesi kullanılır. Tutucular, Http modüllerinin çalışmaları bittikten sonra çalışmaya başlarlar. Web.config dosyasının içerdiği konfigürasyon ayarlarına göre bir istekle ilgilenecek olan tutucu belirlenir.

2.3.15 Http Module(modül)

Http modüllerinin mevcut sistemlerdeki karşılığı ISAPI filtreleridir. Bu modüller, her Web istemiyle birlikte çalıştırılması istenen kodları içeren sınıflardır. Örneğin, ön belleğe yükleme ve kullanıcı durumunun denetlenebilmesi gibi. Web'den gelen her ASP.NET isteği için çalıştırılacak modüllerin listesini görebilmek için machine.config dosyasının <httpmodules> bölümüne bakabilirsiniz.

2.3.16 Global.asax (uygulama konfigürasyon dosyası)

Global.asax dosyaları bir uygulama kapsamında geçerli olan kodun yazılabileceği yerlerdir. Bu dosyalar uygulamanın kök dizini içinde yer almalıdırlar. Global.asax dosyaları birçok yararlı olay için tutucu (handler) içerebilirler. Bu uygulamaların bazıları aşağıda verilmiştir:

- Application_BeginRequest
- Application_AuthenticateRequest
- Application_AuthorizeRequest
- Application_ResolveRequestCache
- Application_AcquireRequestState
- Application_PreRequestHandlerExecute
- Application_PostRequestHandlerExecute
- Application_ReleaseRequestState
- Application_UpdateRequestCache
- Application_EndRequest
- Session_OnStart
- Session_OnEnd

ASP.NET, bu olayların senkronizasyonuna ve kullanıcıların kendi tutucularını eklemelerine izin verir. Bu durum çok daha güçlü ve esnek uygulamaların geliştirilmesine olanak sağlar. Aşağıda örnek bir global.asax dosyası içeriği verilmiştir:

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Web;
using System.Web.SessionState;

namespace WebApplication4 {
public class Global : System.Web.HttpApplication
{
    protected void Application_Start(Object sender, EventArgs e)
    {
        //uygulama başladığında çalıştırılması istenen kod
    }
    protected void Session_Start(Object sender, EventArgs e)
    {
        //oturumun başladığı an çalıştırılması istenen kod
    }

    protected void Application_BeginRequest(Object sender, EventArgs e)
    {
        //uygulamaya bir istekde bulunulduğunda çalıştırılması istenen kod
    }

    protected void Application_EndRequest(Object sender, EventArgs e)
    {
        //isteğin sonlandırıldığında çalıştırılması istenen kod
    }
}
```



```

protected void Session_End(Object sender, EventArgs e)
{
    //oturum sonlandırıldığında çalıştırılması istenen kod
}

protected void Application_End(Object sender, EventArgs e)
{
    //uygulamanın bitiminde çalıştırılması istenen kod
}
}

```

2.4 Klasik ASP'den ASP.NET'e Dönüşüm

Klasik ASP uygulamalarınızı ASP.NET'e dönüştürme işlemi, bazı durumlarda zaman kaybına neden olacaktır. Bunun sebebi, ASP.NET uygulamalarının dizayn ve yapı olarak klasik ASP'den çok farklı olmasıdır. Bu yazımızda, ASP kodunuzu en efektif şekilde ASP.NET'e dönüştürmeniz için gerekli olan adımları öğreneceğiz.

Klasik ASP uygulamalarınızı ASP.NET'e dönüştürme işlemi, bazı durumlarda zaman kaybına neden olacaktır. Bunun sebebi, ASP.NET uygulamalarının dizayn ve yapı olarak klasik ASP'den çok farklı olmasıdır. Bu yazımızda, ASP kodunuzu en efektif şekilde ASP.NET'e dönüştürmeniz için gerekli olan adımları öğreneceğiz.

Klasik ASP ve ASP.NET kendilerine özel Engine'lerde işlenmektedir. Bu yüzden ASP.NET'in yeni özelliklerinden yararlanabilmek için klasik ASP sayfalarımızın **Melez ASP.NET** sayfalarına dönüştürülmesi gerekmektedir. Dönüşüme uğrayacak sayfalarımıza Melez ismini veriyorum çünkü bu sayfalar gerçek bir ASP.NET sayfası olmayacaktır. ASP sayfaları olarak kalacak fakat ASP.NET Engine tarafından işleneceklerdir.

Aşağıdaki 10 adım size fazla zaman kaybettirmeden ASP'den ASP.NET'e dönüşümde yardımcı olacaktır.

1. Sayfa yönergesini (Page Level Directive) .aspx sayfamızın en başına eklemekle başlıyoruz.

```
<%@ Page Language="vb" aspcompat="true">
```

Bu yönerge .NET motoruna sayfa içerisinde Visual Basic dilini kullanacağını bildirmektedir. Yönergenin ikinci kısmı olan `aspcompat="true"` ise, Compartment Threading kullanan eski COM bileşenleri kullanacağınızı bildirmektedir. Bu kısım seçimlidir. Fakat, sayfanızda ADO nesneleri kullanıyorsanız, bu direktifi eklemeniz gerekir.

2. Klasik ASP'de yazılmış kodu yeniden kullanmanın en popüler yolu *#include* ile ASP sayfanıza eklemenizdir. Bu kullanım Melez sayfamızda aynen kalabilir. Fakat buna alternatif yaklaşım olarak ASP.NET'de gelen Kullanıcı Kontrollerini kullanabilirsiniz. Kullanıcı kontrollerinin performans olarak daha iyi olduğunu yeri gelmişken söylemek isterim.

ASP sayfalarınız için genel bir başlığa sahipseniz, yeni Melez dosyamıza bunu aşağıdaki şekilde ekleyebilirsiniz.

```
<!--#include virtual="/includes/header.asp"-->
```

Kullanıcı Kontrollerine dönüşüm için, *Header.asp* dosyasının ismini *Header.ascx* olarak değiştirip aşağıdaki satırı sayfanın başına eklemeniz gerekir.

```
<%@ Control Language="vb" %>
```

Bu satırın eklenmesinden sonra .aspx sayfanızın içerisine aşağıdaki satırı eklemeniz gerekiyor.

```
<%@ Register TagPrefix="cc" TagName="Header"
Src="/Demo/Header.ascx" %>
```

Ayrıca #include deyimi yerine aşağıdaki deyimi sayfanız içerisinde kullanmanız gerekir.

```
<cc:Header id="Header1" runat="server" > </cc:Header>
```

3. .NET Framework, tüm değişkenlerinizi kullanmadan önce tanımlamanızı isteyecektir. Kullandığınız dilin sözdizim kurallarına uygun olarak, sayfa içerisinde değişkenleri kullanmadan önce tanımlamalısınız

4. Set anahtar sözcüğü VB dilinden kaldırılmıştır.

```
Set conn = Server.CreateObject("ADODB.Connection")
```

yerine

```
conn = Server.CreateObject("ADODB.Connection")
```

sözdizimini kullanmanız gerekir.

5. Bütün fonksiyon çağrıları parantezlerle yapılmalıdır

```
Response.Write "MsAkademik"
```

yerine

```
Response.Write ("MsAkademik")
```

kullanmalısınız.

6. Klasik ASP’de kullandığınız Server, Request, Response, Application ve Session nesneleri ASP.NET’te de desteklenmektedir. Herhangi bir değişiklik yapmadan bu nesneleri kullanmaya devam edebilirsiniz.

7. ASP sayfanızda `<% %>` tagları arasında tanımlı *Functions* ve *Sub*'lar varsa, bu yapıları `<Script Language="VBScript" runat="Server">` ile değiştirmeniz gerekir. Örneğin,

```
<%  
  
Function ReturnSomeValue(i)  
  
    ReturnSomeValue = i + 1  
  
End Function  
  
%>
```

Şeklindeki fonksiyonu aşağıdaki şekilde dönüştürüyorsunuz.

```
<SCRIPT LANGUAGE="VBScript" runat="server">  
  
Function ReturnSomeValue(i)  
  
    ReturnSomeValue = i + 1  
  
End Function  
  
</SCRIPT>
```

8. *IsNull* ve *Null* artık VB.NET'de kullanılmıyor. Null değeri ile yapılan tüm karşılaştırmalarda *System.DBNull.Value* kullanacaksınız. Örneğin,

```
if ( isnull(n) ) then ....
```

şeklindeki kod parçası

```
if ( n is System.DBNull.Value ) then ...
```

şekline dönüştürülmelidir.

9. .NET Framework ile yeni gelen bir diğer özellik ise konfigürasyon dosyalarıdır. Bunlardan Web.config dosyası uygulama bazındaki konfigürasyon değerlerini içinde barındırmaktadır. Uygulamanız için değişmeyen ve çok kullanılan değerleri bu konfigürasyon dosyasının içine yazabilirsiniz. Bu konfigürasyon dosyaları XML formatında oldukları için, herhangi bir text editor ile düzenlenebilmektedirler.

Aşağıdaki örnekte, uygulamanız için veritabanı bağlantı cümlesinin Web.config dosyasında nasıl saklanacağı gösterilmiştir.

```
<appSettings>

    <addkey="conn"                                value="server=localhost;Initial
Catalog=MsAkademikDB; userid=sa; password=akademik />

</appSettings>
```

Bu değere .aspx dosyanızda erişmek için aşağıdaki söz dizimini kullanıyorsunuz.

```
System.Configuration.ConfigurationSettings.AppSettings("conn")
```

10. Klasik ASP uygulama geliştiriciler Application Tracing işlemi için Response.Write'ı kullanıyorlardı. .NET Framework ise ayrıntılı bir tracing mekanizmasını beraberinde getirdi. Bu mekanizmayı kullanmak için sayfanın başına eklemiş olduğumuz @Page direktifine **Trace=true** değerinin eklenmesi gerekir. Trace'in kullanımı ise aşağıdaki şekildedir;

```
Trace.Write ("Sisteme giriş onaylandı")
```

Yeni Trace mekanizması ile, kodunuz son halini alıp işleyişe geçtiğinde, Trace satırlarını çıkarmak için zaman kaybetmenize gerek kalmayacaktır. Sayfa başına eklemiş olduğumuz **Trace=true** değerini **Trace=false** olarak değiştirdiğimizde, trace satırları değerlendirilmeyecektir.

Kullanmış olduğum örnek kod parçasının tamamını aşağıda bulabilirsiniz.

```

<!--##FILE           =           asp-convert.aspx##           -->
<%@   Page   Trace=False   Language="vb"   aspcompat="true"   %>
<%@   Register   TagPrefix="cc"   TagName="Header"   Src="Header.ascx"   %>
<cc:Header           id="Header1"           runat="server"           ></cc:Header>

<FORM           name="Form1"           runat=server>
<%
dim                                           conn

conn=Server.CreateObject("ADODB.Connection")   '   no   SET   keyword
conn.ConnectionString   =   GetDBConnectionString   '   test   function   Call
conn.Open dim rs rs= Server.CreateObject("ADODB.Recordset")< BR> ' no
SET                                           keyword
Response.Write("<br>")   '   use   of   ()   as   in   function   calls
rs.Open   (   "select   *   from   Orders",   oCon)

If   Not   (rs.BOF   And   rs.EOF)   Then
Do   While   Not   rs.EOF
if   (rs("orderid").Value   is   System.DBNull.Value   )   then
'                                           null   comparison
Response.Write   ("<Null>")
else
Response.Write   (rs("orderid").Value)
'   notice   explicit   use   of   .Value   property
End   If
Response.Write   ("<br>")
rs.MoveNext
Loop
End   If   %>

<SCRIPT           LANGUAGE="VBScript"           runat="server">

```

```

Function                                GetDBConnectionString
                                GetDBConnectionString                =
ConfigurationSettings.AppSettings("conn_string").ToString()
End                                function
</SCRIPT>
</form>
</body>
</html>

```

```

<!--##FILE                        =                        header.ascx##                -->
<%@                                Control                        Language="vb"                %>
<html>
<body>
<h1>This is a Header</h1> <br> <br>

```

```

<!--##FILE                        =                        Web.Config##                -->
<?xml                            version="1.0"                        encoding="utf-8"                ?>
<Configuration>
  <system.web>
    <compilation                    defaultLanguage="vb"                        debug="false"/>
                                <customErrors                        mode="Off"                        />
    <authentication                    mode=                                "Windows"></authentication>
  </system.web>
  <appSettings>
    <add    key=    "conn_string"    value="server=localhost;Initial    Catalog=
MsAkademikDB;userid=sa;password=akademik"                                />
  </appSettings>
</configuration>

```

BÖLÜM 3

UYGULAMA

TOTES internet sayfası olarak tasarlandı. Hem lokal olarak hem de internet üzerinden hizmet verebilen bir sistem olarak çalışmaktadır. Sistemi tasarlamadan önce test sınavı yapan akademisyenlerden bilgi alındı. Buna göre de uygulamanın tasarımına başlandı.

3.1 Program Analizi

Analize başlarken ilk önce test sınavlarını ve nasıl yapıldıklarını, sınav esnasında süre, soru bankası, raporlar ve sonuçlar gibi gerekli üniteler incelendi. Bu aşamada ilk önce gerekli modülleri düşündük. Ve olması gereken 3 tipimiz vardı. Yönetici, Öğretim Elemanı ve öğrenci. Bunların karşına çıkacak ekran ve elemanları aşağıdaki gibidir.

- Yönetici Ekranı:
 - Kullanıcı Listesi, Aktivasyon İşlemleri, Şifre ve Güncelleme işlemleri
- Öğretim Elemanı Ekranı:
 - Not listesi, soru Ekleme, Not güncelleme, Aktivasyon İşlemleri, Soru bankası İşlemleri
- Öğrenci Ekranı:
 - Sınavlar, Alıştırma Sınavları ve Not bilgileri

Programı düşündükten sonra bunun internet üzerinden de işlem yapabilmesi gereğiyle Asp Web programlama dilini ve aynı zamanda bazı modüllerde ileriki dönemlerde yapılacak uygulamalar için Asp.Net Teknolojisinden örnekler verdik. Veri tabanı olarak Microsoft Access programı seçildi.

3.2 Uygulama İçin kullanılan Programlar

Uygulama olarak web geliştirme dillerinden ASP kullanıldı. Kullanım kolaylığı ve esnekliği açısından Microsoft Access veritabanı seçildi. Web sunucusu olarak Internet Information Servis kullanıldı. Şimdi bu programları teker teker bir inceleyelim

3.2.1 Asp

ASP (Active Server Pages) bir web programcılığı ve elektronik ticaret (e-commerce, B2B, B2C) teknolojisidir. Günümüzde profesyonel anlamda birçok web sayfası bu teknoloji ile oluşmaktadır.

Temelde sunucu taraflı bir scripting dili tanımlaması yapılırsa da, ASP, uygulama geliştiricilerine bundan çok daha fazla imkânlar sunmaktadır. Profesyonel web tasarımcılığı anlamında kendini geliştirmek isteyen herkes mutlaka ASP teknolojisi ve yardımcı scripting dillerini öğrenmek zorundadır.

ASP programcılığında iki tür scripting dili bulunmaktadır; Client /taraflı ve server (sunucu) taraflı.

Client taraflı scriptler, web browser'ı nesne olarak ele alırlar ve alt nesneleri ile birlikte bir browser'da (web sayfalarında) programlama adına ne gerekiyor ise, onun yapılmasına imkân tanırlar. Bu script dilleri JavaScript, VBScript, Jscript'tir. En yaygın kullanılan ve bütün browser'lar tarafından işletilebilen scripting dili JavaScript'tir.

Client taraflı scriptler, kullanıcının bilgisayarında çalışırlar. Sunucu, istek geldiğinde, web sayfasını, içindeki scriptler ile birlikte ziyaretçiye gönderir ve görevini tamamlar ve ziyaretçi ile bağlantısını koparır.

Sunucu taraflı script'i farklı olarak sunucu üzerinde çalışırlar. Özel tag işaretleri arasında yazılan scriptler, sunucu üzerinde işletilir ve sonuçları ziyaretçiye iletilir.

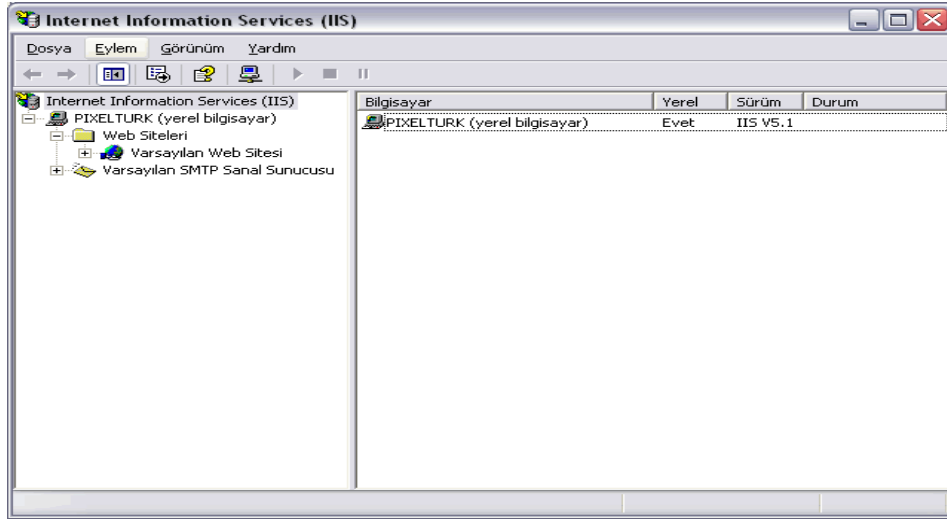
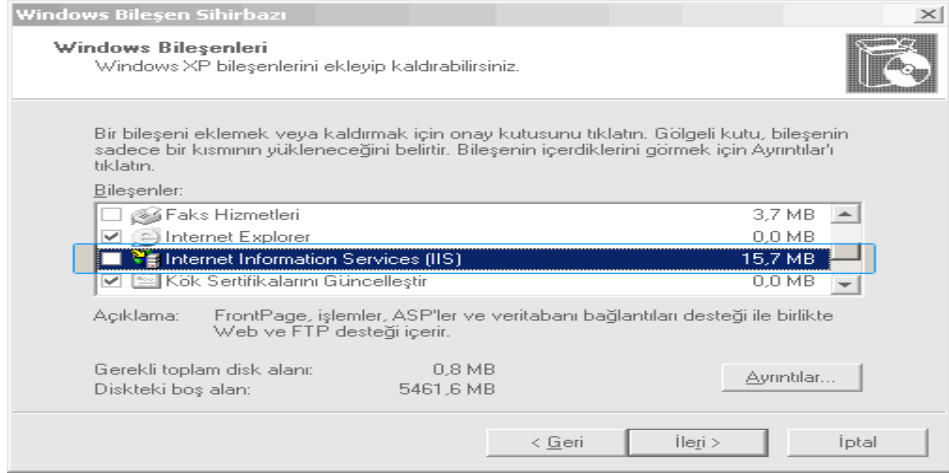
Sunucu üzerinde çalışan scriptler, ziyaretçinin bilgisayarına bilgiyi HTML olarak kodlanmış şekilde gönderir. Ziyaretçi sayfanın kodlarına baktığında, sunucu taraflı gönderilen, işlenmiş verileri HTML olarak görür. Sayfanın kodları içinde, hiçbir şekilde sunucu taraflı bir script ifadesi göremez.

Sunucu taraflı scripting, mantık olarak, kullanıcı taraflı script'in yapamadığı işler için kullanılır. Kullanıcı taraflı script ile yapabileceğimiz hiçbir şeyi, sunucu taraflı ile yapmamalıyız. Çünkü bu, hem sunucuyu gereksiz yere meşgul eder hem de hızı düşürür.

3.2.2 Internet Information Services(IIS)

ASP basit bir tanımla web üzerinde çalışan bir programlama dili diyebiliriz ve bir programlama dilinin çalışma anında derlenmesi gerekir. VBasic, Delphi gibi dillerde bu için özel compiler'lar vardır. ASP de compile işlemi web sayfasının yayınlandığı sunucu üzerinde yapılır. Compiler olarak da web servisini yöneten program kullanılır. ASP'de bu işi IIS (Internet Information Services) görmektedir. Barındırdığı sitede çağrılan bir ASP dosyasını derleyerek çıktısını kullanıcının tarayıcısına gönderir. (Win9x ve WinMe için IIS değil PWS kullanılmaktadır. Sitenizin host ettiğiniz serverda IIS yüklü olacağı için aynı ortamı oluşturmak adına IIS kullanmanızı tavsiye ederim. Eğer Win9x ve ME kullanıyorsanız mevcut işletim sisteminizi kaldırmadan da Win2000 kurabilirsiniz. XP üzerinde IIS kullanabilmek için Professional versiyonu kullanmanız gerekir.

IIS sadece Win2K ve WinXP'de çalışır. Hariçten yüklenen bir program değildir. Windows'un temel bileşenlerinden biridir. Kurulumu için Denetim Masasından Program Ekle/Kaldır'a girin. Buradan Windows bileşenlerine geçiniz ve listeden IIS'i seçiniz.



Tarayıcınıza localhost adresi girdiğinizde Varsayılan Web Sitesi olarak tanımlanmış siteyi görürsünüz. İlk kurulumdan sonra sayfanın dosyaları için tanımlanan klasör c:\inetpub\wwwroot 'dur. Çalıştırmak istediğiniz asp dosyalarınızı bu klasörün altına koymanız halinde çalıştırabilirsiniz. IIS kurulumundan sonra ASP dosyalarına çift tıklamakla bir ASP dosyasını çalıştıramazsınız. Sadece IIS üzerinde tanımlanmış klasör içindeki ASP dosyaları, tarayıcı üzerinden çağrılmaları halinde çalışabilir.

Mevcut bir sitenin kök dizinini değiştirmek için sitenin veya sanal dizinin özelliklerine girin Giriş Dizini sekmesinde siteye ait klasörü tanımlayabilirsiniz.

Şimdi bazı elemanları inceleyelim.

Site Yönetimi

IIS'in server versiyonunda birden fazla web sitesi oluşturabilirken bizim localhost olarak kullanacağımız web sitesini çalıştıran IIS'de bu özellik yoktur. Farklı web sitelerini yönetebilmesi için bir DNS sunucuya ihtiyaç duyacaktır.

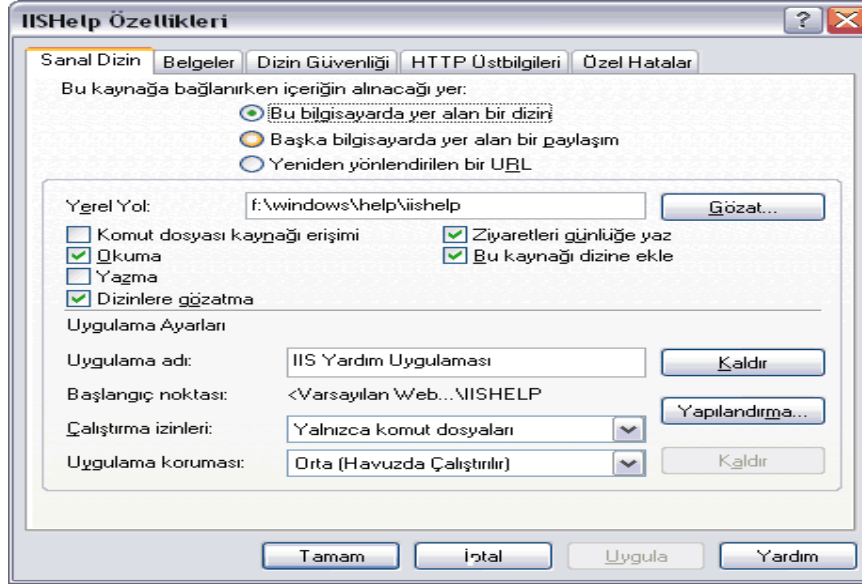
Birbirinden farklı siteleri IIS üzerinde tanımlamak için sanal dizin özelliği kullanılabilir.

SanalDizin

Sabit diskiniz üzerindeki herhangi bir dizini sanal dizin olarak atayabilirsiniz. Sanal dizinlere site altında bir dizin gibi erişilir. (<http://localhost/site>)

Erişim açısından sitenin tanımlı olduğu klasördeki dizinlerden bir farkı yoktur. Yeni sanal dizin oluşturmak için Varsayılan Web Sitesi'ne mouse ile sağ tıklayın ve açılan menüden Yeni/Sanal Dizin seçeneğini seçin. Karşınıza çıkan sihirbazda sırasıyla sanal dizinin adını ve sabit diskinizde bulunduğu yeri tanımlayınız. Bu işlemi bitirdiğinizde sanal dizininiz oluşturulmuş olacaktır.

Daha sonra bu sanal dizine ait özellikleri değiştirmek için değişiklik yapacağınız sanal dizine Mouse'un sağ tuşunu tıklayın ve özellikleri seçin.



3.2.3 Microsoft Access

Access bir veritabanı programıdır. Veritabanı, bir konuyla ilgili çok sayıda kaydın tutulduğu bir bilgi havuzu olarak nitelendirilebilir. Veritabanı programları, veritabanı denilen bilgi havuzunu oluşturmak ve veritabanından istenilen bilgiyi istenildiği şekilde alıp kullanabilmeyi sağlar.

Access programına girdiğimizde ilk gelen ekranda 3 ayrı seçenek bulunur. Yeni bir veritabanı oluşturmak istiyorsak bunlardan Boş veritabanı seçeneğini seçip Tamam düğmesine basılır. Daha sonra veritabanına bir dosya ismi vermemizi isteyen bir ekran gelir. Burada veritabanının hangi klasöre kaydedeceğimizi seçer ve dosya adı olarak da bir isim yazarız. Örneğin dosya adı olarak öğrenci ismini verelim.

Açılan veritabanında şu bölümler bulunur:

- **Tablolar:** Veritabanının temel nesnesi tablolardır. Bilgilerin asıl tutulduğu yer tablodur. Diğer veritabanı nesneleri tablolar esas alınarak oluşturulur. Bir veritabanında birden çok tablo bulunabilir.
- **Sorgular:** Tablolardaki çok sayıda kayıt içerisinde istenilen kriterlere uyan kayıtları seçerek görebilmek için oluşturulan bir nesnedir.
- **Formlar:** Tablolara doğrudan bilgi girişi yapmak daha zor ve sıkıcı olabilir. Formlar tablolara bilgi girişini kolaylaştıran ve daha anlaşılır bir ekran

görüntüsü ile çalışmayı sağlayan nesnelerdir. Paket programlardaki kullanıcı ara yüzü olarak düşünülebilir.

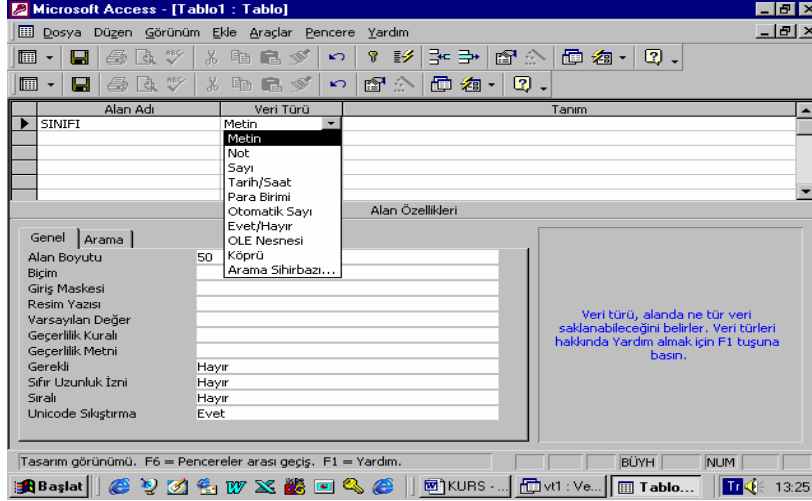
- Raporlar: Tablolardaki bilgileri kâğıda dökkebilmek için değişik şekillerde sayfa dizaynları oluşturmak için kullanılır.
- Makrolar: Veritabanında birden çok adımdan oluşan bir işlemin bir seferde yapılabilmesini sağlayan küçük program parçalarıdır.
- Modüller: Makrolar ile aynı amaca sahip olmakla birlikte Visual Basic programlama dili komutları ile yazılan küçük program parçalarıdır.

3.2.3.1 Tablo Oluşturmak



Veritabanında ilk önce tablo oluşturmak gerekir. Tablo oluşturmak için Tablolar bölümünde iken sağ taraftaki Yeni düğmesine basılır. Burada tablo oluşturma seçenekleri gelir. Tablo alanlarını ve alan özelliklerini tek tek kendimiz belirleyeceksek Tasarım görünümü seçeneği seçilerek Tamam düğmesine basılır.

Tasarım görünümünde tablo hazırlarken tablonun başlıklarını oluşturacak her bir alanın alan adını, veri türünü ve alanın özelliklerini ayrı ayrı belirleriz.



Alan adı verilirken boşluk verilebilir, Türkçe noktalı harf kullanılabilir. Alan adı en fazla 64 karakter uzunluğunda olabilir.

3.2.3.2 Veri türleri

O alana girilecek bilginin türünü belirtir. Veri türleri şunlardır:

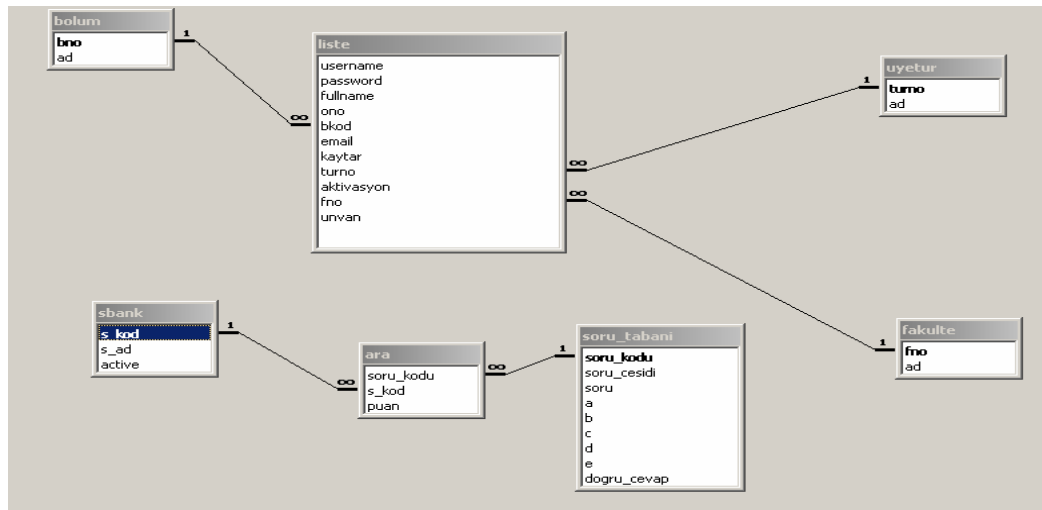
- Metin: Alfabetik ve sayısal türde her türlü bilgi girilebilen alan türüdür. En fazla 255 karakter bilgi alır.
- Not: Bir tablo alanı için metin türünün uzunluğu yetmiyorsa Not türü seçilebilir. Bu tür alanlara 65535 karaktere kadar bilgi girilebilir.
- Sayı: Tabloda üzerinde işlem yapılabilecek sayısal değerlerin girilebileceği alan türüdür. Sayısal alanların alan boyutunu belirlerken aşağıdaki boyut türleri seçilebilir:
 - Bayt: 0 ile 255 arasında değer girilebilir.
 - Tamsayı : -32768 ile 32767 arasında değer girilebilir.
 - Uzun tamsayı : -2.147.483.648 ile 2.147.483.647 arasında değer girilebilir.
 - Tek ve Çift: Çok büyük sayılar için ve ondalıklı sayılar için kullanılan veri tipleridir.
- Tarih/Saat: Tarih ve saat türündeki bilgiler için tanımlanır.

- Para birimi: Parasal değerler için tanımlanır. Bu tür alanlara 15 basamak tamsayı, ondalıktan sonra da 4 basamak bilgi girilebilir.
- Otomatik sayı: Bu tür alandaki sayı değeri girilen her kayıta otomatik olarak artar.
- Evet/Hayır: Mantıksal alan türüdür. İki ihtimalli bilgiler için kullanılır. Örneğin; Askerliğini yaptı/yapmadı, disiplin cezası var/yok gibi.
- Köprü: İnternet üzerinden bağlantı kurulacak web adresleri girilecekse bunun için köprü türü alan tanımlanır. Bu alandaki bilgiye tıklandığında eğer internet bağlantısı varsa belirtilen web adresine gidilebilir.
- OLE nesnesi: Veritabanına resim, ses, film gibi nesnelerin eklenmesi için kullanılır.

Temel olarak programların neler yaptıkları anlatıldı. Örnekler verildi. TOTES 'de bunlar nasıl kullanıldı gibi ya da veritabanı nasıl geliştirildi gibi durumları devam eden kısımda anlatılmaktadır.

3.2.3.3 Veri Tabanı Tasarımı

Access'te SORU adlı mdb uzantılı bir veri tabanı oluşturuldu. Bunun için ilişkili tablolar oluşturuldu. Aşağıda da veri tabanının ilişki şeması gösterilmektedir.



Yukarıdaki ilişkiler doğrultusunda tabloların özellikleri aşağıdaki gibidir:

Bölüm Tablosu	Ono char(50)	Active true/false
Bno integer	Bkod integer	Ara Tablosu
Ad char(50)	Email varchar(30)	Soru_kodu integer
	Kaytar datetime	S_kod integer
Fakülte Tablosu	Turno integer	Puan integer
Fno integer	Aktivasyon integer	
Ad char(50)	Fno integer	Soru_tabanı Tablosu
	Unvan char(35)	Soru_kodu integer
Konu Tablosu		Soru_cesidi char(25)
Id integer	Uyetur Tablosu	Soru char(35)
Baslik char(50)	Turno integer	A char(5)
	Ad char(35)	B char(5)
Liste Tablosu		C char(5)
Username char(50)	Sbank Tablosu	D char(5)
Password char(50)	S_kod integer	E char(5)
Fullname char(50)	S_ad char(35)	Dogru_cevapchar(5)

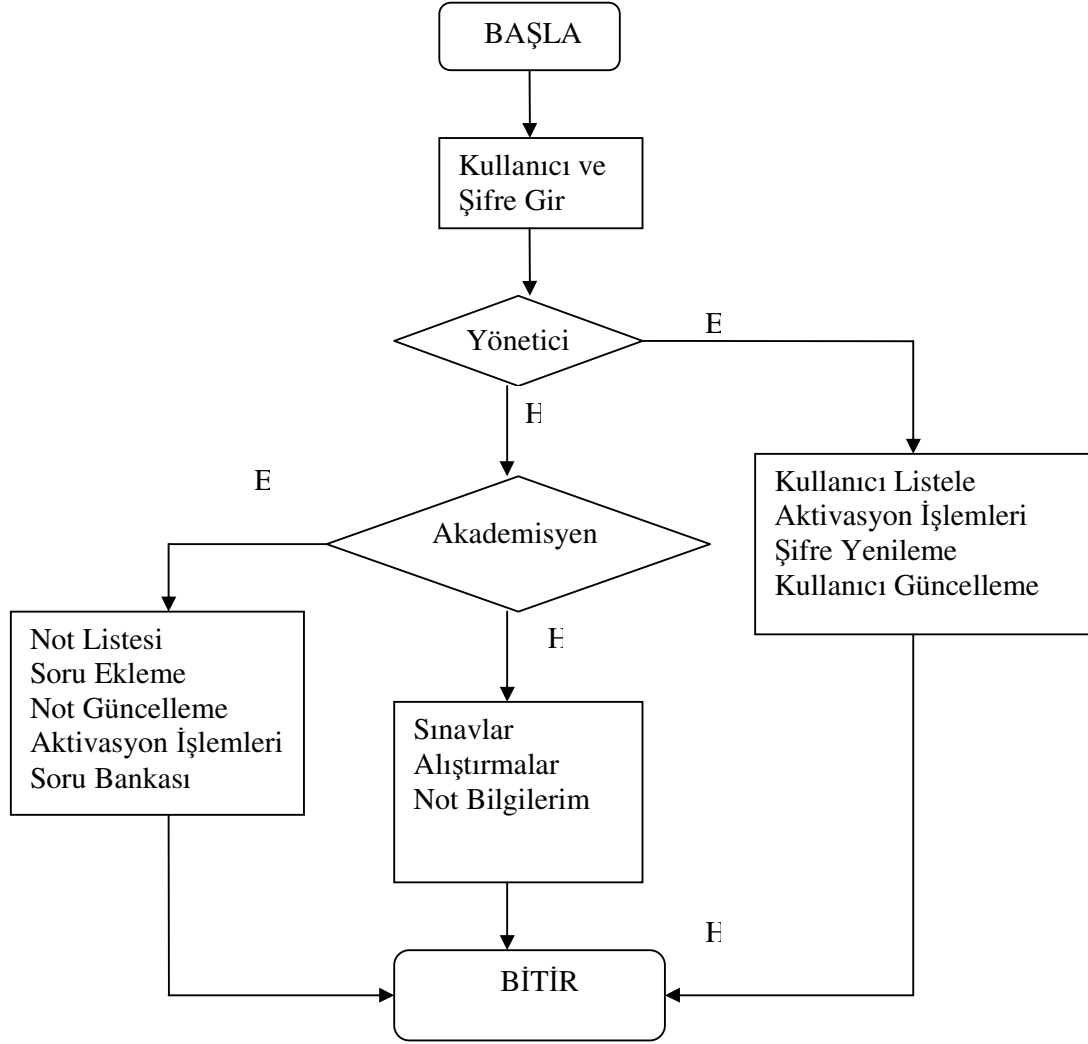
Tabloları tasarlarken ilişkisel database modeli kullanıldı. Böylece hem yerden tasarruf hemde veri tekrarlarını önleyerek sistemin daha hızlı çalışması düşünüldü. Veri sorgulama dili olarakta bir evrensel sorgulama dili olan SQL kullanıldı.

3.3 Programın akış şemaları

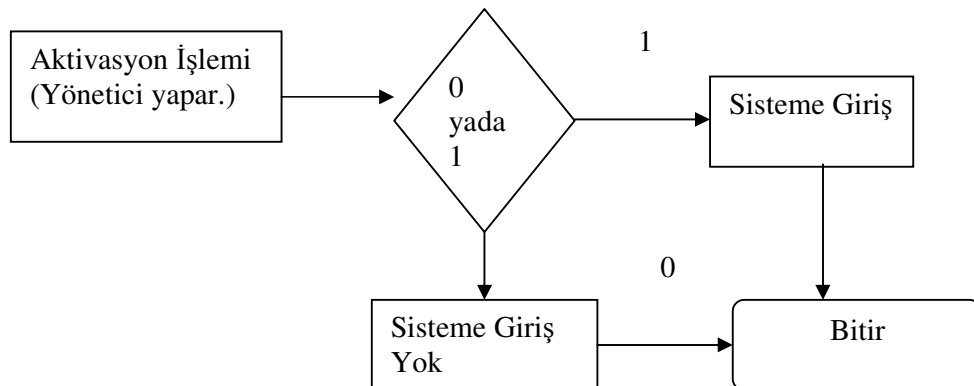
Programı tasarlarken ilk önce akış diyagramı ve algoritması çıkarıldı. Sistemde 3 tip kullanıcı var. Bunlar yönetici, öğretim elemanı ve öğrencidir. İlk giriş ekranında kullanıcı adı ve şifresini giren kişi bir aktivasyon işlemine tabi tutulur. Yönetici öğretim elemanlarının sistemde yetkili olmalarını sağlarken, öğretim elemanları da öğrencilerin sisteme girişlerinde aktivasyon işlemini gerçekleştirirler. Aktivasyon işlemi onay verilirse true, onay verilmezse false değer alarak sisteme giriş yada izin vermez. Sisteme giriş yapıldığında aşağıdaki akış diyagramında da görüldüğü gibi ekranlarında yapabilecekleri maddeler sıralanmıştır. Öğrenciler hem alıştırma sınavları ile sınavlara kendileri hazırlanabilmekte ve aynı zamanda akademisyenlerin soru bankaları ile online sınav olabilmekteler. Not bilgilerine bakabilmekteler. Akademisyen ise sorular ekleyip, soru bankaları oluşturup, öğrencilerin bilgilerini ve notlarını güncelleyebilirler. Yöneticiler ise sistemin başında giriş işlemleri,

güncelleme işlemleri gibi olaylara bakarlar. Böylece sistem üzerindeki hatalarda müdahale gerekirse, bunu sistemin başından halledebilirler.

Algoritma1: Sisteme Giriş ve İşlemler



Algoritma2: Aktivasyon işlemi



3.4 Programın Çalışması :



Giriş Ekranı

Tasarım Freehand ve Fotoshop kullanılarak yapıldı.



Giriş Ekranı-2

Yukarıdaki ekranda kayıt olan kullanıcılar sisteme dahil oluyor.

Yönetici ve Aktivasyon İşlemi Ekranı

Yönetici aktivasyon işlemi sayesinde bilgilerini kontrol ettiği kullanıcılara sisteme giriş için yetki veriyor.

Soru bankası	Onay Durumu	Onaylama İşlemi
vize2006Tekstil	Sınav başladı	Sınavı Başlat
2006vizeC	Sınav başladı	Sınavı Başlat
btp2006	Sınav başladı	Sınavı Başlat

Öğretim Elemanı Ekranı (Soru bankası Oluşturma)

Öğretim Elemanı sınav için soru bankası hazırlar. Sınavı başlat linkini tıkladığında öğrenci ekranında artık sınav başlamıştır.

TOTES
TRAKYA ÜNİVERSİTESİ ONLINE TEST VE SINAV SİSTE

Merhaba, dr
Aktif Kullanıcı: 1
ANASAYFA ÇIKIŞ

Network Word Excel **Donanım** Bilgisayar Kültürü Outlook

Konu :Donanım

Sorunuzu giriniz:

Şıklarınızı giriniz :

a)

b)

c)

d)

e)

Doğru cevabı giriniz.(a,b,c gibi):

Öğretim Elemanı Soru ekleme Ekranı

Öğretim elemanı soru bankasına sorular ekleyebiliyor. Aynı zamanda notları güncelleyebiliyorlar, öğrencilerin listelerini takip edebiliyor.

TOTES
TRAKYA ÜNİVERSİTESİ ONLINE TEST VE SINAV SİSTE

Merhaba, coacxiel
Aktif Kullanıcı: 1
ANASAYFA ÇIKIŞ

Adı-Soyadı: deniz gezzin
Numara : coacxiel
Tarih : 5/2/2006
İmza :

1 Aşağıdakilerden hangisi ağ protokollerinin özelliklerinden değildir?

A) Verilerin nasıl paketleneceğini belirtmek

B) Verilerin nasıl kullanılacağını belirtmek

C) Verilerin nasıl iletileceğini belirtmek

D) Bilgisayarlar arası bağlantıları düzenlemek

E) Verilerin ne zaman yollanacağını bilmek

2 TOPLA (B3:B12) fonksiyonunun işlevi aşağıdakilerden hangisidir?

A) B3 hücresinin içeriğini , B12 hücresine yazar.

B) B3 ile B12 hücresindeki değerleri toplar.

C) B3 ve B12 hücrelerindeki değerleri toplar.

D) B3'ten B12 'ye kadar olan hücrelerdeki değerleri toplar.

E) B12 hücresindeki değeri B3'e toplayarak yazar.

Öğrenci Sınav Ekranı

Öğrenci sınavını seçerek sınava başlatıyor. Tüm öğrencilerin ekranına sorular kopyayı bir şekilde engellemek için karışık (random) bir şekilde dağılıyor. Öğrenciler bunun yanında alıştırma soruları da çözebiliyor. No bilgilerine bakabiliyor.

Sonuç olarak sistem test sınavlarında kolaylıklar sağlayan, internetten de erişilebilen bir sistemdir.

BÖLÜM 4

TARTIŞMA ve SONUÇ

Türkiye’de, özel sektör olsun, üniversiteler olsun uzaktan eğitim sistemi kullanılmaktadır. Çünkü insanların zaman ve mekân için yeterli ya da uygun vakti olmayabilir. Bunun için sertifika programları ve üniversitelerden diploma almak için vakti olmayan kişiler birinci bölümde örnek verildiği gibi Açık öğretim fakülteleri aynı zamanda Netron gibi bazı sertifika veren özel kuruluşlar eğitimden sonra bir uzaktan eğitim modülü olan e-sınav sistemleri ile öğrencileri sınav yapmaktadırlar. Bunun için e-sınav sistemler çok önemli sistemlerdir.

Aslında bizim ele aldığımız model daha çok üniversite içersinde yerel olarak yapılacak test sınavlarının değerlendirilmesi üzerine kurulmuş bir yapıdadır. Böylece okullardaki gözetmenlik sistemlerinde güçlüklerin özellikle kopya çekilmesinin önüne geçilir olması aynı zamanda internetten de bağlanıp alıştırma sorularını çözüp sınavlara kendilerini hazırlamış olmaktadır.

E-sınav olarak hazırladığımız TOTES’in ileriki dönemlerde daha da geliştirilip tüm üniversitenin bölümlerinde kullanılacak bir ünite olması sağlanacaktır. Pilot olarak Edirne Meslek Yüksekokulu seçilmiş olup, vize sınavlarında kullanılmıştır.

TOTES sayesinde zamanda tasarruf ve akademisyenlerin evlerinden yada internetin olduğu her yerden sorularını yükleyip sınav yapabilmelerini, öğrencilerin sınavlara hazırlanabilmelerini ve sınavların değerlendirilme süresinin kısaltılıp daha şeffaf sonuçların elde edilmesi sağlanmaktadır.

TOTES’i ilerideki e-sınav tekniklerine yol gösterici bir çalışma olarak değerlendirebiliriz. E-Learning kapsamında bir modül kullanılarak ileriki dönemlerde bölüm 1’de anlatıldığı gibi tüm modüller toplanarak tam anlamıyla bir e-Learning sistem yapılabilir.

KAYNAKLAR

- [1] Halil Özmen, E-Sınav Sistemi(ESS) ,2004
- [2] Orçun Mardan, Web Tabanlı Uzaktan Eğitim Sistemleri(WTUES),2002
- [3] David S. Platt, İşte Microsoft. Net, 2002
- [4] Zafer Demirkol, Asp.Net, 2004
- [5] Mithat Uysal, Sql Veri tabanı Sorgulama Dili ve Asp Uygulamaları, 2004
- [6] Ünal Yarımağan, Veri tabanı Sistemleri, 2000

TEŞEKKÜR

Bu çalışmanın hazırlanmasında bana yol gösteren, destek ve yardımlarını benden esirgemeyen danışman hocam Yrd. Doç. Dr. Erdem UÇAR'A, tasarım işleminde yardımını esirgemeyen ve destek veren Öğretim Görevlisi arkadaşım Elçin DOKSATLI' YA, program esansında yorumlarını ve desteklerini esirgemeyen arkadaşlarım Hakan GÜLDAL ve Erdiñ UZUN'a ve Aileme teşekkürlerimi sunarım.

ÖZGEÇMİŞ

Deniz Mertkan GEZGİN, 1978 yılında Kütahya’da doğdu. 1995 yılında İzmir Selçuk Lisesini bitirdikten sonra aynı sene Çanakkale 18 Mart Üniversitesi Mühendislik – Mimarlık Fakültesi Bilgisayar Mühendisliği bölümüne başladı. 1999 yılında mezun olduktan sonra İzmir’de Retina Göz Merkezi’nde Network Administrator olarak çalıştı. 2001 yılında Askerlik görevimi yaptıktan sonra halen çalıştığı Trakya Üniversitesi Edirne Meslek Yüksekokulunun Bilgisayar Teknolojisi ve Programlama Bölümünde öğretim görevlisi olarak göreve başladı. Yüksek lisans tez aşamasındadır.