

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ÖĞRENCİ İŞLERİ OTOMASYONU İÇİN WEB TABANLI
PROGRAM GELİŞTİRME**

Tuncay Yavuz ÖZDEMİR

Tez Yöneticisi
Yrd. Doç. Dr. İbrahim TÜRKOĞLU

YÜKSEK LİSANS TEZİ
ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ ANABİLİM DALI

ELAZIĞ, 2006

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ÖĞRENCİ İŞLERİ OTOMASYONU İÇİN WEB TABANLI
PROGRAM GELİŞTİRME**

Tuncay Yavuz ÖZDEMİR

YÜKSEKLİSANS TEZİ
ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ
ANABİLİM DALI

Bu tez, 19/01/2006 tarihinde aşağıda belirtilen jüri tarafından oybirliği /oyçokluğu ile başarılı / başarısız olarak değerlendirilmiştir.

Danışman: Yrd. Doç. Dr. İbrahim TÜRKOĞLU

Üye: Doç. Dr. Z. Hakan AKPOLAT

Üye: Yrd. Doç. Dr. Ali KARCI

Üye:

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun/...../..... tarih ve sayılı kararıyla onaylanmıştır.

TEŞEKKÜR

Beni bu konuya yönlendiren, çalışmalarım süresince hiçbir konuda yardımlarını esirgemeyen danışman hocam Sayın Yrd. Doç. Dr. İbrahim TÜRKOĞLU'na ve manevi destekleriyle sürekli beni destekleyen mesai arkadaşlarım Sayın Murat DEMİRKOL, Murat KARABATAK, Abdülkadir ŞENGÜR ve Yalın Kılıç TÜREL'e ve maddi, manevi her türlü desteği sağlayan, çok sevdiğim ve değer verdiğim aileme teşekkürü bir borç bilirim.

Tuncay Yavuz ÖZDEMİR

İÇİNDEKİLER

Sayfa

TEŞEKKÜR

İÇİNDEKİLERI

ŞEKİLLER LİSTESİ.....III

TABLolar LİSTESİ.....V

KISALTMALAR LİSTESİ.....VI

ÖZETVII

ABSTRACT.....VIII

1. GİRİŞ	1
1.1 Problemin Tanımlanması.....	2
1.2 Sistemin Yapısı	3
1.3 Tezin Organizasyonu	3
2. WEB TABANLI PROGRAMLAMA TEKNİKLERİ	5
2.1 CGI.....	6
2.1.1 CGI Programlamanın Özellikleri.....	7
2.2 ASP	10
2.2.1 ASP Programlamanın Özellikleri	12
2.3 PHP	14
2.3.1 PHP Programlamanın Özellikleri	15
2.4 Karşılaştırma	19
3. ÖĞRENCİ İŞLERİNİN ANA UNSURLARI	22
3.1 Öğrenci Bilgileri	22
3.2 Devamsızlık Bilgileri	22
3.3 Listeler	23
3.4 Yıl Sonu İşlemleri	26
3.5 Mezuniyet ve Diploma.....	26
3.6 İstatistikler	27
3.7 Sınavlar	28
4. GELİŞTİRİLEN OTOMASYON SİSTEMİNİN YAPISI	29
4.1 Yönetici Sayfası	29
4.1.1 Şifre İşlemleri	32

4.1.2 Kullanıcı İşlemleri	32
4.1.3 Kullanıcı Tanımlama	32
4.1.4 Ders Tanımlama.....	33
4.2 Öğretmen Sayfası.....	34
4.2.1 Not Girişi	35
4.3 Öğrenci Sayfası.....	36
4.3.1 Kişisel Sayfam	37
5. GELİŞTİRİLEN OTOMASYON PROGRAMININ KULLANIMI.....	38
5.1 Kullanıcı Sınıfları	38
5.1.1 Yönetici Olan Kullanıcılar.....	39
5.1.2 Öğretmen Kullanıcılar	51
5.1.3 Öğrenci Kullanıcılar	59
6. SONUÇ.....	64
7. KAYNAKLAR	66
8. ÖZGEÇMİŞ.....	68

ŞEKİLLER LİSTESİ

Şekil 2.1 İstemci Sunucu çalışması.....	8
Şekil 2.2 Sunucuda Taleplerin Yorumlanması	9
Şekil 2.3 Giriş Sayfası Yüklenirken CPU Kullanım Oranı	20
Şekil 2.4 Üye Girişi Yapılırken CPU Kullanım Oranı	20
Şekil 2.5 Yeni Üye Kaydı Yapılırken CPU Kullanım Oranı	21
Şekil 4.1 Öğrenci İşleri Otomasyonu Giriş Sayfası	29
Şekil 4.2 Yönetici Sayfası	30
Şekil 4.3 Öğretmen Sayfası	34
Şekil 4.4 Öğrenci Sayfası	36
Şekil 5.1 Otomasyon Ana Sayfası	39
Şekil 5.2 Yönetici Sayfası	40
Şekil 5.3 Şifre Değiştirme Ekranı	40
Şekil 5.4 Kullanıcı İşlemleri Penceresi	41
Şekil 5.5 Kullanıcı Şifresi Değiştirme Penceresi.....	41
Şekil 5.6 Kullanıcı Adı Değiştirme Penceresi	42
Şekil 5.7 Kullanıcı Silme Penceresi.....	43
Şekil 5.8 Kullanıcı Ekleme Ekranı	43
Şekil 5.9 Yılsonu Ders Ortalamaları	44
Şekil 5.10 Devamsızlık İşlemleri	44
Şekil 5.11 Devamsızlık Girişi	45
Şekil 5.12 Sınıf Bazında Devamsızlık Sorgulama	46
Şekil 5.13 Öğrenci Bazında Devamsızlık Sorgulama	46
Şekil 5.14 Öğrenci Bazında Not Sorgulama	47
Şekil 5.15 İstatistik İşlemleri	47
Şekil 5.16 Şube ve Öğrenci Sayıları	48
Şekil 5.17 Yaş Analizi	48
Şekil 5.18 Yeni Kayıt Olan Öğrenciler Listesi	49
Şekil 5.19 Ders Yetkilendirme Modülü.....	49
Şekil 5.20 Listeler	50
Şekil 5.21 Sınıf Listeleri	50

Şekil 5.22 Bölüm Listeleri	51
Şekil 5.23 Öğretmen Sayfası.....	52
Şekil 5.24 Öğretmen Kişisel Bilgileri.....	53
Şekil 5.25 Öğretmen Kişisel Bilgileri Bilgi Güncelleme Ekranı	53
Şekil 5.26 Öğretmen Şifre Değiştirme Ekranı	54
Şekil 5.27 Yöneticiye Mesaj Gönderme Ekranı	54
Şekil 5.28 Yılsonu Ders Ortalamaları	55
Şekil 5.29 Sınıf Not Bilgileri	56
Şekil 5.30 Sınıf Listeleri	57
Şekil 5.31 Devamsızlık Sorgulama Ekranı-1	57
Şekil 5.32 Devamsızlık Sorgulama Ekranı-2	58
Şekil 5.33 Devamsızlık Sorgulama Ekranı-3	58
Şekil 5.34 Öğretmen Not Giriş Ekranı	59
Şekil 5.35 Öğrenci Giriş Sayfası	60
Şekil 5.36 Öğrenci Kişisel Bilgiler Penceresi	61
Şekil 5.37 Yöneticiye Mesaj Gönderme Ekranı	62
Şekil 5.38 Öğrenci Şifre Değiştirme Ekranı	63

TABLÖLAR LİSTESİ

Tablo 2.1 PHP 3.0.6 – PHP / Zend Performans Karşılaştırması	18
Tablo 2.2 ASP – PHP / Zend Performans Karşılaştırılması	19
Tablo 2.3 Ortalama CPU Kullanımı (%)	21

KISALTMALAR LİSTESİ

WWW	: World Wide Web
HTTP	: Hyper Text Transfer Protocol
HTML	: Hiper Text Markup Language
CGI	: Common Gateway Interface (Ortak Giriş Ara yüzü)
ASP	: Active Server Page
API	:Uygulama Programı Ara yüzünün
ISAPI	:Internet Sunucu Uygulama Programı Ara Yüzü
ÖSYM	: Öğrenci Seçme ve Yerleştirme Merkezi
IIS	: Internet Information Server
PHP	: Personel Home Page-Hypertext Preprocessor

ÖZET

ÖĞRENCİ İŞLERİ OTOMASYONU İÇİN WEB TABANLI PROGRAM GELİŞTİRME

Tuncay Yavuz ÖZDEMİR

Fırat Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik ve Bilgisayar Eğitimi Anabilim Dalı

2006, Sayfa:68

Bu tez çalışmasında, okullarda yoğun bir şekilde yer teşkil eden öğrenci işleri için web üzerinde çalışabilecek bir otomasyon geliştirilmiştir. Bunun öncesinde yaygın olarak kullanılan Web programlama dillerinden ASP, PHP ve CGI incelenerek bu üç dilin performans analizi yapılmıştır. Yapılan bu performans analizi sonucu olarak en iyi performansı PHP'nin sergilediği görülmüştür. Bu bilgiler eşliğinde öğrenci işleri otomasyonu için PHP web programlama dili seçilmiştir. Program içerisinde tanımlanan üç farklı kullanıcı gurubu vardır ve her kullanıcı gurubunun farklı yetkileri bulunmaktadır.

Anahtar Kelimeler: Web programlama, Performans analizi, Otomasyon, Öğrenci İşleri

ABSTRACT

WEB BASED PROGRAM DESIGNING FOR STUDENT AFFAIRS AUTOMATION

Tuncay Yavuz ÖZDEMİR

Firat University
Graduate School of Natural and Applied Sciences
Department of Electronics and Computer Science

2006, Page:68

In this thesis, the student affairs in high schools which is troublesome for the employees is overcome with web based automation software. Before starting thesis, we examined ASP, PHP and CGI web based programming languages and we compared the performance analysis of these languages which are mostly used. According to the results of the performance analysis, PHP is seen to the best. Thus, we chose PHP for coding the automation software. There are three different user groups and each of this group has different authorizations.

Keywords: Web programming, Performance analysis, Automation, Student Affairs

1. GİRİŞ

Bilim ve teknolojinin buluştuğu yerler bilgisayar ve benzeri ortamlardır. İnsanların temel bilgisayar bilgilerini öğrenmeleri ve bu bilgileri modern yaşamda kullanmaları kendilerinin olduğu gibi gelecek nesillerin de yönlendirilmesi, onlara meşale tutması açısından önemlidir. Bu çağdaş gelişmelere paralel olarak bilgisayarlar bankalarda, sağlık sektöründe, sanayi alanında, iletişimde, eğitim alanında özel tüm sektörlerde, sosyal yaşamda, otomasyon dünyası vb. alanlarda etkin olarak karşımıza çıkmaktadır. Bilgisayar sadece bilgi depolama ve hesap yapma özelliğini çok çabuk aşarak insan yaşantısının bir kesiti haline gelmiş ve eğitim ortamı özelliğini kazanmıştır.

İnternet'in hızla büyüdüğü ve yaygınlaştığı günümüzde istenilen bilgiye erişim kadar bilginin sunuş şekli de giderek önem kazanarak en iyi şekilde biçimlenmesi yolunda çalışmalar hızla yapılmaktadır. Web Tabanlı öğretim, uzaktan eğitim ve örgün eğitimde kullanılan yenilikçi bir yaklaşımdır. Web üzerinden eğitim ortamlarında WWW'nin özellikleri, kullanılabilirliği öğrenmenin desteklendiği anlamlı, olumlu bir öğrenme ortamı oluşturmak için kullanılabilir. WWW grafik tabanlı olduğu için kullanılması basittir. İnternet'e bağlı olan bilgisayarlarda doküman, resim, animasyon, müzik, video görüntülerinin birleştiği sayfalar içeren web tabanlı sayfalar hizmet veren materyal olmakla birlikte aynı zamanda eğitim ortamıdır.

Eğitimde bilgisayar kullanımı bilgiye ulaşım ve bilgilerin iletimi konusunda büyük kolaylıklar sağlayacaktır. Dolayısıyla bu sistemlerin en etkili olarak kullanımı; ancak bilgili, teknolojiyi kullanma konusunda iyi yetiştirilmiş eğitimcilerin yol gösterici rolü oynadığı eğitim-öğretim ortamlarının oluşturulmasıyla mümkün olabilecektir.

Web tabanlı öğretim, uzaktan eğitim, bilgisayar destekli eğitim ve internet teknolojilerinin gelişmesiyle ortaya çıkmış bir teknolojidir. Web üzerinde uzaktan eğitimin desteklenmesi şeklinde hazırlanmış zamandan ve mekândan bağımsız olarak yürütüldüğü kişisel veya kurum sayfalarına ve bu sayfaların öğretim ve sunum aracı olarak oluşturduğu eyleme "Web tabanlı eğitim" denilmektedir [1]. Bilgisayarların (internet ağ sistemleri kullanılarak) desteği ile öğrenme ve öğretme etkinlikleri ve hizmetinin sunulmasıdır. Bunun sonucunda da geleneksel sınıf ortamından farklı bir yapılanmayı gerektirir. Zaman ve mekânı belli olan öğreticinin merkezde olduğu, var olan bilginin aktarılmasına dayalı olarak yapılandırılan geleneksel sınıf ortamından farklı olarak web tabanlı öğretim ortamında ise öğretici bilginin tek kaynağı değildir. Senkron (eşzamanlı) ya da asenkron (eşzamansız) olarak eğitim öğretim faaliyetlerine katılması öğrenmede esneklik kazandırır. Öğrenciler istedikleri zaman ve

mekânda eğitime katılabilirler. Bu da onlara özgür bir şekilde kendilerini rahat hissedebilecekleri ve ifade edebilecekleri bir ortam sağlar.

Veritabanlarını yedeklemek ya da tutarlılık denemeleri yapmak gibi çoğu bakım ile ilgili birçok veritabanı yönetim görevi otomatikleştirilebilir. Otomasyon, verimliliği artırır, bilgisayar başında olmadan veritabanı işlemlerinin yerine getirilmesini ve daha fazla işlemlerin yapılmasını sağlar [2]. Sunucuyu, ilgili olay gerçekleştiğinde işlemleri ve kullanıcı etkinliklerini izleyecek, hataları denetleyecek ve gerekli uyarılar alınacak şekilde yapılandırılabilir. Bu uyarılar doğru biçimde yapılmışsa kullanılan veritabanı kendini izleyebilir ve kullanıcıya diğer yönetim görevlerini yerine getirmek için zaman kazandırır. Sürekli yenilenen yönetim görevlerini otomatikleştirmek için bu görevler zamanlanabilir. Bu görevler yalnızca bir defa yapılabileceği gibi istenilen periyotlarda tekrarlanmak üzere yapılandırılabilir.

1.1 Problemin Tanımlanması

Başlangıçta basit matematiksel işlemlerin yapılabildiği elektronik bir alet olan bilgisayar; artık günümüzde her türlü işlemlerin hızlı, pratik ve daha ekonomik bir platformda yapılmasını sağlayan teknolojik bir alete dönüşmüştür. Bu bağlamda okullarda daha çok idareciler tarafından yapılması zorunlu işleri şunlardır. Her gün öğrencilerin devamsızlığı belirli bir sistematığa göre işlenecek, devamsızlığı fazla olan öğrencilerin velilerine uyarı amaçlı mektuplar gönderilecek, dönem sonlarında tüm öğretmenlerin öğrenci başarı değerlendirme sonucu olan notlar gene belirli bir sistematığa göre işlenecek ve dönem sonunda bu notlar ve devamsızlık bilgileri öğrenci karnelerine işlenecektir. Tabi öğrenciler karne almadan önce sürekli idarecilerden devamsızlık bilgilerini, öğretmenlerinden not bilgilerini öğrenmek isteyeceklerdir. Bununla birlikte duyarlı veliler; öğrencilerinin okuldaki başarı ve devamsızlık durumlarını ve okuldaki genel durumunu öğrenmek için belirli aralıklarla idarecileri ve öğretmenleri ziyaret edeceklerdir. Dikkat edilecek olursa aynı bilgi etrafında sürekli öğrenci, veli, öğretmen ve idareci dörtlüsü birbiriyle temas halinde olmak durumunda kalacaktır. Tabi buda hem zaman ve hem de işgücü kaybına yol açacaktır. Tüm bu bilgileri sadece klasik bir şekilde dosyalara işlemekle yetinmeyip bilgisayara işlendiğini ve bu bilgilere zaman ve mekândan bağımsız olmak kaydıyla internetten erişme imkânının olduğu durumda; yukarıda bahsedilen her türlü zaman ve işgücü kaybının giderilebileceği aşikârdır. Öğretmen öğrenci notlarını, dersteki tutum ve davranışlarını, idareci öğrenci devamsızlığını web ortamında girecek, veli ve öğrenci bu bilgilere web ortamında zaman ve mekândan bağımsız bir şekilde ulaşabilecektir. Bu durumda yukarıda bahsedilen öğrenci, öğretmen, idareci ve velinin aynı bilgileri işlemek ve bu bilgilere ulaşmak için bir araya gelmelerine gerek kalmayacaktır.

1.2 Sistemin Yapısı

Yukarıda bahsedildiği gibi öğrenci ile tüm bilgilerin kaydedilebileceği ve web’de bu bilgilere ulaşılacak bir platform geliştirildi. Program; üç farklı kullanıcı grubuna hitap etmektedir. Bunlar; idareci, öğrenci-veli ve öğretmen kullanıcı guruplarıdır. Önce idareciler tarafından kullanıcılar belirlenerek tüm kullanıcılara programa giriş için kullanıcı adı ve parola belirlenir. Ders tanımlamaları yapılır ve öğretmenler girdikleri her ders için veri girişi yapabilmek yetkilendirilir. Öğretmenler sadece yetkilendirildikleri derse ilişkin öğrenci notlarını ve öğrenci hakkında gözlem ve düşüncelerini girebilir. Öğrenci ise, sisteme kayıtlı bilgilerden sadece kendine ait bilgileri görebilecektir. İdareci ya da öğretmen modunda sisteme giriş yapan kullanıcı yetkisi dahilinde sisteme veri kaydı yapabilecektir. Öğrenci-veli modunda sisteme giriş yapan kullanıcı ise sadece öğretmen ve idareciler tarafından girilen bilgilere ulaşabilecektir. Yani öğrenci-veli modunda ki kullanıcı sisteme veri girişi yapamayacak, buna karşılık önceden girişi yapılan verileri görebilecektir.

1.3 Tezin Organizasyonu

Birinci bölümde; internet ve bilgi teknolojileri alanlarındaki çalışmalarda sürekli karşımıza çıkacak konular açıklanmaya çalışılmıştır. Bu açıklamalardan sonra öğrenci işleri ile ilgili işlemlerin bir nebze olsun rahat ve hızlı yapılabilmesini sağlayacak programın gereklilik nedenleri açıklanması sonrasında sistemin yapısı açıklanmıştır. Sistemin yapısı başlığında; sisteme dahil olan kullanıcıların farklı kullanıcı grubuna üye olduğu, üyelik grubuna göre farklı yetkilerinin olduğu, v.b. sistemin çalışması ile ilgili bilgiler verilmiştir.

İkinci bölümde; web programlama teknikleri üzerinde durulmuştur. Web sayfalarının salt HTML kodlarından dinamik bir içeriğe bürünmesinde ki süreç irdelenmiştir. Bu bağlamda çalışma prensibine göre ikiye ayırabileceğimiz web programlama teknikleri incelenmiştir. Çalışma prensibine göre web programlama teknikleri; sunucu taraflı ve istemci taraflı olmak üzere ikiye ayrılır. Bu bilginin ışığında sunucu ve istemci tarafında çalışan web programlama tekniklerinin genel özellikleri verildikten sonra, sunucu taraflı çalışan web programlama tekniklerinin, istemci taraflı çalışan web programlama tekniklerine göre üstün yanlarının olduğu sonucuna ulaşılmıştır. Bu sonuç ile tez çalışması için hazırlanacak projede kullanacak platform seçimini yapabilmek için günümüzde yaygın kullanım alanı bulmuş ve sunucu tarafında çalışan web programlama tekniklerinin özellikleri incelenmiştir.

Üçüncü bölümde; bir okulda öğrenci ile ilgili yapılması gereken işler anlatılmıştır. Genel olarak yedi bölüme ayrılan öğrenci işlerinin; öğrenci işleri ile ilgili tüm detay bilgilerinin tek tek ve toplu olarak irdelendiği kısım, birinci aşamadır. Öğrenciye ait tüm bilgiler işlendikten sonra öğretim yılı içerisinde yapmış olduğu devamsızlıklarının da kayıt altına alınması gerekmektedir.

Devamsızlık bilgileri, öğrenci işlerinin ikinci aşamadır. Üçüncü aşamada; yüzlerce ve hatta binlerce öğrencinin öğrenim gördüğü bir okulda yapılması gereken işlerin daha pratik ve hızlı yapılabilmesi için belli bir sistematikte hazırlanması gereken listeler konusu incelenmiştir. Dördüncü aşamada; yılsonunda (İkinci sömestri sonu) ve birinci dönem sonunda yapılması gereken bazı işlemler anlatılmıştır. Beşinci aşamada; yılsonlarında öğrencilerin mezuniyet ve diploma işlemleri anlatılmıştır. Altıncı aşamada; okul tarafından Milli Eğitim Müdürlüklerine gönderilmesi zorunlu olan bazı istatistikî bilgiler irdelenmiştir. Yedinci ve son aşamada ise; dönem içi, dönem sonu ve yılsonunda yapılan

1. Ortalama Yükseltme Sınavı,
2. Seviye Tespit Sınavları ve
3. Sorumluluk Sınavları ile ilgili Giriş, Not girişi ve gereken tüm yazılı belgeleri ile ilgili açıklamalarda bulunulmuştur.

Dördüncü bölümde; geliştirilen “Öğrenci İşleri Otomasyonunun” HTML kodlarından arındırılmış, salt PHP kodlarını içeren kod blokları yordamsal anlatım yoluyla açıklanmaya çalışılmıştır. Yordamsal bir anlatım seçilirken, programın her bölümü değil, sadece önemli görülen kısımları irdelenmiştir. Öncelikle programın giriş sayfasında; kullanıcı adı ve parolasın giren kullanıcının kullanıcı sınıfına göre yetkilendirildiği sayfaya yönlendirilmesinde yürütülen kodlar incelenmiştir. “Yönetici Sayfası” başlığında şifre işlemleri, kullanıcı tanımlama ve ders yetkilendirme modülleri yordamsal anlatım yoluyla incelenmiştir. “Öğretmen Sayfası” başlığında not girişi, “Öğrenci Sayfası” başlığında “kişisel sayfa modülleri irdelenmiştir.

Beşinci bölümde; geliştirilen “Öğrenci İşleri Otomasyonu” programını oluşturulan tüm menüler tek tek kullanım açısından açıklanmıştır. Bu bölüm için, programın yardım menüsü olarak ta isimlendirilebilir.

Altıncı bölümde ise tez çalışmasının irdelemesi yapılarak, ulaşılmak istenen amaçlar açısından sonuçlar tartışılmıştır.

2. WEB TABANLI PROGRAMLAMA TEKNİKLERİ

HTTP'nin ortaya çıkışı ile Internet gerçek kimliğine kavuşmuştur. Bu sayede, Web sunucusu ile konuşan bir Web tarayıcının sunucudan aldığı zengin içerikli bilgi, uç kullanıcılara resim, ses, görüntü gibi zengin içerikte ve kullanıcı dostu bir ara yüzle ulaştırılabilir hale gelmiştir [3].

Bu zengin içeriğin değişmeyen sayfalar olarak tutulması kullanıcılar için çok şey ifade etmediği için onlara içeriği otomatik değişebilen ve hatta veri tabanı erişimi sunan siteler oluşturmak için çok çeşitli teknolojiler geliştirilmiştir.

Web ortamında dinamik sayfalar oluşturma'nın temel olarak iki yolu vardır:

- Sunucu tarafında çalışan uygulamalar kullanmak ve
- İstemci tarafında çalışan uygulamalar kullanmak.

İstemci tarafında çalışan uygulama olarak; Java Appletleri, Netscape'in JavaScript'i ve Microsoft'un VBScript'i gibi betik dilleri kullanılabilir. Avantajları:

- Sunucuyu meşgul etmemesi: Veri girişi kontrolleri, menüler, genişleyebilir listeler gibi istemci tarafında yapılabilecek işler için sunucu meşgul edilmemiş olur.
- Bant genişliğini etkili kullanma: Yukarıdaki avantaj bant genişliğini de etkilemiş oluyor.

Sunucu tarafında çalışan uygulamaları kullanmak ise, şu noktalarda istemci tarafı uygulamalara üstünlük sağlamaktadır:

1. Web tarayıcılarında betikler için standart bulunmamaktadır. Bu nedenle bir tarayıcıda çalışan bir betik diğerinde çalışmayabilir. Tarayıcıların Java Appletlerini yorumlamada kullandıkları Java sınıflarının versiyonu tüm Appletleri çalıştıramayabilir. Kullanıcıdan yeni sınıfları bilgisayara indirmesini sağlamak gerekebilir.

2. Sunucu tarafta çalışması zorunlu bazı uygulamalar olabilir (veri tabanı erişimi, işletim sistemi komutları veya başka bazı araçlar sadece sunucu üzerinde çalıştırılabilirler).

3. Bant genişliğini kullanım açısından, işlemlerin önemli bir bölümünün sunucu tarafında yapılması gerekir.

4. Uygulamaların güncel tutulması ve bunu yaparken de istemciler üzerinde değişiklik yapılmayıp sadece sunucu üzerinde (tek bir merkezden) gereken değişikliği yapmak tercih edilir.

Günümüzde sunucu tarafı uygulama geliştirme yönünde genel bir eğilim bulunmaktadır. Uygulamaların tek bir merkezden sunumu ve hatta kullanıcılara program değil ağ üzerinden hizmet satma giderek daha çok önem kazanmaktadır [4].

Ancak Web tarayıcılarının neredeyse bir işletim sistemi kadar yüklendiği günümüzde hem istemci, hem de sunucu taraflarda yukarıdaki faktörleri göz önüne alarak dengeyi koruyacak şekilde uygulamalar geliştirmek gerekiyor.

Yukarıda genel olarak verilmeye çalışılan Web programlama dillerinin türleri ve detayları aşağıda sunulmuştur.

2.1 CGI

Web geliştiriciler statik sayfalardan, akıllı dinamik sayfalara geçmek istediklerinde, kısa adı CGI olan teknolojiyi geliştirdiler. Bu geliştiricilerin asıl istediği, Web sunucusu üzerinden sunucu tarafında programlar çalıştırabilmektir [5]. Bu yeni teknolojiye “Common Gateway” denilmesinin üç önemli sebebi vardır:

- CGI programları sunucudan bağımsızdır,
- CGI programları hemen her dille yazılabilir ve
- Hemen her istemcide çalışabilir.

CGI programlarında temel fikir, uygulamaların mantık katmanının, sunucu tarafında oluşturulmasıdır.

CGI Programları İle Yapılabilenler

Bir CGI programında aşağıdaki dört basamak genel olarak gerçekleştirilir. Ancak bunların bazıları her uygulamada olmayabilir.

1. Kullanıcıdan bilgi alma,
2. Alınan verilerin bir veritabanına işlenmesi,
3. İstenen verilerin veritabanından bir sorgu sonucu alınması,

4. Kullanıcıya işlenmiş bilgilerin gönderilmesi.

Bu adımları kullanabilecek uygulama sayısı sınırsızdır denilebilir. Aşağıda bu uygulamalara örnek teşkil edebilecek konular bulunmaktadır.

- Siteyi ziyaret eden kişilerin bırakacağı e-posta adreslerine, sitedeki yeniliklerden haberdar olunması için düzenli olarak e-posta gönderilmesi,
- Ziyaretçilerin, site hakkında görüşlerini bildirebileceği bir erişim formu,
- Site içerisinde uygulanacak anketler,
- Site içi arama motorları,
- Web tabanlı stok yönetimi, muhasebe, iş akışı, proje yönetimi vb.

2.1.1 CGI Programlamanın Özellikleri

1. **CGI için dil seçimi:** CGI programları için istenen birçok dil kullanılabilir. Ancak yaygın olarak PERL ve C kullanılmaktadır.

2. **CGI programlarında en önemli konu:** İşlenen verinin ham veri değil, bir kullanıcının girdiği bilgiler olduğudur. Dolayısıyla programcı, karşı tarafta rasgele bir bilgi kaynağı olmadığını unutmamalıdır.

3. **Güvenlik:** CGI programları Web sunucusunun çalıştığı makineyi normale göre daha güvensiz hale getirebilir. İnternet'te bulunan birçok makine CGI programlarının güvenlik açıkları nedeniyle kırılmıştır ya da geçici süreler için işlemez hale getirilmiştir.

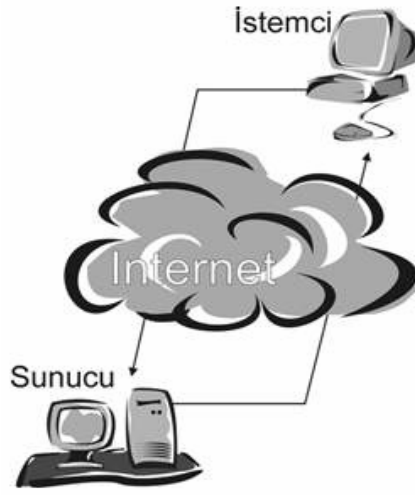
4. **Hız ve bellek kullanımı:** CGI programları yazılırken mümkün olduğu kadar hızlı çalışması ve az bellek kullanması için optimizasyon yapılmalıdır. Sunucu, bir istemci için gerekli CGI programını çalıştırırken, diğer istemcilerin isteğini bekletmemelidir [6].

5. **Taşınabilirlik:** Uygulamalar yazılırken mümkün olduğu kadar kullanılan işletim sistemi ve Web sunucudan bağımsız olmasına dikkat edilmelidir. Bu şekilde, kullanılacak işletim sistemi veya sunucunun değiştirilmesi durumunda programların küçük değişikliklerle çalıştırılması mümkün olur.

6. **Kodları tekrar kullanım:** Uygulama birden fazla program kullanıyorsa, ortak parçaların tek bir parça altında toplanması yararlı olacaktır. Bu şekilde hem kodların yönetimi daha kolay olur, hem de bazı kritik hataları her programcıkta düzeltmek gerekmez [7].

7. Siteye bütünleştirme: Mmkn CGI uygulamaları sitenin genel grnmn bozmayacak ıktılar retecek ekilde programlanmalıdır. Bunu yaparken ablonlama sistemi kullanılırsa, sitenin grnm deęişince, CGI uygulamaları yeni grnme kolayca adapte edilebilir.

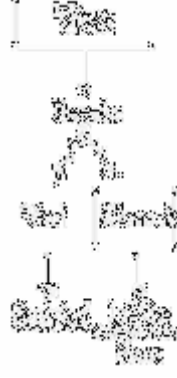
Şekil 2.1’de grldę zere Web sunucu programı, istemci programdan kendisini alıřtıran bilgisayara gnderilen komutlar iin bir geit noktasıdır.



Şekil 2.1 İstemci Sunucu alıřması

Internet’te istemciler ve sunucular vardır. İstemci, bir Internet tarayıcı programı kullanır. Bu program, kullanıcı tarafından istenilen Internet adresini bulmak ve bu adresteki HTML belgesini tarayıcı penceresinde grntlemekle grevlidir [8]. Internet’in dięer ucunda bulunan sunucu da tarayıcı programı olup, bir bilgisayarda alıřır. İstemci olarak gnderilen talep, sunucuya ulařtıęında;

- nce Web sunucu programı, talebi inceler.
- Taleple birlikte kendisine gelen birok bilgiyi kaydedeceęi bir ortam oluřturur.
- Sonra talep edilen HTML dosyasını kendi bilgisayarında bulur ve istemciye gnderir.



Şekil 2.2 Sunucuda Taleplerin Yorumlanması

Sunucu, kendisine gelenleri inceler ve ikiye ayırır: Bunlar veriler ve komutlardır. Veriler, kullanıcı için o anda oluşturulan ortamda kaydedilir; komutlar ise çalıştırılmak üzere önce kendi derleyicisine sonra işletim sistemine aktarılır. Web sunucu programının çalıştığı işletim sistemi, kendisine Web sunucu tarafından iletilen komutu inceler ve gereğini yerine getirir.

Bu anlamda CGI, istemcinin Web sunucu ve onun işletim sistemine "iş yaptırttığı" noktadır. CGI programı, bu işleri belirten programdır. PERL, bu programları yazıldığı ve çağırıldığı programı yazmakta kullanılan bir dildir.

PERL veya hangi dille yazılırsa yazılsın, CGI programı kendi başına iş yapmaz, Web sunucuya o işin yapılmasını bildirir. Başka bir deyişle PERL ile yazacağımız CGI programı aslında sadece Web sunucuyu "programlamaya" yarar.

Web Sitesi ve CGI Uygulamalarının Güvenliği

Çok karşılaşılan bir durum: Web sitesi geliştirilirken, veritabanlarından bilgi alma, onlara yazma gibi gereksinimler olmaktadır. Bunun için de CGI uygulamaları tercih edilebilir. Web sunucusuna bağlandığında makineye izinsiz giriş yapan kötü niyetli kullanıcıların varlığı fark edilebilir.

Bu bir senaryodan çok, sıkça karşılaşılan bir durumdur. Bir bilgisayarın güvenlik riskleri fişe takıldığı andan itibaren başladığı halde, güvenlik hala ihmal ediliyorsa, bunun birkaç sebebi vardır:

- Yetersiz bilgi,
- Makinenin girilemez olduğunu varsayma,
- Art niyetli hiç kimsenin bu sunucuya giriş yapmak istemeyeceği düşüncesi,

- Programları yazarken yeterince sınamama ve
- Tembellik.

Güvenlik deliklerinin başlıca sebepleri: Bütün bilgisayarların ve programların insan elinden çıktığı düşünülürse, onlara zorla girebilecek başka kişilerin var olması da çok normaldir. Web sunucuları gibi çoklu kullanıcı ve dış dünyaya açık olması zorunlu sistemlerde ise bu risk artmaktadır [9]. Genel olarak bir Web sunucusunun güvenlik açıklarının sebepleri şunlardan bir kısmı veya hepsidir:

1. Karşı uç
2. Kullanıcı verileri
3. Web sunucusu
4. Web tarayıcısı
5. CGI, ya da genel olarak sunucu tarafı çalışan programlar ve onların çağırdığı programlar.

Karşı uç: Hattı dinleyen, sistemdeki açıkları taratan bir saldırgan da olabilir, virüs gibi sisteme istenmeden yerleşen bir program da olabilir.

Kullanıcı verileri: İstemciden alınan veriler; bu verileri her ne şekilde işlenirse işlensin, bir kontrolden geçirmek faydalı olacaktır.

Web sunucusu: Kullanılan Web sunucusunun çeşitli açıkları biliniyor olabilir. Güvenlik sitelerinde araştırma yaparak, kullanılan sunucu sürümüne özel, ya da eski sürümlerde olup, mevcut sürümde hala kapatılmamış açıkların bir listesini çıkarılması faydalı olacaktır.

Web tarayıcısı: 2. maddeyle birleşik düşünülebilir. Güvenlik, kullanıcıdan gelen verileri doğrulamak için istemci tarafı tekniklere (Örnek: JavaScript'le form doğrulama) dayandırılmamalıdır.

CGI, sunucu tarafı programlar: Web sunucu, CGI programlarıyla saldırıya daha açık hale gelebilir ancak bu her zaman daha güvensiz olacaktır şeklinde yorumlanmamalıdır.

2.2 ASP

ASP, Microsoft firması tarafından klasik HTML sayfalarına dinamik bir yapı kazandırmak amacıyla ortaya çıkarılmış bir teknolojidir. Internet, yeni bir teknoloji olduğundan

insanlar hayal gücü sınırlarını bu yeni teknoloji ile zorlamaktadırlar. ASP de bu yeni teknolojinin yeni bir parçası olmuştur.

Web sunucu programları, sabit disk üzerinde yer alan bilgileri istemcinin tarayıcı programına göndermekle sınırlı olan bir yapıya sahiptir. Web sunucusunun kullanıcıya statik bilgi sunmasının yanı sıra ziyaretçiden de veri kabul etmesinin gerekli olacağı düşünülür. Böylelikle Web sitesi üzerinden kullanıcının taleplerine göre tepki veren bir bilgi sunumu gerçekleşmiş olacak, ayrıca ziyaretçiden de veri alınabilecek bir yapı kurulmuş olacaktır.

Bu hedefle, CGI katmanında ziyaretçinin tarayıcı programından veri ve komut gönderilmesi sağlanır. Sadece metin ve basit çoklu ortam nesnelerini destekleyen Web sunucularının fonksiyonellikleri CGI aracılığı ile artırılmış olur.

Microsoft Web işlemleri konusundaki stratejisini, harici programları işletim sistemine kaydırmak olarak belirlemiştir. Bu yüzden CGI yerine işletim sisteminin bir bileşeni olan AP kullanılması uygun bulunmuştur. Bu amaçla birçok ISAPI tasarlanmıştır. Gerek hız gerekse de sağlıklı bir işleyiş açısından bu yeni yapı üzerindeki Microsoft'un ürünü ASP olarak piyasaya çıkarılmıştır [10].

Microsoft ASP, etkileşimli Web sayfalarını oluşturmak, güçlü ve hızlı Web uygulamalarını geliştirmek için, sunucu tarafında işleyen komutlar dizisidir. Tasarımı yapılmış bir ASP uygulaması uzantısı .asp ile kaydedilmiş olan bir metin dosyası içerisinde yer almaktadır. Başka bir ifadeyle ASP, JavaScript gibi, VB Script gibi komutları, düz yazı dosyası olarak kaydedilen ve çalışabilmesi için yorumlayıcıya ihtiyacı olan bir dildir. ASP dosyaları kendi başına çalışabilecek bir dosya olarak düşünülmemelidir. Kendisini aktif hale getirecek bir tetiklenmeyi Web sunucusu üzerinde bekler ve sunucu tarafında çalışır. Sunucu kendisinden .asp uzantılı bir dosya talep edildiği zaman, dosya da yer alan komutları kontrol edip işler.

ASP, tamamen sunucu tarafında çalışan bir programlama dilidir. Tüm kodlar sunucu tarafında çalıştırılır ve kullanıcıya sadece HTML kodları gönderilir. Böylece performans olarak büyük bir avantaj elde edilmiş olur. Fakat bu, Web sunuculuğu yapacak bilgisayarın hızına da bağlıdır. Örnek olarak ÖSYM ele alınırsa; kullanıcıdan aldığı bir ÖSYM numarası ile bir program çalıştırılır ve bu program kullanıcıya sadece girmiş olduğu numaraya ait sonucu bir HTML dokümanı olarak döndürür [11].

Bilgisayar programcılarının geliştirdikleri uygulamaları dünyanın her yerinden erişilebilecek bir platforma kavuşturmak ASP ile mümkün olmuştur.

ASP dosyaları;

- 1- Ziyaretçi tarafında çalışan HTML komutlarını,
- 2- Veritabanına bağlanmayı, veritabanı işlemlerini belirleyecek cümleleri,
- 3- Yürütülecek işlemler zincirinin hizmet ettiği çeşitli görevleri sağlayan COM bileşeni çağrılarını içermektedir.

ASP için WEB platformunun bütünü birer nesnedir. Dolayısıyla bu ortam içindeki her şeye müdahale edebilme ve programlanabilme imkânı sağlanır.

ASP'yi sadece interaktif Web sayfalarını üreten bir yapı olarak ele almak eksik bir tanımlama olacaktır. Bilişim sektöründe yazılım alanında çalışan uzmanların, süratle yaygınlaşan internetin dünyasına yönelmeleriyle yeni bir kavram ortaya çıkmıştır: Internet Programcılığı. Burada, ASP Internet programcılarının Web ortamında çözüm sunmalarına imkân sağlamaktadır.

Basit HTML sayfalarıyla, istemci herhangi bir sunucudan herhangi bir Web adresini talep eder. Sunucu ise, bu talep geldiği anda sayfayı istemcinin bilgisayarına gönderir. ASP'de ise; sunucu, sayfayı istemci bilgisayarına göndermeden önce üzerinde değişiklikler yapabilir. Yani uzantısı .asp olan her dosya, istemci bilgisayarına gönderilmeden önce sunucuda ASP.DLL ismi verilen bir DLL'de çalıştırır.

2.2.1 ASP Programlamanın Özellikleri

ASP kullanabilmek için, Microsoft Web sunuculardan birinin kurulu olması gerekir. Microsoft NT Server Kurulu makinede, ASP'yi destekleyen IIS 3.0 veya IIS 4.0 yüklü olmalıdır. Windows 2000 makinede ise, IIS 5.0 yüklü olmalıdır. Windows 95–98 makinede ise, Personal Web Server yüklü olmalıdır. Eğer Web sitesi UNIX üzerinde kurulu ise, bu durumda ASP'yi istemciye göndermeden önce çevirecek üçüncü parti yazılımları kullanılmalıdır. Bunlardan birisi de Chilisoft'tur.

1. ASP'nin çalışması: ASP kodları, SQL, Access, Oracle, Informix gibi bir veritabanına bağlantılı olabilir ve verileri dinamik olarak sayfalara eklenebilir.

ASP'nin nasıl çalıştığının anlaşılabilmesi için; istemci/sunucu modelini ve Web programlama dillerinin bunlarla nasıl ilişkilendiğinin anlaşılması gerekir. Internet ortamında, herhangi bir adres ziyaret edilince, ziyaret eden istemci olur ve ziyaret edilen sitenin üzerinde bulunduğu bilgisayar ise sunucu olur. ASP'nin sunucu taraflı teknoloji olmasından dolayı, talep edilen sayfa sunucu üzerinde çalıştırılır. HTML, gerçek zamanlı olarak hazırlanır ve istemciye

sayfa yollanır. Yani eğer tarayıcıda sayfanın kaynak kodu görüntülenmek istenirse, ([view]->[source]), gösterilecek tek şey salt HTML kodudur.

Bunun yanında JavaScript gibi istemci taraflı diller, istemci makinesinde çalıştırılır. Yani; içerisinde JavaScript kodu bulunan bir sayfa çağrıldığında, sayfada hiçbir değişiklik yapılmadan aynen kodlandığı gibi gönderilir ve her şey istemcinin makinesinde yapılır. Hem ASP hem de JavaScript aynı sayfa içerisinde kullanılabilir. Bilinmesi gereken, önce ASP'nin, sonra JavaScript'in icra edildiğidir. ASP'yi bu kadar kolay ve kullanışlı yapan, HTML sayfaları içerisine gömülmüş ASP kodlarından oluşmasıdır.

ASP sayfaları nasıl çalıştırılacak? İlk olarak, ASP'nin hangi dil kullanarak kodlanacağına karar verilmelisi gerekir. Aslında bütün betik dilleri ile ASP kodlanabilir. Fakat kullanılan en popüler iki dil olan VBScript ve JScript'dir. İçlerinde en çok kullanılanı VBScript'tir. Çünkü bu dilin yapısı diğerine göre biraz daha kolaydır [12].

2. ASP'nin çalıştırılması: Asp kodları yazmak için sadece basit bir metin editörü gereklidir. Çalıştırmak için ise, bir Web sunucusuna gereksinim vardır. Asp kodları sadece Web sunucunun kök dizinine kaydedilir, bu dizin C:\inetpub\wwwroot klasörüdür. Tarayıcının adres satırına http:// bilgisayar_adı yazınca kök klasöre gider ve içindeki default.asp dosyasını çalıştırır. Default.asp'den farklı bir dosya çalıştıracaksa adres satırına http:// bilgisayar_adı/dosya_adı.asp biçiminde yazılmalıdır.

3. ASP'nin kullanım alanları ve avantajları: ASP dinamik Web sayfaları oluşturmak için kullanılır. Web sayfalarını canlandıracak bir güce sahiptir. Web sayfalarına işlem işleme yeteneği kazandırır.

ASP Web Programcılarına HTML, betik dili ve kullanıcıdan bağımsız veritabanı işlemlerini rahatlıkla kullanabilme olanağı sağlar. Bu özgürlük ve platform bağımsızlığı, işlem gören tüm işlemlerin sunucu tarafında halledilerek, sadece sonuçları üretmesi ile sağlanır. Sonuç salt HTML kodunda üretildiği için tarayıcı üzerinde tanımsız komut gibi bir problemle karşılaşmaz.

ASP, kendi içinde JavaScript, VB Script, JScript, PERL Script gibi geniş bir yelpazedeki kodların kullanılmasına olanak tanır. Hatta Windows betik host ile uyumlu kalmak şartıyla oluşturulan her yeni betik dil, Web sunucuya tanıtılarak ASP üzerinde kullanılabilir.

ASP, sunucu tarafında uygulama çalıştırmaya imkân sağlar. Hazırlanan uygulamaları çalıştıran komutlar sadece sunucu tarafında kalır. İstemci sadece sonuçları görebilir. Bu yönüyle ASP, Web programcılarına güvenilirlik ve özgünlük sağlamaktadır.

Uygulama ile ilişkilendirilmiş bir veritabanı üzerindeki veriler ASP ile Web sayfalarına aktarılır ve bilginin Internet üzerinden güncellenmesi otomatik hale dönüşebilir.

ASP ile hazırlanan projeler, kaynakların yönetimi, kontrolü ve uzaktan sistem yönetimine imkân verir. Şirketler arası bilgi paylaşımı için ideal bir ortam oluşturur.

2.3 PHP

PHP 1994 yılında Rasmus Lerdorf tarafından Web sunucusuna koyduğu özgeçmişinin kaç kişi tarafından okunduğunu izlemek için geliştirilmeye başlanmıştır. İlk başta bu versiyonlar halka açık olmamakla birlikte 1995 yılında Personal Home Page Tools adı altında diğer insanların kullanımına açık hale getirilmiştir. O yıllarda, içerisinde çok basit bir iki makroyu anlayabilen ufak bir derleme motoru kişisel sayfalarda kullanılmaya alışıktır. 1995 yılında derleme motoru tekrar yazılarak adına PHP/FI V2 denildi. Bu sürüm, Rasmus tarafından yazılan ve HTML form bilgilerini işleyebilen bir yapıya sahipti. Buna MySQL desteği de eklenince PHP/FI tam anlamıyla doğmuş oldu. Birçok kişinin desteği ve kendi yazdıkları kodları paylaşımları sonucu çok hızlı bir şekilde gelişti ve 1997 de Zeev Suraski ve Andi Gutmans tarafından tamamen yeniden yazılan derleme motoru PHP V3 için bir taban oluşturdu. Son zamanlarda PHP V3 gibi sıfırdan yeni yazılan PHP V4 kullanıma sunulmuştur [13].

PHP, bir betik dilidir; yani kodları düz yazı dosyaları halinde kaydedilir ve kullanılacağı ortamda bir yorumlayıcı tarafından yorumlanır. Bu, PHP ile yazılacak programları, derlemek yani ortaya bir EXE veya çalıştırılabilir başka bir dosya çıkartmaya gerek olmadığı anlamına gelir. PHP Betikleri çalıştırabilmek için bu dili bilen bir program gereklidir. Bu programın tek başına çalışması mümkün olduğu gibi, Web sunucu tarafından da çalıştırılabilir. İster kişisel bilgisayarınızda, ister gerçek Internet ortamında olsun, bir Web Sunucu, ziyaretçinin (Internet istemci programının, Tarayıcının) talep ettiği dosya adının uzatması olarak .php, .php2, .php3 veya .php4 görünce, işlem yapmaya başlar. Çünkü Web sunucu, uzatması ".htm" veya ".html" olan bir dosyayı, kendi sabit disklerinden birinde bulur ve ziyaretçinin bilgisayarına gönderir. Fakat .asp, .pl, .cfm gibi, .PHP uzantısı, Web sunucuya bu sayfayı olduğu gibi alıp ziyaretçinin tarayıcısına göndermek yerine, önce PHP yorumlayıcıyı çağırır. PHP yorumlayıcısı (yani Windows ortamında PHP.exe veya 4. sürümle gelen PHP4isapi.dll, Unix ve türevlerinde çalıştırılabilir PHP dosyası), kendisine teslim edilen bu belgede önce "<?PHP" ve ">" ayrıçaları arasındaki PHP kodlarını seçerek alır ve işlemi yapar. Bu ayrıçaların içinde kalan kodlar, yapılacak işlemin komutlarıdır. HTML ile yetinmeyip, PHP kullanılmasının sebebi, bu komutların HTML'in yapamayacağı şeyleri yapmasıdır.

HTML'in yapamayacağı şeylerin başında, Web sunucusunda yapılacak işler vardır. Sözelimi HTML etiketlerini kullanarak, Web sunucusundaki bir veritabanı dosyasını açıp, içindeki veriler okunamaz. HTML etiketleri ile Web ziyaretçisinden bilgi edinebilir ve bunlar sunucuya yollanabilir. Bu bilgileri, işe yarar şekle sokmak için, sunucuda çalışan bir program gerekir. Sunucuda çalışan program, EXE, DLL veya Java Server Pages gibi bir gerçek program veya PERL, ASP ve PHP gibi betik diliyle yazılmış bir program olabilir.

Web Sunucu, .php uzantısını görünce işlem yapmaya başlar, bu dosyayı PHP yorumlayıcısına verir ve o da "<?PHP" ve ">" ayraçları içindeki kodları icra eder. Bu işlemin sonunda ortaya çıkan ürün eğer ziyaretçiye gönderilecekse, HTML etiketleri içinde gönderilir; yani ziyaretçi asla PHP kodlarını göremez. Kimi zaman PHP programının çalışması sonucu, ortaya ziyaretçiye gönderilecek bir ürün çıkmaz. Elde edilen sonuç ya başka bir programa (örneğin elektronik ileti gönderen bir Sendmail programına) hammadde olarak verilir, ya da sunucuda sabit diske yazılır. Fakat her durumda, Web ziyaretçisi PHP kodlarını göremez. Bu ise Web sayfalarında PHP kodu kullanılması durumunda, Web sayfalarının tarayıcı tarafından tanınması veya tanınmaması gibi bir durumun ortaya çıkmaması demektir. Ziyaretçi ne tür Tarayıcı kullanırsa kullansın, kendisine sunucudan daima saf HTML kodu geleceği için, Web sayfalarını rahatça görebilecektir. Fakat Web sayfalarında herhangi bir tarayıcının hatalı yorumladığı veya hiç yorumlayamadığı HTML etiketleri bulunabilir, ya da sözelimi ziyaretçiye giden HTML'in içinde tarayıcıda çalıştırılacak olan başka betik kodları bulunabilir ve tarayıcı bu kodları anlamayabilir. Örneğin, HTML sayfalarında VBScript kodları kullanılmışsa, Netscape ile ilgili siteye bağlanan kişiler bu kodun oluşturması istenilen etkiyi göremeyeceklerdir. Çünkü Netscape VBScript kodlarını yorumlayamaz. PHP ile yazılacak Web programlarının oluşturacağı HTML belgesinin tarayıcı türleri ve sürümleri ile uyumluluğunu sağlamak yine Web tasarımcısının sorumluluğundadır.

Özet olarak, PHP bir CGI programlama dilidir. Bu dille, Web sunucusu ile Web ziyaretçisi arasındaki buluşma noktası olan CGI'da bilgi alışverişi yapılabilir. Sunucuda bulunan başka programlar çalıştırılabilir ve böylece Web sayfalarına HTML'in sınırlamalarının ötesinde hareket ve ziyaretçi ile etkileşme olanağı kazandırılabilir.

2.3.1 PHP Programlamanın Özellikleri

PHP, bir Web sunucu programıdır. Kurulacağı sistemdeki Web sunucusu ile uyumlu olması gerekir. 4. sürümü itibariyle, PHP bütün Unix ve türevi sistemler (örneğin Linux) ile Microsoft Windows sistemleri ile uyumludur. Birlikte çalışabileceği Web sunucu programları arasında Apache, IIS, FHTTP, Omni HTTPd, Xitami ve Windows 95/98 için Personal Web Server vardır. Gerçek Internet ortamında sunucu programlara PHP yeteneği kazandırma işlemi,

Web programcısını ilgilendirmez. Web Programcısı için, Internet sitesine ev sahipliği yapan hosting firmasının, PHP desteği verip vermediğini bilmek yeterlidir. Fakat tasarım aşamasında, ilgili çalışmaları denemek için programcı bilgisayarına bir kişisel Web sunucusu kurmak ve bu programı PHP ile uyumlu hale getirmek gerekmektedir.

1. Veritabanıyla Uyumlu Çalışır: PHP, veritabanlarıyla mükemmel iletişim ve platformdan bağımsız olarak çalışır. Ayrıca, açık kaynak kodludur. PHP Web’de çalışmak üzere tasarlanmıştır. Bir veritabanına bağlanmak ve sorgulama yapmak iki veya üç satır kodla mümkün hale getirilmiştir [14].

PHP özellikle 4.0 versiyonu ile yeni Zend motorunu kullanmaya başlamış, performans üst seviyeye çıkarılmıştır. Daha fazla performans için, PHP’yi Web sunucusunun bir parçası haline getirebilir.

Amaç sadece betiklerin hızını arttırmak olsaydı, PHP tek başına birçok problemin cevabı olurdu. Fakat PHP’nin özellikleri bundan çok daha fazladır. Programlama dili basit, veritabanlarına ulaşım kolay ve hızlı, nesne yönelimli programlama ile tekrar kullanılabilir kodlar yazmak için çok kullanışlıdır.

PHP ile basit bir geri bildirim formu yapılabilir. Site üyeleri dileklerini form üzerinden Web tasarımcıya gönderebilirler veya baştan aşağıya veritabanı, ağırlıklı bir doküman yönetim sistemi oluşturabilir.

Amaç e-ticaret ise, alışveriş sepetleri yapılabilir. Yazılım geliştirme hızını arttırmak için, Internet üzerindeki hazır kütüphaneleri kullanabilir.

2. Yaygın Bir Kullanım Alanı Olması: PHP ile, formdan bilgi almak, dinamik sayfa içeriği yaratmak, cookie göndermek ve almak gibi diğer bütün CGI programlarının yaptıklarını yapılabilir. PHP’nin en güçlü olduğu ve en çok kendini gösterdiği özelliği, geniş veritabanı desteğinin olmasıdır. PHP’de veritabanı kullanan bir site hazırlamak çok kolaylaşmıştır.

En yaygın olarak kullanılan veritabanı MySQL’dir. MySQL dışında, ODBC kullanarak Access gibi diğer veritabanlarına da ulaşılabilir.

PHP aynı zamanda diğer hizmetlerle iletişimi sağlayan IMAP, SNMP, NNTP, POP3 ve hatta HTTP protokollerini destekler. Bütün bu protokollerin dışında soket programlayabilmeyi sağlar.

3. Güvenlik: PHP çok güçlü bir dil ve yorumlayıcıdır. Gerek sunucunun içinde bir modül olarak çalışırken, gerekse ayrı bir CGI programı olarak çalışırken komutları yorumlama, dosyalara erişme ve ağ bağlantıları açabilmektedir. Normalde bu özellikler bir Web sunucusu üzerinde çalışan herhangi bir şey için düşük güvenlik anlamına gelir. Fakat PHP, PERL ve C ile yazılmış CGI programlarından çok daha güvenli olacak şekilde tasarlanmıştır. Özellikle derleme/kurulum aşaması seçeneklerinin ve çalışma zamanı seçeneklerinin doğru şekilde ayarlanması, ihtiyaç duyulan özgürlük ve güvenliğin en belirginini sağlar. Özelleştirme için verilmiş olan çok büyük bir seçenekler listesi sayesinde, birçok amaç için rahatlıkla ve güvenle kullanılabilir.

4. PHP'nin Yapı Taşları: PHP, bir programlama dili olarak, değişkenler, değişkenlerin değerleriyle bir işlem yapmayı sağlayan işlemciler, işlemcilerle oluşturulan deyimler ve bunların tümünü toplu olarak kullanmamızı sağlayan işlevlere sahiptir. PHP, nesne yönelimli bir dil olduğu için, nesne oluşturma imkânına ve bunların kullanılmasını sağlayan metotlara da sahiptir. Bütün bu imkânları kullanarak, PHP ile bir veritabanından verileri alarak, bunları HTML etiketlerinin değerleri gibi işler. Web sitesini ziyaret eden kişilerden bilgi alınabilir, bu bilgilerle çeşitli işlemler yapılabilir. PHP'nin çeşitli komutlarını, deyimlerini ve fonksiyonlarını kullanarak, programın çalıştığı Web sunucusunun bulunduğu bilgisayarda da çeşitli dosya işlemleri yaptırabilir.

5. PHP ve Veritabanı: PHP'nin varlık sebeplerinin başında, ticari Unix ve Microsoft Windows işletim sistemlerinden ayrı, ticari olmayan açık sistem ortamlarında çalışan bir betik dili ile bu dille kullanılabilecek, yine açık sistem ürünü bir veritabanı erişim ve yönetim aracına sahip olmak gelir. PERL, isteyen Web programcısının ücretsiz edineceği bir CGI programlama dilidir. Ama PERL, metin dosyalarından yazı çekmek ve bunları biçimlendirerek rapor haline getirmek amacıyla geliştirilen bir dil olduğu için, özellikle veritabanına dayalı işlemlerde programcıya çok güçlük çıkarmaktadır. ASP ise (ASP desteği sağlayan ve ücretsiz Web sitesi veren firmaların giderek artmasına ve veritabanı konusunda hem kolaylık, hem de büyük bir etkinlik sağlamasına rağmen) çoğunlukla ücretli Web sitesi alınan ticari amaçlı hosting firmalarının sağladığı bir teknoloji olarak görülmektedir. PHP, tasarımcılarının MySQL ve PostgreSQL teknolojilerini geliştiren kurum ve firmalarla yaptığı yakın işbirliği sonucu ikinci sürümünden itibaren veri yönlendirmeli, Web uygulaması alanında önemli bir araç olarak ortaya çıkmıştır. Dolayısıyla, PHP kurulumu dendiği zaman, kendiliğinden, MySQL kurulumu da kastedilmiş olmaktadır.

PHP ile Web sunucuya sayfayı Internet ziyaretçisine göndermeden önce, bir takım komutları icra etmesini, bir takım veri dosyalarını açıp içindeki bilgileri alıp bunları HTML kodlarıyla bezeyip, tarayıcıya bilinen klasik HTML sayfası olarak göndermesi talimatı verilir.

MySQL ve PostgreSQL ise, Windows sistemlerinde Denetim Masası'nda görülen ODBC Veri Kaynağı Yönetmeni adlı araç gibi, bir veritabanı sunucusu, yani Database Server programlarıdır. Yakın zamana kadar, bu iki veritabanı yönetim aracı arasında fazla bir çekişme olmamıştır.

MySQL, bir kullanıcı lisansı 200 \$ olan ticarî bir program iken, 2000 yılı ortalarında General Public License/Kamusal Lisans uygulamasına geçmiş ve ücretsiz dağıtılır hale gelmiştir [15]. PHP için MySQL'e gerek yoktur. PHP, bir NT veya Win9x tabanlı Web sunucuda çalıştırılıyorsa, Microsoft'un ODBC sürücülerini ile çalışabilir. ODBC'nin okuduğu bütün veritabanlarından veri çekebilir. Ayrıca PHP için Adabas, dBase, Empress, FilePro, Informix, InterBase, MySQL, Oracle, Solid, Sybase, Velocis ve birçok Unix veritabanı yöneticisi için geliştirilmiş add-on (sonradan eklenen) modüller vardır. PHP, birçok LDAP istemci programı için yazılmış API'lere de sahiptir. PHP ile IMAP, SMTP gibi Internet elektronik mektup protokollerini kullanmak da mümkündür.

6. Yüksek Performans: PHP 3.x versiyonları popüler olmalarına rağmen ASP yorumlayıcısına karşı yavaş kalmaktadır. PHP, 4.0 versiyonunda Zend firmasının betik yorumlayıcı motoru kullanılarak tamamen yenilenmiş ve hız olarak ASP ile yarışır hale gelmiştir. Zend'in kendi yaptığı testlerde PHP3.x ile PHP4.x arasındaki fark gözlemlenmiştir. 100MHz Pentium işlemci ve 24MB hafızalı Linux Kurulu bir sistemde şu sonuçlar alınmıştır [16].

Tablo 2.1 PHP 3.0.6 – PHP / Zend Performans Karşılaştırması

	PHP 3.0.6	PHP/Zend		PHP/Zend+optimizer	
	Süre	Süre	Performans Artışı	Süre	Performans Artışı
QuickSort algoritması (1000 öge)	5.26dk.	1.11dk	460%	31sn	1050%
QuickSort algoritması (10 öge)	0.1sn	0.06sn	60%	0.06sn	66%
MySQL'de tablo listeleme (1000 kayıt)	4.35sn	0.83sn	525%	0.57sn	765%
MySQL'de tablo listeleme (10kayıt)	1.1sn	0.6sn	185%	0.5sn	220%

PHP 4.0 pre-alpha versiyonu ile Microsoft'un ASP'si arasında yapılan benchmark sonuçları ilginçtir. QuickSort algoritması çalıştıran özdeş ASP ve PHP betikleri 128 MB RAM'li P2-233 sistemi üzerinde Windows NT/Service Pack4 ve Web sunucu olarak IIS 4.0 kullanılarak çalıştırılmaktadır. Burada, PHP yorumlayıcısı ISAPI modülü olarak çalışmakta ve işletim sisteminin NT ve Web sunucunun IIS olması ASP için bir avantaj olması gerekirken öyle olmamaktadır [16, 17].

Tablo 2.2 ASP – PHP / Zend Performans Karşılaştırılması

	ASP	PHP/Zend		PHP/Zend+optimizer	
	Süre	Süre	Performans Artışı	Süre	Performans Artışı
Basit Döngü (10 Milyon iterasyon)	44sn	43sn	2%	12.5sn	352%
QuickSort algoritması (200 öge)	10sn	44sn	-440%	6sn	66%

Ayrıca ASP, özellikle yoğun hit alınan durumlarda Web sunucusunu yavaşlatırken, modül olarak kurulu Apache-PHP ikilisinde bu sorun pek yaşanmamaktadır.

2.4 Karşılaştırma

Günümüzde yaygın kullanım alanı bulmuş olan Web programlama dillerinden CGI, ASP ve PHP dilleri incelenmiştir. Web tasarımında HTML ile oluşturulan sayfaların dinamik içerikli olabilmesi için gerekli olan Web programlama dilleri ile uygulamalar geliştirilmiştir [18].

Web programlama teknikleri hakkında elde edilen bulgular ışığında, CGI, ASP ve PHP Web programlama dilleri kullanılarak aynı işlemi yapan üç ayrı uygulama hazırlanmıştır. Uygulama yapılırken ulaşılması istenen amaç; günümüzde yaygın olarak kullanılan Web programlama dillerinin, performans analizini yapmaktır.

Örnek uygulama olarak; Web'de yayımlanan tüm Web siteleri için kullanılabilir olan, site içi üyelik sistemi seçilmiştir. Üyelik sistemi, günümüzde geçerliği olan CGI, ASP ve PHP Web programlama dilleri kullanılarak ayrı birer uygulama olarak yapılmıştır.

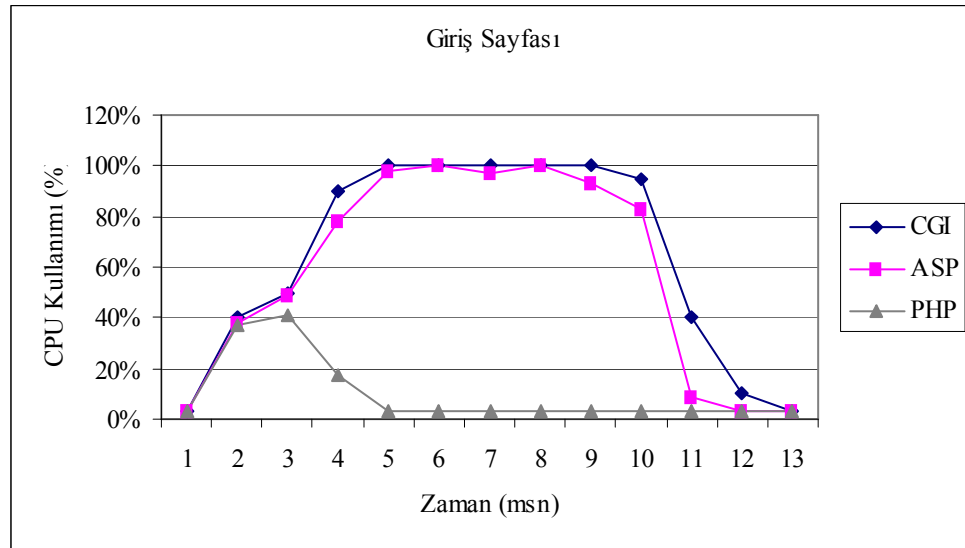
Tüm bu uygulamalar yerel olarak çalıştırılarak, test sonuçları elde edilmiştir. Performans testlerini yapmak için kullanılan sistem özellikleri:

- Sunucu donanımı Celeron 1700 Mhz İşlemci, 256 MB hafıza,

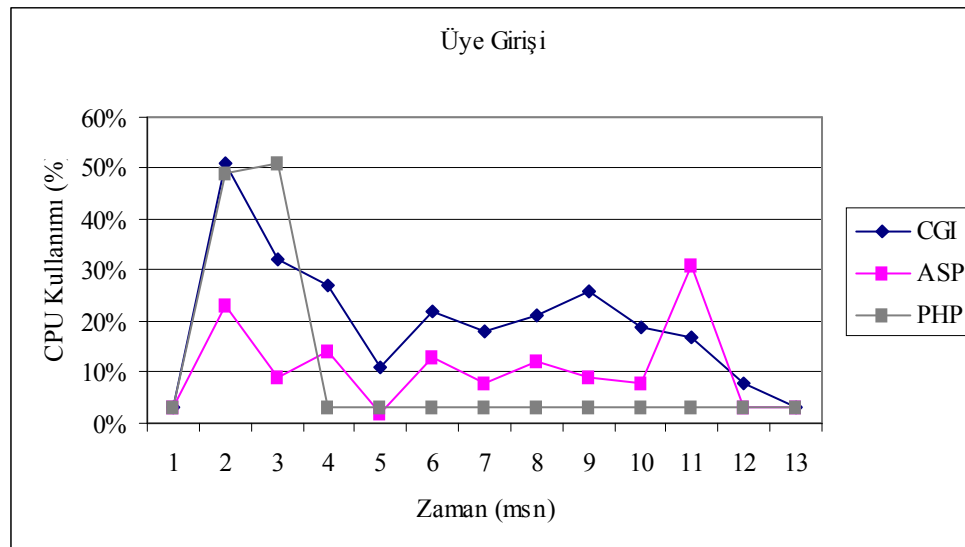
- Sunucu Yazılımı Internet Information Server, Apache Web Server,
- Yük Üretme İstemcisi Windows XP yüklü PC.

Örnek program uygulamalarında sayfada hata olmaması durumunda hafızaya fazla iş düşmemektedir. Yapılan değerlendirmelerde, her bir uygulamada %1 ile %3 arasında hafıza kullanımında farklılıklar görülmüştür. Bu nedenle, uygulamaların üç aşamada CPU'ya bindirdiği yük gözlenerek aşağıya çıkarılmıştır [19].

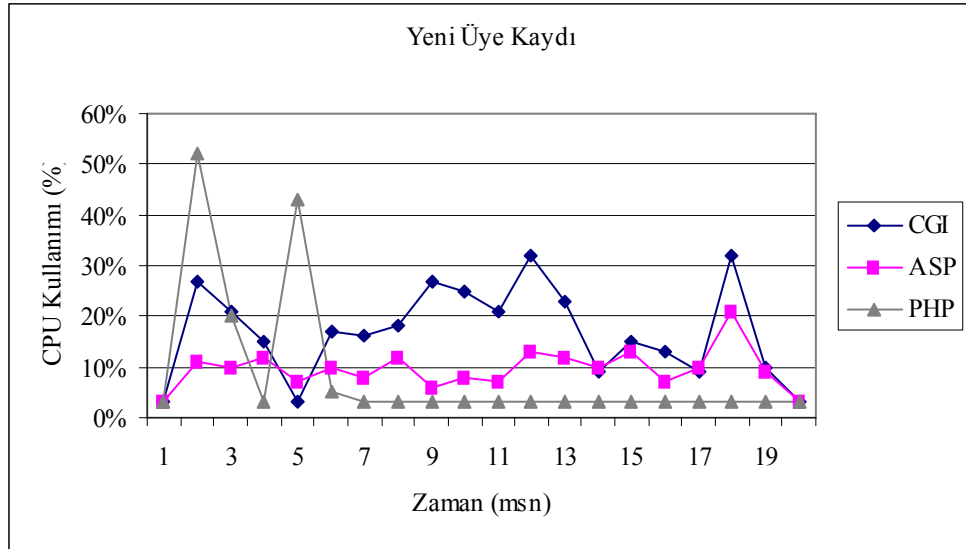
1. Giriş sayfasının yüklenmesinde:
2. Sisteme kayıtlı kullanıcının, kullanıcı adı ve parola girmesiyle, bu bilgilerin kontrolü ve yeni sayfaya yönlendirilmesinde:
3. Sisteme yeni üye kaydının yapılması sürecinde



Şekil 2.3 Giriş Sayfası Yüklenirken CPU Kullanım Oranı



Şekil 2.4 Üye Girişi Yapılırken CPU Kullanım Oranı



Şekil 2.5 Yeni Üye Kaydı Yapılırken CPU Kullanım Oranı

Karşılaştırma Sonuçları

Web tasarımında HTML ile oluşturulan sayfaların dinamik içerikli olabilmesi için gerekli olan Web programlama dilleri ile uygulamalar geliştirilmiştir.

CGI, ASP ve PHP kullanılarak birer örnek uygulama üzerinde, performans değerlendirmesi yapılmıştır. Performans değerlendirmesi; seçilen uygulamada, en çok kullanılan bölümler olduğu için, her bir uygulamada ki giriş sayfası, üye girişi ve üye kaydı bölümleri üzerinde yapılmıştır. Her bir bölümün çalıştırılması durumunda, CPU'yu meşgul etmesi % cinsinden ölçülmüştür. Sunucu bilgisayarında, uygulama öncesi CPU'nun %3 oranında meşgul olduğu şekil 2.3, 2.4 ve 2.5'te görülmektedir.

Aşağıda ki Tablo 2.3'te ise bu değerlendirme oranlarının ortalamaları verilmiştir. Buradan da görüleceği üzere CPU'ya bindirdiği yük açısından değerlendirildiğinde, PHP ~9.3 % ile en iyi performansı sağlamaktadır [19].

Bu nedenle tez çalışması kapsamında geliştirilen “Öğrenci İşleri Otomasyonu Programının” hazırlanması için Web programlama tekniklerinden PHP tercih edilmiştir.

Tablo 2.3 Ortalama CPU Kullanımı (%)

	Giriş Sayfası	Üye Girişi	Üye Kaydı	Toplamda
CGI	64	20	17	~33.6
ASP	58	11	10	~26.3
PHP	10	10	8	~9.3

3. ÖĞRENCİ İŞLERİNİN ANA UNSURLARI

Tüm orta dereceli okullarda rutin işler arasında olan ve her biri idareciler tarafından itina ile yapılması zorunlu olan öğrenci ile ilgili işleri kendi arasında gruppalayacak olursak yedi başlık altında toplayabiliriz. Bunlar:

3.1 Öğrenci Bilgileri

Öğrenci işleri ile ilgili tüm detay bilgilerinin tek tek ve toplu olarak irdelendiği bölümdür. Bu başlık altında sadece öğrencilerin devamsızlıkları ile ilgili bilgiler incelenmez. Bu bölüm 9 ayrı parçadan oluşmaktadır. Bu bölümler şunlardır;

Öğrenci Bilgileri Kayıt

Öğrenci ile ilgili tüm bilgiler detaylı olarak girilmelidir. Öğrenci numarası, adı, soyadı, sınıfı ve cinsiyeti gibi bilgiler kayıt altına alınmalıdır.

Not Girişi

Her bir sınıftaki öğrencilerin her bir derse ait notları girilir. Bu aşamada işlemler belli bir sistematığe göre yapılmalıdır. Önce not girişi yapılacak sınıf belirlenir. Sonra notun ait olduğu ders belirlenerek her bir öğrencinin o dersten aldığı notlar işlenir.

Öğrencinin okuduğu yıla ilişkin notların girişi ile birlikte geçen yıllardan başarısız olduğu derslere ait notlarında girilmesi gerekir.

Ödev İşleme

Öğrencilerin dönemlik veya senelik hangi derslerden ödev aldıkları bu aşamada kayıt altına alınır. Bu işlemin sınıf bazında yapılması en sağlıklı yöntemdir.

Dönem Sonu Notları İşleme

Bu aşamada öğrencilerin dönem sonu notları işlenir. Öğrencilerin her bir derse ilişkin dönem sonu not bilgileri işlenir.

3.2 Devamsızlık Bilgileri

Öğrenciye ait tüm bilgiler işlendikten sonra öğretim yılı içerisinde yapmış olduğu devamsızlıklarının da kayıt altına alınması gerekmektedir. Devamsızlık bilgileri üç başlıkta incelenebilir.

Devamsızlık Girişleri

Öncelikle okulda verilen eğitim türüne göre girilecek devamsızlık şekli belirlenir. Bu tam gün ya da yarım gün olabilir. Eğer okulda normal öğretim uygulanıyorsa yarım gün, ikili öğretim uygulanıyorsa tam gün olarak devamsızlık bilgileri işlenir.

Devamsızlık Sınırını Aşanlar

Milli Eğitim Bakanlığı tarafından belirlenen devamsızlık sınırına göre öğrencilerin devamsızlık bilgilerinin kontrolü yapılmalıdır. Bu aşamada eğitimin aksamaması için okul yönetimi tarafından belirlenen kadar devamsızlık yapan öğrenciler önce sözlü olarak uyarılır, devamsızlık eylemi tekrarlanınca öğrenci velisine yazılı bir metin gönderilir.

Yıllık Devamsızlık Defteri

Yönetmelik gereği öğretim yılı sonunda devamsızlık defterinin tutulması gerekmektedir. Devamsızlık defterinde her bir sınıfta ki öğrencilerin yıl içerisinde yapmış olduğu devamsızlıkların toplamı yazılır.

3.3 Listeler

Yüzlerce ve hatta binlerce öğrencinin öğrenim gördüğü bir okulda yapılması gereken işlerin daha pratik ve hızlı yapılabilmesi için belli bir sistematikte bazı listelerin hazırlanması gerekir.

Sınıf Listesi

Bilindiği üzere tüm okullarda öğrenciler bir sınıfın üyesi olarak öğrenim hayatlarını sürdürürler. Herhangi bir sınıfta ki öğrenciler ile ilgili bir işlemin yapılabilmesi için öncelikle sınıf listesinin oluşturulması gerekir.

Alan Bölüme Göre Öğrenci Listesi

Sınıf bazında işlem yapılabilceği gibi orta öğretim öğrencileri üzerinde seçmiş olduğu alana göre ya da meslek liseleri için öğrencilerin devam ettiği bölüme göre listelenmesi gerekebilir.

Ödev Alanlar Almayanlar Listesi

İlköğretim ikinci kademe ve orta öğretimde öğrenimine devam eden öğrenciler en az bir dersten yıllık ödev almak zorundadır. Öğrencilerin yıllık ödev alıp almadıkları ve hangi dersten yıllık ödev aldığı bilgileri tutulmalıdır.

Öğrenci Nüfus Bilgileri

Öğrencilerin nüfus bilgilerinin nüfus sureti örneğine göre okul içersinde kayda alınması gereklidir. Bu işlemler yapılırken sınıf bazında öğrenci listesi göz önüne alınarak yapılması, işlemlerin biraz daha sistematik bir şekilde yapılmasını sağlayacaktır.

Öğrenci Hasta Sevk Kâğıdı

Öğrenci bazında alınması gereken bir listedir. Bu nedenle öğrencinin bulunduğu sınıf belirlenerek "Sayı" ve "Kurum" (Öğrencinin gönderileceği sağlık kurumu) bilgileri doldurulur.

Öğrenci Durum Formu

Öğrenci ve velisi hakkında bilgiler tutulur. Sınıf bazında işlem yapılır. Gerekğinde yazılı evrak olarak alınması durumunda evrağın kim tarafından onaylanacağı belirtilir.

Ara Karne

Belirli zamanlarda öğrencilere veya velilerine ara karnelerin verilmesi gerekebilir. Öğrenci bazında alınacak bir evraktır.

Askerlik Belgesi

Askerlik çağına gelmiş öğrencilere verilen belgedir.

Resmi Belgeler

Öğrenci belgesi, öğrenim belgesi, öğrenci nakil belgesi, tasdikname, öğrenci kütük defteri gibi resmi evrak durumunda ki belgelerin tutulması zorunludur.

Öğrenci Belgesi

Öğrencinin okulun öğrencisi olduğuna dair bir belgedir. Öğrenci bazında hazırlanacak bir belge olduğu için sınıf listeleri göz önüne alınarak işlemler yapılır. Evrakın kim tarafından onaylanacağı belirtilir.

Öğrenim Belgesi

Öğrenci öğrenim çağı dışına çıktığında veya velisinin isteği ile okuldan ayrıldığında verilen belgedir. Öğrenci bazında hazırlanacak bir belge olduğu için sınıf listeleri göz önüne alınarak işlemler yapılır.

Nakil İçin Öğrenim Belgesi

Nakil isteği yapan öğrenciye veya öğrencinin gittiği okula verilen belgedir. Öğrenci bazında hazırlanacak bir belge olduğu için sınıf listeleri göz önüne alınarak işlemler yapılır. Evrakın kim tarafından onaylanacağı belirtilir.

Tasdikname

Nakil isteği yapan öğrenciye veya öğrencinin gittiği okula verilen belgedir

Dosya Öğrenim Belgesi İsteme

Nakil gelen öğrencinin geldiği okulundan dosyasını ve öğrenim belgesini istemek için hazırlanan belgedir. Öğrenci bazında hazırlanacak bir belge olduğu için sınıf listeleri göz önüne alınarak işlemler yapılır. Evrakın kim tarafından onaylanacağı belirtilir.

Dosya Öğrenim Belgesi Gönderme

Nakil giden öğrencinin gittiği okula dosyasını ve öğrenim belgesini göndermek için çıkarılan belgedir. Öğrenci bazında hazırlanacak bir belge olduğu için sınıf listeleri göz önüne alınarak işlemler yapılır. Evrakın kim tarafından onaylanacağı belirtilir.

Öğrenim Belgesini Kaybedenlere Mahsus Belge

Öğrenim belgesini kaybeden öğrenciye bir defaya mahsus verilen belgedir. Öğrenci bazında hazırlanacak bir belge olduğu için sınıf listeleri göz önüne alınarak işlemler yapılır.

Diplomasını Kaybedenlere Mahsus Belge

Diplomasını kaybeden öğrenciye bir defaya mahsus verilen belgedir. Öğrenci bazında hazırlanacak bir belge olduğu için sınıf listeleri göz önüne alınarak işlemler yapılır. Evrakın kim tarafından onaylanacağı belirtilir.

Tasdiknamesini Kaybedenlere Mahsus Belge

Diplomasını kaybeden öğrenciye bir defaya mahsus verilen belgedir. Öğrenci bazında hazırlanacak bir belge olduğu için sınıf listeleri göz önüne alınarak işlemler yapılır. Evrakın kim tarafından onaylanacağı belirtilir.

Öğrenci Kütük Defteri

Öğrencilerin kütük bilgilerinin dökümünün alındığı bir belgedir. Öğrenci bazında hazırlanacak bir belge olduğu için sınıf listeleri göz önüne alınarak işlemler yapılır.

3.4 Yıl Sonu İşlemleri

Yılsonunda (İkinci s mestri sonu) ve birinci d nem sonunda yapılması gereken bazı işlemler vardır.

Yıllık Başarı Ortalamaları Hesaplama

 ğrencilerin yıl i ersinde aldığı notlarının yılsonunda ortalamasının alınması gerekir. Bu işlem sınıf bazında yapılır.

Karne

 ğrencilere verilen karnelerin basımı yapılmalıdır. Bu bağlamda sınıf bazında veya  ğrenci bazında karne basımı yapılabilir.

Takdirname Belgesi

Not ortalaması tutan  ğrencilere verilmesi gereken bir belgedir. Bu b l mde sınıf bazında veya  ğrenci bazında "Takdirname belgesi" hazırlanabilir.

Teşekk r Belgesi

Not ortalaması tutan  ğrencilere verilmesi gereken bir belgedir. Bu b l mde sınıf bazında veya  ğrenci bazında " Teşekk r Belgesi " hazırlanabilir.

Onur Belgesi

Ders başarı durumu ve disiplin notları g z  n nde bulundurularak hazırlanan bir belgedir. Sınıf bazında veya  ğrenci bazında “Onur belgesi” hazırlanabilir.

Sınıf Ge me Defterleri

Yılsonlarında  ğrencinin sınıfını ge tiğini veya kaldığını belgeleyen sınıf ge me defterleri d zenlenir. Bu işlem sınıf bazında veya  ğrenci bazında yapılabilir. "Yılsonu Derecesi" ağırlıklı ya da aritmetik ortalama olarak hazırlanır.

Ge enler / Kalanlar Listesi

Yılsonunda ge enler ve kalanların listelenmesi gerekir.

3.5 Mezuniyet ve Diploma

Yılsonlarında  ğrencilerin mezuniyet ve diploma işlemlerinin yapılması gerekir.

Mezuniyete Hak Kazananlar Listesi

Mezuniyete hak kazanan öğrenciler belirlenerek, listelenir. Bu listede diploma tarih ve numarası belirlenir

Mezun Olamayan Öğrenciler

Mezuniyete hak kazanan öğrenciler gibi, öğrenim süresinde mezun olamayan öğrencilerin de listesinin, öğrencinin mezun olamama sebepleri ile birlikte tutulması gerekir.

3.6 İstatistikler

Okul tarafından Milli Eğitim Müdürlüklerine gönderilmesi zorunlu olan bazı istatistikî bilgiler vardır. Bunları şöyle sıralayabiliriz.

Şube ve Öğrenci Sayısı

Şube ve öğrenci sayıları detaylı olarak listelenerek Milli Eğitim Müdürlüklerine gönderilir.

Yaş ve Cinsiyet Analizi

Okuldaki tüm öğrencilere ait yaş ve cinsiyet analizi yapıp, bu analiz sonuçları "Liste Tarihi" bilgisiyle Milli Eğitim Müdürlüklerine gönderilir.

Yeni Kayıt Olanlar

Belirtilen kayıt yılına göre yeni kayıt olan öğrencilerin sayısı listelenmelidir. Sınıf bazında istatistik hazırlanır. "Kayıt Yılı" ve "Liste Tarihi" bilgileri listede olmalıdır.

Burslu Ve Yatılı Öğrenciler

Burslu ve yatılı öğrencilerin sayısı belirlenmelidir. Sınıf bazında istatistik hazırlanır.

Yabancı Dillerine Göre Öğrenci Sayıları

Yabancı dillere göre öğrenci sayısı belirlenmelidir.

Not Ortalamasına Göre Öğrenci Listesi

Sınıf bazında not ortalamasına göre öğrenci istatistiği hazırlanır.

Geçenler Kalanlar Sayısı

Yılsonlarında geçen ve kalan öğrencilerin sayı olarak analizi yapılır. Okul bazında bu analiz yapılarak ilgili birimlere gönderilir.

Okulla İlişği Kesilenler

Herhangi bir sebeple okulla ilişği kesilen öğrencilerin sayı olarak analizi yapılır.

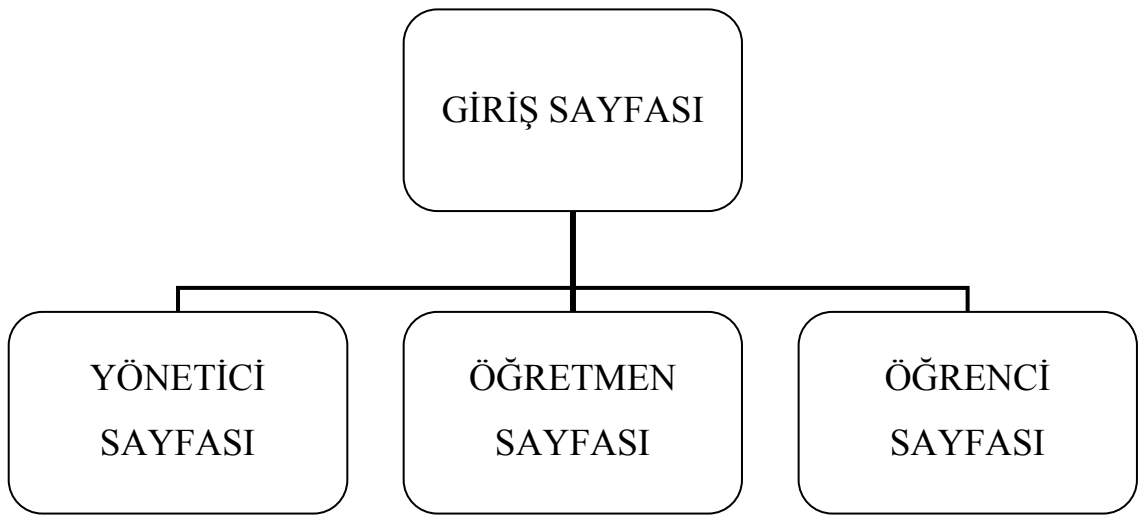
3.7 Sınavlar

Bu aşamada dönem içi, dönem sonu ve yılsonunda yapılan

1. Ortalama Yükseltme Sınavı,
2. Seviye Tespit Sınavları ve
3. Sorumluluk Sınavları ile ilgili giriş, not girişı ve gereken tüm yazılı belgeleri hazırlanır.

4. GELİŞTİRİLEN OTOMASYON SİSTEMİNİN YAPISI

Geliştirilen “Öğrenci İşleri Otomasyonu Programı” ikinci bölümde incelenen web programlama teknikleri ile geliştirilen uygulamaların performans analizi sonucunda elde edilen bulgular ışığında PHP web programlama tekniği kullanılarak hazırlanmıştır. PHP ile birlikte veritabanı işlemleri için MySQL kullanılmıştır ve tasarım aşamasında uygulamalar, yerel olarak Apache web sunucu üzerinde denenmiştir. Geliştirilen “Öğrenci İşleri Otomasyonu Programında” üç farklı kullanıcı grubu tanımlanmıştır. Bunlar aşağıda ki şekilden görüleceği gibi “yönetici”, “öğretmen” ve “öğrenci” dir.

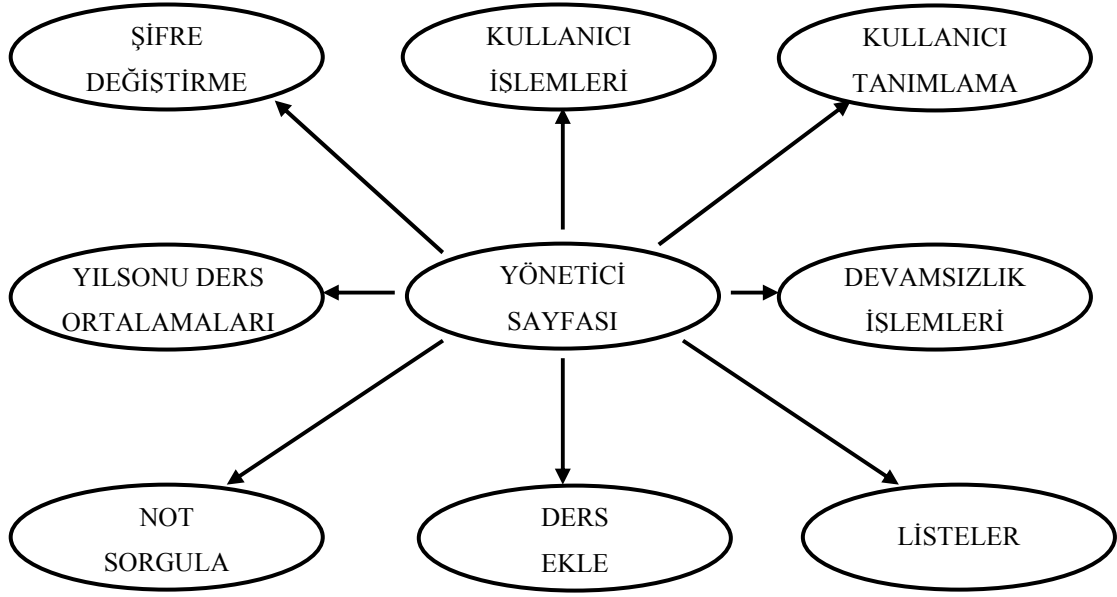


Şekil 4.1 Öğrenci İşleri Otomasyonu Giriş Sayfası

Sisteme giriş için öncelikle kullanıcı, yönetici tarafından belirlenen kullanıcı adı ve parolasını giriş ekranına girmelidir.

4.1 Yönetici Sayfası

Sisteme yönetici haklarıyla giriş yapan kullanıcı şu işlemleri yapabilir.



Şekil 4.2 Yönetici Sayfası

Kullanıcının giriş sayfasında, sisteme giriş yapabilmesi için geçerli bir kullanıcı adı ve parolası olması gerekir. Kullanıcı adı ve parola bilgileri doğru girilince kullanıcı, program tarafından yetkilendirildiği sayfaya yönlendirilir. Bu işlem aşağıda verilen kodlar ile yapılır.

Bu bölümde, Şekil 4.2’de gösterilen “Yönetici Sayfasında” yapılabilecek işlemlerden sadece önemli olanları, icra edilen komutlarla birlikte yordamsal bir yaklaşımla verilmiştir.

<pre> mysql_select_db(\$dbname) or die("Tabloya ulaşamadım!"); \$sor=@mysql_query("select * from login"); \$abdn=mysql_result(\$sor,\$say,"no"); \$abdi=mysql_result(\$sor,\$say,"kul_adi"); \$abds=mysql_result(\$sor,\$say,"sifre"); \$abdc=mysql_result(\$sor,\$say,"ogrenci"); if((\$name==\$abdi)&&(\$pass==\$abds)) { \$durum=true; \$code=\$abdc; \$usno=\$abdn; } \$say++; } if(\$durum) { if(\$code=='2') { header("Location:tyor.php"); exit(); } elseif(\$code=='0') { header("Location:ogretmen.php"); exit(); } elseif(\$code=='1') { header("Location:ogrenci.php"); exit(); } } else { echo "Hatalı kullanıcı adı,şifre veya sisteme kayıtlı olmayan kullanıcı olduğunuz tespit edildi"; } </pre>	} Giriş sayfasında ki kullanıcı adı ve parola, veritabanına kayıtlı bilgilerle karşılaştırılır. } Girilen bilgiler doğru ise kullanıcı sınıfı belirlenir. } Kullanıcı adı ve parolası yöneticiye ait ise yönetici sayfasına yönlendirilir. } Kullanıcı adı ve parolası öğretmene ait ise öğretmen sayfasına yönlendirilir. } Kullanıcı adı ve parolası öğrenciye ait ise öğrenci sayfasına yönlendirilir. } Kullanıcı adı ve parolası geçersizse sisteme girilemeyeceğine dair mesaj görüntülenir
--	---

Bu şekilde kullanıcı adı ve parolası kontrol edilen kullanıcının, kullanıcı sınıfının tespit edilebilmesi için icra edilen kodlardan anlaşılacağı üzere; kullanıcı gurupları 0,1 ve 2 rakamlarıyla kodlanmıştır. Yöneticiler 2, öğretmenler 0 ve öğrenci-veliler 1 kodu ile sistemde

tanımlıdır. Bu kodlar programın yapım aşamasında yönetici tarafından her bir kullanıcı için belirlenir.

4.1.1 Şifre İşlemleri

Daha çok güvenlik sebebiyle gereksinim duyulan şifre değiştirme modülünde işletilen kodlar şöyledir:

```
mysql_select_db($dbname) or die("Tabloya ulaşamadım!");
mysql_select_db($dbname) or die("Veritabanına ulaşamıyor!");
if(!$sr=mysql_query("select sifre from login where no='$no'"))
{echo "Tabloya ulaşamıyor ";exit();}
$sifre=mysql_result($sr,0,"sifre");
if ($B1){if($s1!=$sifre || $s2!=$s3 || (strlen($s1)<4) ||
(strlen($s2)<4) || $s2=="")}{echo " Hatalı giriş yaptınız!
Lütfen formu dikkatli doldurunuz.";?><?exit(); }
elseif(!mysql_query("UPDATE login SET sifre='$s3'
where no='$no' and sifre='$sifre'"))
{echo "<b>Şifreniz güncellenemedi</b>";?><?exit();}?>
TEBRİKLER ! ŞİFRENİZ YENİLENDİ
<?exit(); }?>
```

Veritabanı test ediliyor.

Girilen bilgiler veritabanı ile karşılaştırılıyor

Kullanıcı tarafından belirlenen yeni şifre veritabanına kaydediliyor

4.1.2 Kullanıcı İşlemleri

Bu başlık altında kullanıcılarla ilgili; şifre değiştirme, kullanıcı adı değiştirme ve kullanıcı silme işlemleri yapılır.

4.1.3 Kullanıcı Tanımlama

Geliştirilen “Öğrenci İşleri Otomasyonu Programında” ki en önemli modüllerden biridir. Programın kullanıldığı okulda kullanıcıların programa tanıtıldığı aşamadır. Programda tanımlı üç farklı kullanıcı gurubu vardır. İşte programda tanımlı kullanıcı guruplarına yeni kullanıcı ekleme bu modülde yapılır. Kullanıcı gurubuna üye yapılan kullanıcı, aynı zamanda sisteme de tanıtılmış olur. Bu aşamada işlem gören kodlar şunlardır:

<pre> <? if (\$B1){ if(!\$no !\$kul_adi !\$sifre) {echo "Lütfen tüm alanları doldurunuz!"; \$B1=0;\$B2=1;\$cc=0;?> <?} if(!mysql_query("select * from login where no='\$no'")) {echo "Bu kullanıcı daha önce sisteme eklenmiş"; \$B1=0;\$B2=1;\$cc=0;} if(\$cc){ if(!mysql_query("insert into login (kul_adi,sifre,ogrenci) values('\$kul_adi','\$sifre','\$ogrenci)")) {echo "tabloya veri eklenemiyor";\$B1=0;\$B2=1;\$cc=0;} else {echo "Kullanıcı sisteme başarıyla eklendi"; ?> </pre>	{	HTML kodlarıyla oluşturulan tablo ile alınan bilgiler veritabanı ile karşılaştırılarak önceden aynı kullanıcının eklenmiş olup olmadığı kontrol edilir.
<pre> </pre>	}	Kullanıcı ekleme modülünde girilen kullanıcı adı, şifre ve kullanıcı sınıfının belirleneceği bilgiler veritabanına kaydediliyor.

4.1.4 Ders Tanımlama

Yukarıda anlatılan “Kullanıcı Tanımlama” modülü kadar önem arz etmektedir. Bu aşamada onlarca derse, onlarca öğretmenden hangi öğretmenin gireceği belirlenecektir. Yani hangi derse hangi öğretmenin not girişi yapabileceği belirlenir. Öncelikle HTML kodlarıyla hazırlanan tablodan dersin kodu, dersin adı, haftalık ders saat sayısı, okutulduğu sınıf ve derse girecek öğretmenin programa kayıtlı numarası bilgileri alınarak veritabanına girilir. Bu aşamada işlem gören kodlar şunlardır:

```

<?
if($B1){
if(!$kod || !$ders_adi || !$bolum || !$kredi)
{echo "Lütfen tüm alanları doldurunuz! ";
$B1=0;$B2=1;$cc=0;?> <?}
if(!$ogrtno)
{echo " Bu ders için herhangi bir öğretmen yetkilendirilmemis";}
if(!mysql_query("select * from ders where ders_kodu='$kod'"))
{echo "Bu ders daha önce sisteme eklenmiş";
$B1=0;$B2=1;$cc=0;}
if($cc){
if(!mysql_query
("insert into ders (ders_kodu,adi,bolum,kredi,ogrtno,sinif)
values('$kod','$ders_adi','$bolum','$kredi','$ogrtno','$sinif')"))
{echo "tabloya veri eklenemiyor";
$B1=0;$B2=1;$cc=0;}
else
{echo "Ders sisteme başarıyla eklendi";?>
<?
    $B1=0;$B2=1;$cc=0;}
    }
} else{
?>

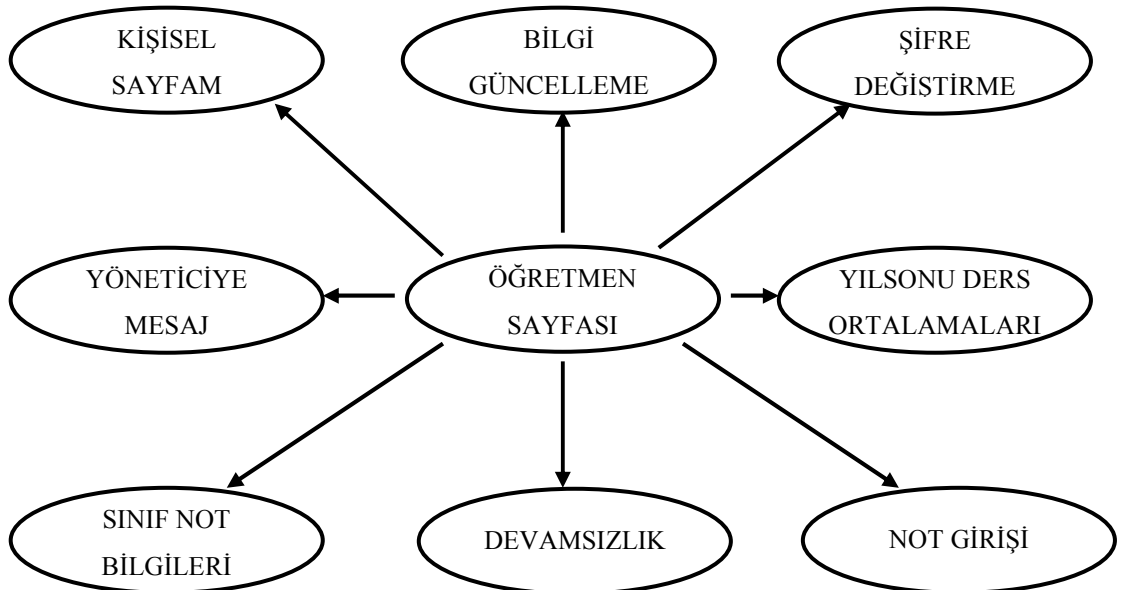
```

HTML kodlarıyla alınan bilgiler veritabanına kayıtlı bilgiler ile karşılaştırılarak dersin eklenmiş olup olmadığı ve derse yetkilendirilen öğretmen olup olmadığı kontrol edilir.

HTML kodlarıyla oluşturulan form ile alınan bilgilerin geçerli olduğu tespit edilince bu bilgiler veritabanına kaydedilerek işlem tamamlanır.

4.2 Öğretmen Sayfası

Sisteme öğretmen haklarıyla giriş yapan kullanıcı şu işlemleri yapabilir.



Şekil 4.3 Öğretmen Sayfası

Programın işleyişi açısından, “Öğretmen Sayfasında” en önemli nokta “Not Girişi” modülüdür. Be nedenle “Öğretmen Sayfasında” sadece not girişi modülünde icra edilen kodlar incelenecektir. Kodların daha anlaşılır bir şekilde incelenebilmesi için; kodlar işlevine göre numaralandırılarak açıklanmıştır.

4.2.1 Not Girişi

```

<?include "lib.inc";
$no=$HTTP_COOKIE_VARS["no"];
/*eğer çerez atılmamış ise kullanıcının siteye girmesi engelleniyor.*/?>
<? if(!$sr=mysql_query("select * from ogretmen where no='$no'"))
{echo "Tabloya ulaşılamıyor";exit();}
if(@mysql_num_rows($sr))
{ $ad=mysql_result($sr,0,"adi");
$soyad=mysql_result($sr,0,"soyadi");
echo "Sayın $ad $soyad not giriş modülüne hoş geldiniz ..."; }?>
<?$dgoster=(@mysql_query("select * from ders where ogrtno='$no'"));
while ($sira=mysql_fetch_array ($dgoster)){
echo .$sira["adi"];}?>
<form name="form1" id="form1" method="post" action="ngir1.php">
<select name="derss">
<option selected>Bir ders seçiniz...</option>
<?$dgoster=(@mysql_query("select * from ders where ogrtno='$no'"));
while ($sira=mysql_fetch_array ($dgoster))
{echo "<option>". "$sira["adi"].</option>";}?>>> </select>
<select name="class"> <?
$dgoster=(@mysql_query("select * from ders where ogrtno='$no'"));
while ($sira=mysql_fetch_array ($dgoster))
{echo "<option>". "$sira["sinif"].</option>";}?>>> </select>
<input type="submit" value="Gönder" name="B1" >

```

1

2

3

4

Yukarıda numaralandırılarak gösterilen kod kısımlarının işlevi:

1. Programa giriş sayfasında kullanıcı adı ve parolası girildikten sonra, program tarafından, uzaktan erişimi sağlayan bilgisayara kullanıcı bilgilerini içeren bir çerez gönderilir. 1. grup kod bloğunda; uzaktan erişimi sağlayan bilgisayarda kullanıcı bilgilerini içeren çerez olup olmadığı kontrol edilir ve bu bilgisayarda çerez yoksa, programın kullanılması engellenir.

2. “Öğrenci İşleri Otomasyonu” programı giriş sayfasında, kullanıcı adı ve parolasını girerek not girişi yapmak isteyen kullanıcı, giriş sayfasında kullandığı bilgilerden, kullanıcı adı bilgisi, ilgili veritabanı dosyası ile karşılaştırılarak, bu veritabanı dosyasından kullanıcının adı

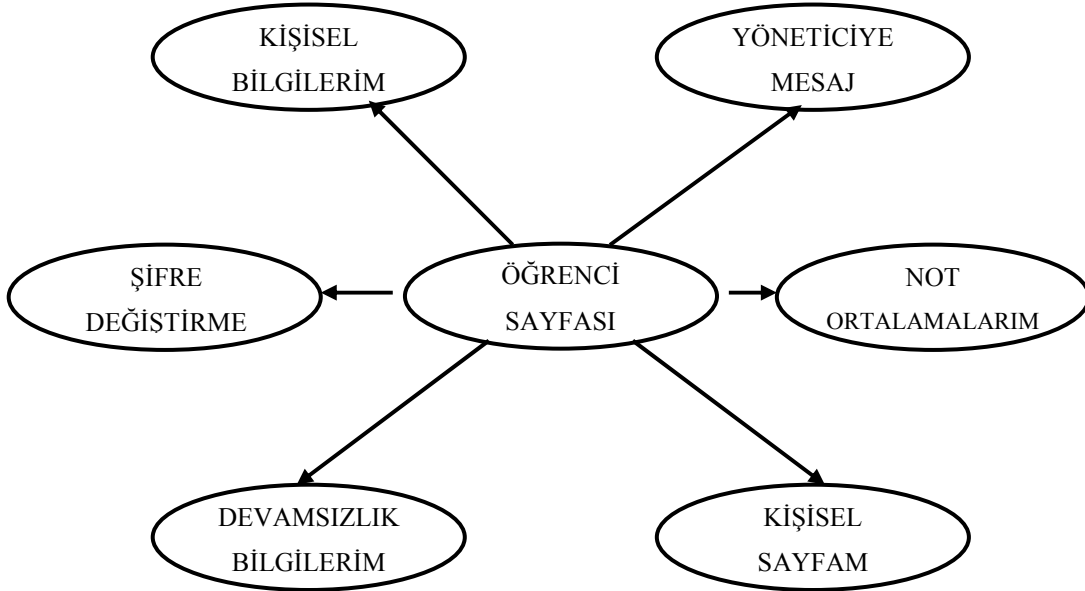
ve soyadı bilgileri alınır. Bu aşamada kullanıcı “Not Girişi” ekranında sisteme kayıtlı adı ve soyadı ile karşılaşır.

3. Bu kod bloğunda adı ve soyadı ile karşılanan kullanıcı, sadece “Ders Tanımlama” modülünde yetkilendirildiği dersler ile bu derslerin okutulduğu sınıf bilgilerini içeren açılan kutuları da görecektir. Yani bu aşamada kullanıcı okulda öğretimi yapılan tüm dersleri değil, sadece yetkili kılındığı dersleri görecektir.

4. Bir önceki kod bloğunda kullanıcı, açılan kutulardan seçtiği ders ve sınıf bilgileri ile kullanıcı bilgisi, ilgili veritabanı dosyasına işlenerek kullanıcının not girişini yapması sağlanır.

4.3 Öğrenci Sayfası

Sisteme öğrenci haklarıyla giriş yapan kullanıcı şu işlemleri yapabilir.



Şekil 4.4 Öğrenci Sayfası

Öğrenci sayfasında, her üç kullanıcı grubu sayfasına da yerleştirilmiş olan Kişisel sayfam modülünde işletilen kodlar incelenecektir. Kodların yeterince açıklanabilmesi için bu bölümde de kodlar yaptıkları işlere göre gruplandırılarak açıklanacaktır.

4.3.1 Kişisel Sayfam

```
<?
include("lib.inc");
mysql_select_db($dbname) or die("Tabloya ulaşamadım!");
$dosya_dizin = "$yol_dizin/img/ogren";
$dosya_url = "$yol_url/img/ogren";
$no=$HTTP_COOKIE_VARS["no"];
if(!$sr=mysql_query("select * from ogrenci where no='$no'"))
{echo "Tabloya ulaşamıyor";exit();}

$sira=mysql_fetch_array($sr);
$kadi=$sira["kadi"];
$adi=$sira["adi"];
$soyadi=$sira["soyadi"];
$email=$sira["email"];
$evtel=$sira["evtel"];
$ceptel=$sira["ceptel"];
$adres=$sira["adres"];
$resim=$sira["resim"];
$dogtar=$sira["dogtar"];
$dogyer=$sira["dogyer"];
$anaadi=$sira["anaadi"];
$babaadi=$sira["babaadi"];
$biyog=$sira["biyog"];
$hobi=$sira["hobi"];
$kimlikno=$sira["kimlikno"];

?>
<? if (!$resim) { echo ("Sisteme yüklenmiş resim yok");}
else { print ("<img src=\""$dosya_url/$resim
E-Posta Adresi<?echo $email; ?>
...
...
```

Burada öğrenci bilgilerinin tutulduğu veritabanı dosyası test edilerek, uzaktan erişim yapılan bilgisayara çerez gönderilir.

Bu blokta işletilen kodlar ile “Öğrenci İşleri Otomasyonu” programına giriş yapan öğrencinin kişisel bilgileri, ilgili veritabanı dosyası ile bağlantı kurulup, elde edilen bilgiler değişkenlere atanır.

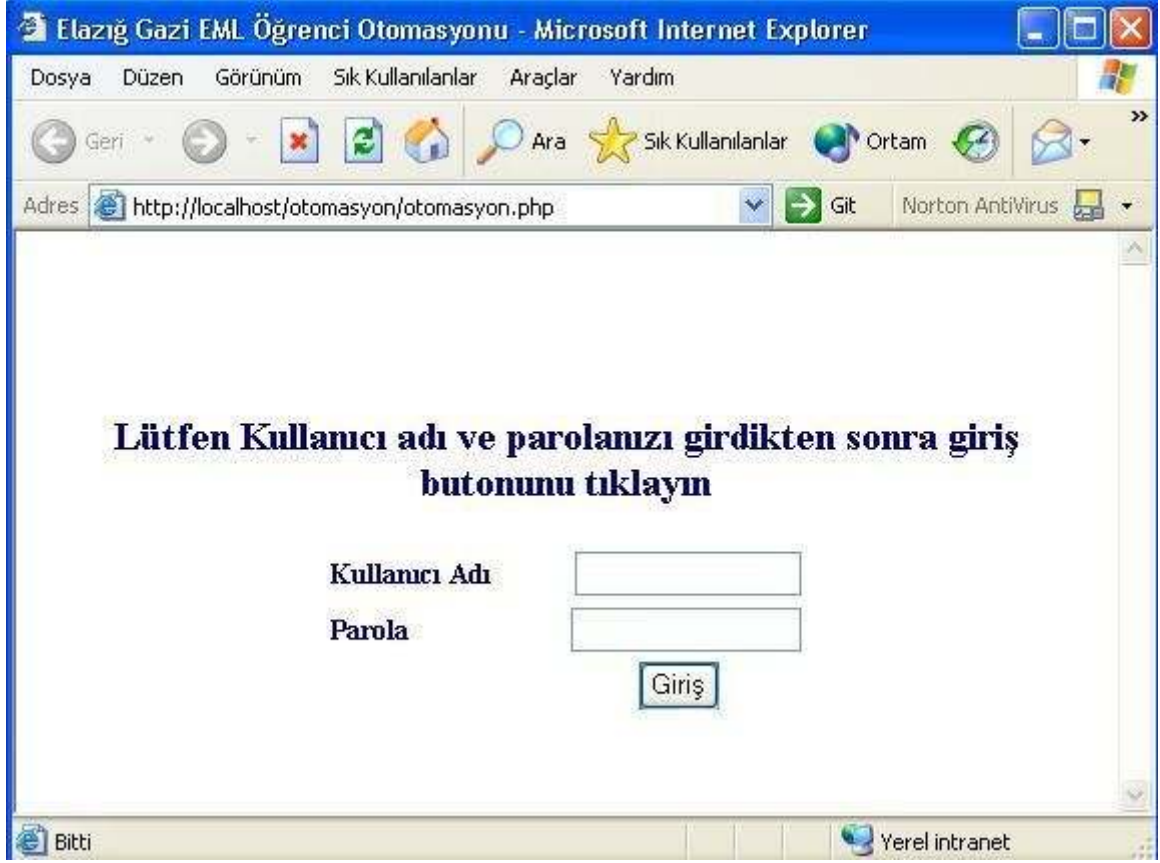
Bir önceki blokta değişkenlere atanan değerler HTML kodlarıyla oluşturulan tabloda hücrelere eklenerek görüntülenir

5. GELİŞTİRİLEN OTOMASYON PROGRAMININ KULLANIMI

Öğrenci işleri otomasyon programı, 3. bölümde anlatılan ve her biri idareciler tarafından dikkatle ve özenle yapılması gereken işlem basamaklarından önemli olanlarını bilgisayar ortamına taşımak, idarecilerin ve öğretmenlerin işlerini bir nebze olsun kolaylaştırmak amacıyla hazırlanmıştır. Sisteme girişi yapılması gereken tüm bilgiler zaman ve mekândan bağımsız olarak sisteme girilebilmektedir. Bu aşamada sadece internet erişimi olan bilgisayarın olması yeterlidir. Bilgi girişi gibi öğrencilerin not ve devamsızlık bilgilerini, öğrenci ve öğretmenlerin sistemde kayıtlı bilgilerini görebilmek / güncelleyebilmek ve yönetici tarafından ders ve kullanıcı tanımlama gibi işlem basamaklarının yapılabilmesi için de sadece internet erişimi olan bir bilgisayarın olması yeterlidir. Buradan da anlaşılacağı üzere sisteme her türlü bilgi girişi ve bu bilgilerin görüntülenmesi için herhangi bir ortamda ve bir bilgisayarın olması yeterlidir.

5.1 Kullanıcı Sınıfları

Geliştirilen otomasyon programında üç farklı kullanıcı gurubu tanımlanmıştır. Programda tanımlı olan kullanıcılar; öğrenci, öğretmen ya da yönetici gurubunda olabilirler. Her gurubun üyeleri farklı yetkilerle donatılmıştır. Kullanıcıların zaman ve mekândan bağımsız olarak sisteme giriş yapabilmesi için yönetici tarafından tanımlanmış bir kullanıcı olması gereklidir. Bu bağlamda kullanıcıların yönetici tarafından belirlenmiş bir kullanıcı adı ve parolası olmalıdır. Yönetici tarafından belirlenen kullanıcı adı sadece yönetici tarafından değiştirilebileceği gibi kullanıcı isterse parolasını değiştirebilir.



Şekil 5.1 Otomasyon Ana Sayfası

Programa giriş yapacak kullanıcılar; Şekil 5.1’de görülen otomasyon ana sayfasında kullanıcı adı ve parolasını doğru olarak girmek zorundadır. Giriş başarılı bir şekilde gerçekleştirilince kullanıcı, üyesi olduğu grubun yetkilerine göre kişisel sayfasına yönlendirilir. Tanımlanmış kullanıcı grupları şunlardır:

5.1.1 Yönetici Olan Kullanıcılar

Programın organizasyonu, programla ilgili tüm yetkilendirme ve kullanıcı tanımlamaları gibi tüm yönetimsel işlemler, yönetimsel işlemler yönetici grubunda ki kullanıcılar tarafından yapılır. Bu grupta ki kullanıcıların yapabilecekleri işlemler Şekil 5.2’de görüleceği üzere şunlardır:

1. **Şifre Değişirme:** Sisteme yönetici olarak tanımlanmış kullanıcı şifresini değiştirmek için bu adımı izleyecektir. Daha çok güvenlik sebeplerinden dolayı yapılması gereken bir işlemdir. Belirlenecek yeni şifre için sadece dört karakterden oluşma şartı getirilmiştir.



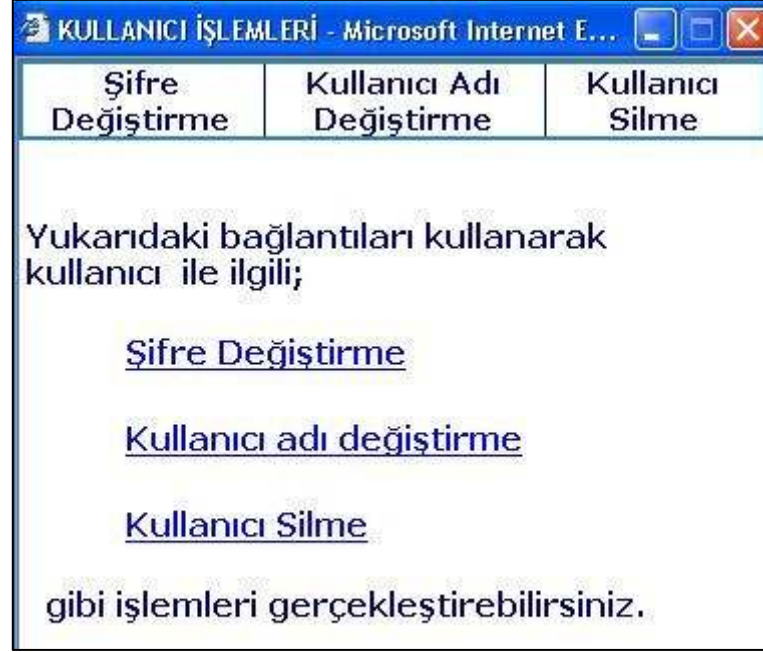
Şekil 5.2 Yönetici Sayfası

Şekil 5.3 Şifre Değişirme Ekranı

Bu aşamada Şekil 5.3'te görüldüğü üzere yönetici eski şifresi ile yeni şifresini iki defa girip sayfada ki gönder butonunu tıklayınca şifresini değiştirebilecektir.

2. Kullanıcı İşlemleri: Bu başlık altında kullanıcılar Şekil 5.4'te görüldüğü gibi; şifre değiştirme, kullanıcı adı değiştirme ve kullanıcı silme işlemleri yapılabilir.

a. Şifre değiştirme: Güvenlik sebeplerinden dolayı izlenecek bir yoldur. Kullanıcı şifresini unutunca ya da birileri tarafından ele geçirilince yöneticiler tarafından yapılır.



Şekil 5.4 Kullanıcı İşlemleri Penceresi

Şekil 5.5 Kullanıcı Şifresi Değiştirme Penceresi

Bu aşamada yönetici Şekil 5.5'te görüldüğü üzere şifresi değiştirilecek kullanıcının sistemde kayıtlı olan kullanıcı adı ile şifresini girdikten sonra yeni şifresini iki kez girerek işlem tamamlanır.

b. Kullanıcı adı değiştirme: Program hazırlanma aşamasında tüm kullanıcıların kullanıcı adları yönetici tarafından belirli bir sistematığe göre belirlenmiştir. Kullanıcılar kendilerini ifade edecek tarzda bir kullanıcı adı ile sisteme giriş yapabilirler.

Bu amaçla Şekil 5.6'da görülen “Kullanıcı Adı Değiştirme” ekranında önce eski kullanıcı adı ardından yeni kullanıcı adı girilerek işlem tamamlanır.

c. Kullanıcı silme: Örneğin mezun olmuş öğrencilerin ya da okuldan ayrılmış öğretmenlerin artık “Öğrenci İşleri Otomasyonu Programını” kullanmalarına gerek kalmayacaktır.

Bu durumlarda programa kayıtlı kullanıcılardan bazılarının silinmesi gerekebilir. Bunun için Şekil 5.7'de görülen “Kullanıcı Silme” ekranında programdan silinecek kullanıcının kullanıcı adı girilip gönder butonu tıklanarak işlem tamamlanır.



Şekil 5.6 Kullanıcı Adı Değiştirme Penceresi

3. **Kullanıcı Tanımlama:** “Öğrenci İşleri Otomasyonu Programının” uygulandığı okulda yeni öğrenci kaydı yapılması ya da okula yeni bir öğretmenin gelmesi durumunda programı kullanabilmesi, programa tanıtılması gerekir. Yani yeni kullanıcı kaydı yapmak için kullanılır.

Bu işlemde izlenecek yol; Şekil 5.8’de görülen “Kullanıcı Ekleme” sayfasında kullanıcı adı, kullanıcının sisteme girişi esnasında kullanacağı şifre ve dahil edileceği grup bilgileri girmek olacaktır. Bu bilgiler girildikten sonra “Kullanıcı Ekleme” sayfasında gönder butonu tıklanarak yeni kullanıcı oluşturulmuş olur.



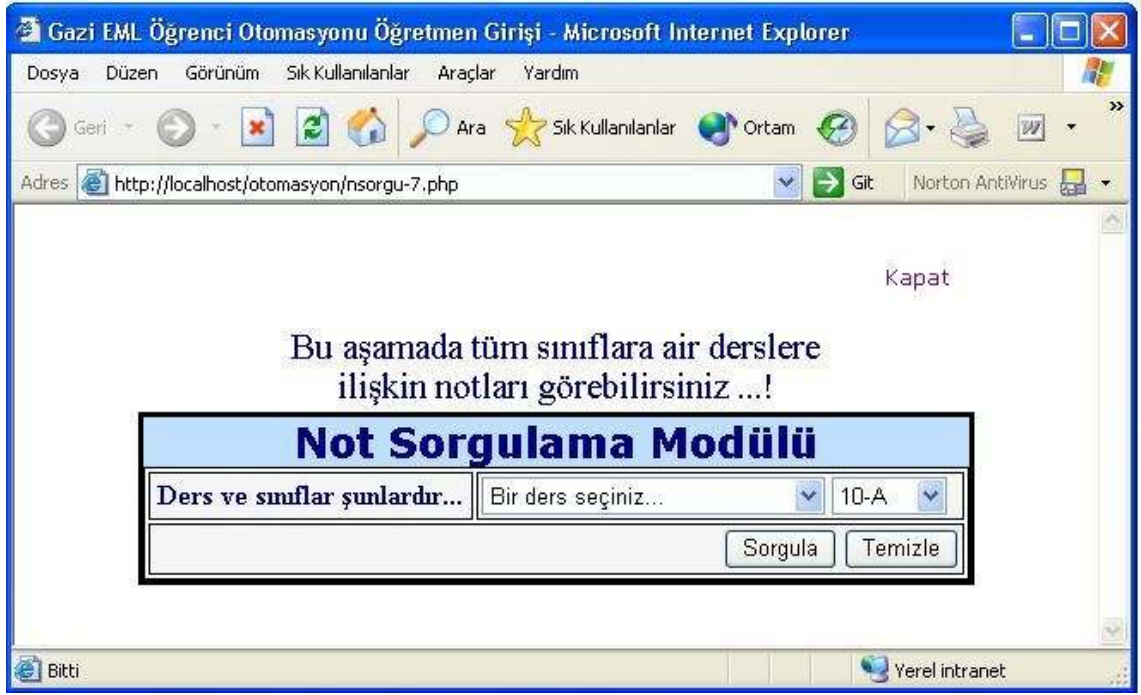
Şekil 5.7 Kullanıcı Silme Penceresi



Şekil 5.8 Kullanıcı Ekleme Ekranı

4. **Yılsonu Ders Ortalamaları:** Bu aşamada “Öğrenci İşleri Otomasyonu Programına” yönetici haklarıyla giren kullanıcının tüm öğrencilere ait yılsonu ders ortalamalarını görebilmesi amaçlanmıştır. Şekil 5.9’da görüldüğü gibi kullanıcı önce programa girişi yapılan derslerden birini, sonra mevcut sınıflardan birini seçerek sorgulama işlemini yapabilir.

5. **Devamsızlık İşlemleri:** Devamsızlık işlemleri başlığı altında yapılabilecek üç işlem vardır. Bunlar; “Devamsızlık Girişi”, “Sınıf Bazında Devamsızlık” ve “Öğrenci Bazında Devamsızlıktır”.



Şekil 5.9 Yılsonu Ders Ortalamaları



Şekil 5.10 Devamsızlık İşlemleri

a. Devamsızlık girişi: Bu aşamada sisteme yönetici haklarıyla giren kullanıcı; öğrenci devamsızlıklarını kaydeder. Bu işlem için Şekil 5.11’de görüldüğü gibi öğrenci numarası, sınıfı, adı ve soyadı, devamsızlık ile devamsızlığın yapıldığı tarih bilgileri girilir. Tabi bu bilgilerin günlük girilmesi hem sistemden sorgulama yapacak kullanıcıların doğru bilgiye ulaşmasını sağlayacak hem de giriş işlerinin birikmesinin önüne geçilerek işler daha rahat yürütülebilir olacaktır.

b. Sınıf bazında devamsızlık: Öğretim yılı boyunca günlük olarak girilen devamsızlık bilgilerine ulaşmak için programa dahil edilen bir modüldür. Yönetici devamsızlık bilgisini almak istediği sınıfı seçerek, seçilen sınıfa ait devamsızlık bilgisini görüntüleyebilir.

Pencereyi Kapat

DEVAMSIZLIK GİRİŞ MODÜLÜ
Devamsızlık Sütununda;
Yarım Gün için --->1 Tam Gün için --->2 İzin için --->3 Rapor için --->4 Giriniz

Öğrenci No	Sınıfı	Adı ve Soyadı	Devamsızlık	Tarih
	10-A		1	

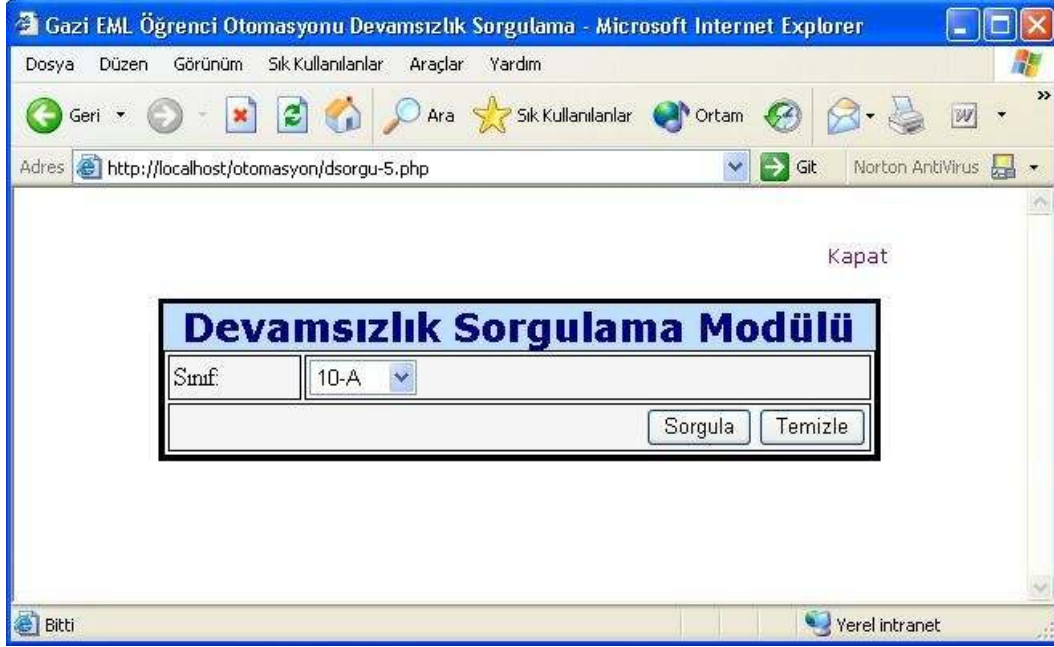
Kaydet Temizle

Bu site Elazığ Gazi Endüstri Meslek Lisesi için Hazırlanmıştır.

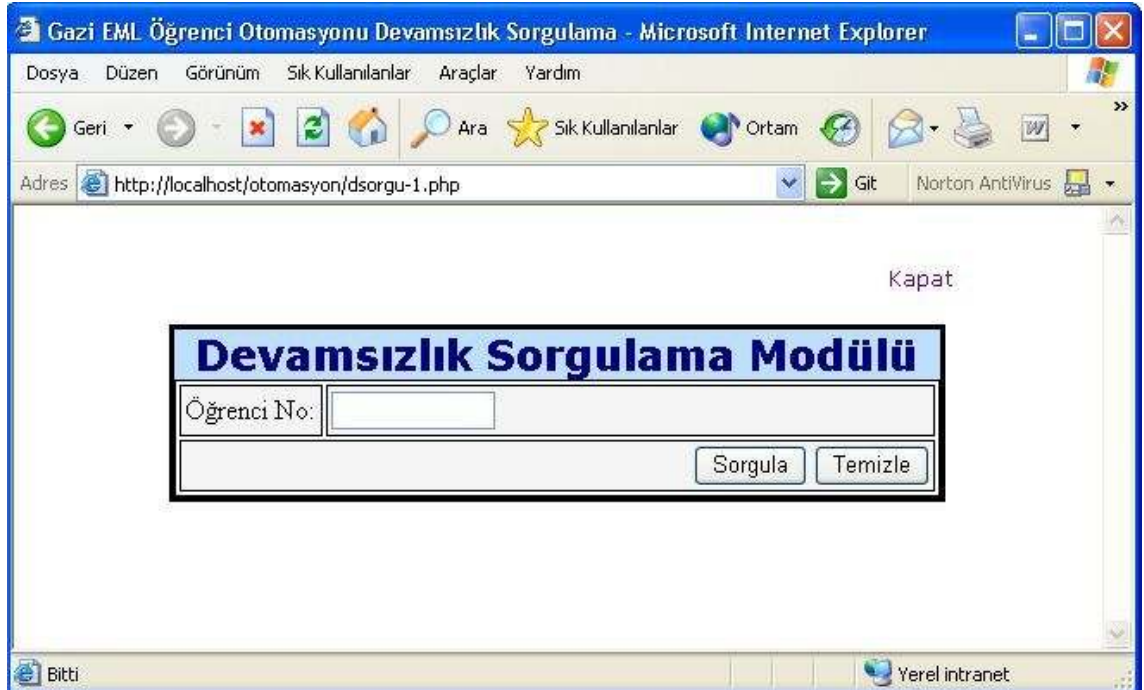
Bitti Yerel intranet

Şekil 5.11 Devamsızlık Girişi

c. Öğrenci bazında devamsızlık: Yukarıda anlatılan sınıf bazında sorgulamadan farklı olarak sadece öğrenci numarası girilerek sorgulama yapılır ve sadece girişi yapılan öğrenciye ait devamsızlık bilgisi görüntülenir.



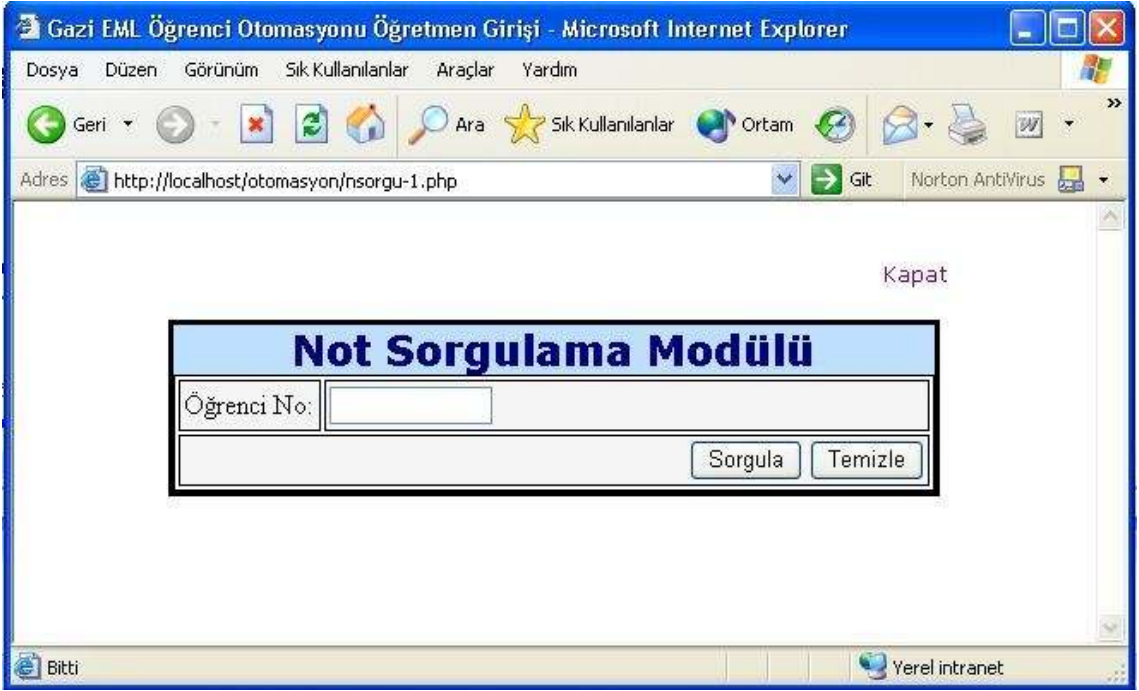
Şekil 5.12 Sınıf Bazında Devamsızlık Sorgulama



Şekil 5.13 Öğrenci Bazında Devamsızlık Sorgulama

6. **Not Sorgula:** Yılsonu ders ortalamaları modülünden farklı olarak öğrenci bazında tüm notların görülmesi amaçlanmıştır. Yani; yılsonu ders ortalamaları modülünde her bir dersin sınıf not ortalamalarına ulaşılabilirken, not sorgulama modülünde sadece numarası girilen öğrencinin aldığı tüm derslerin notlarına ulaşılabilir. Şekil 5.14'te görüldüğü gibi, öğrenci numarası girildikten sonra sorgula butonuna tıklamak yeterlidir.

7. **İstatistikler:** İstatistikler başlığı altında yapılabilecek üç işlem vardır. Bunlar Şekil 5.15'te görüleceği üzere; “Şube ve Öğrenci Sayıları”, “Yeni Kayıt Olanlar” ve “Yaş ve Cinsiyet Analizi” işlemleridir.

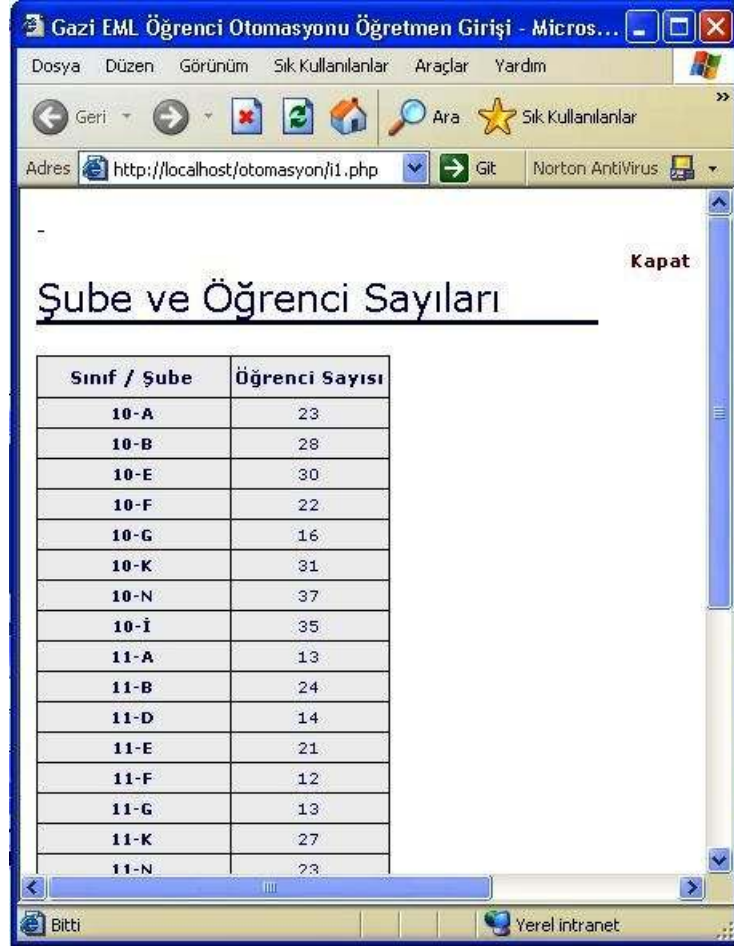


Şekil 5.14 Öğrenci Bazında Not Sorgulama



Şekil 5.15 İstatistik İşlemleri

- a. Şube ve öğrenci sayıları: “Öğrenci İşleri Otomasyonu Programının” uygulanacağı tüm okullarda her bir sınıfta bulunan öğrenci sayılarının gösterilmesi amaçlanmıştır.
- b. Yeni kayıt olanlar: “Öğrenci İşleri Otomasyonu Programının” uygulandığı okula yeni kayıt olan öğrencilerin listesini görebilmek için sisteme eklenmiştir.



Sınıf / Şube	Öğrenci Sayısı
10-A	23
10-B	28
10-E	30
10-F	22
10-G	16
10-K	31
10-N	37
10-İ	35
11-A	13
11-B	24
11-D	14
11-E	21
11-F	12
11-G	13
11-K	27
11-N	23

Şekil 5.16 Şube ve Öğrenci Sayıları



Öğrenci No	Adı ve Soyadı	Sınıfı
245	SIYAMI ÇINAR	9-A
256	ADİL SURETTİ	9-A
327	NIHAT KAYA	9-A
673	ARİ ÖZCELEK	9-A

Şekil 5.17Yeni Kayıt Olan Öğrenciler Listesi

c. Yaş analizi: Öğrenci yaş dağılımını görebilmek için sisteme dahil edilmiştir.

8. **Ders Yetkilendirme:** “Öğrenci İşleri Otomasyonu Programının” uygulanacağı tüm okullarda onlarca ders ve onlarca öğretmen ve yüzlerce öğrenci olacaktır. Hangi derse hangi öğretmenin gireceği, hangi derse ilişkin öğrenci notlarının hangi öğretmen tarafından girileceği, hangi öğrenciye hangi dersin notlarının gösterilmesi gibi konular tam bir karmaşa haline gelecektir. Bu aşamada hangi derse hangi öğretmenin gireceği, öğretmenin girdiği derse ilişkin öğrenci notlarının gireceği belirlenerek o derse ilişkin bilgilerin sadece o öğretmen tarafından düzenlenmesi amaçlanmıştır.

Öğrenci No	Adı ve Soyadı	Doğum Tarihi	Yaşı
271	SERHAT ALSU	15-10-1989	16

Şekil 5.18 Yaş Analizi

Dersin Kodu	
Dersin Adı	
Bölümü	
Kredisi	
Sınıf	
Öğretmen No	

Şekil 5.19 Ders Yetkilendirme Modülü

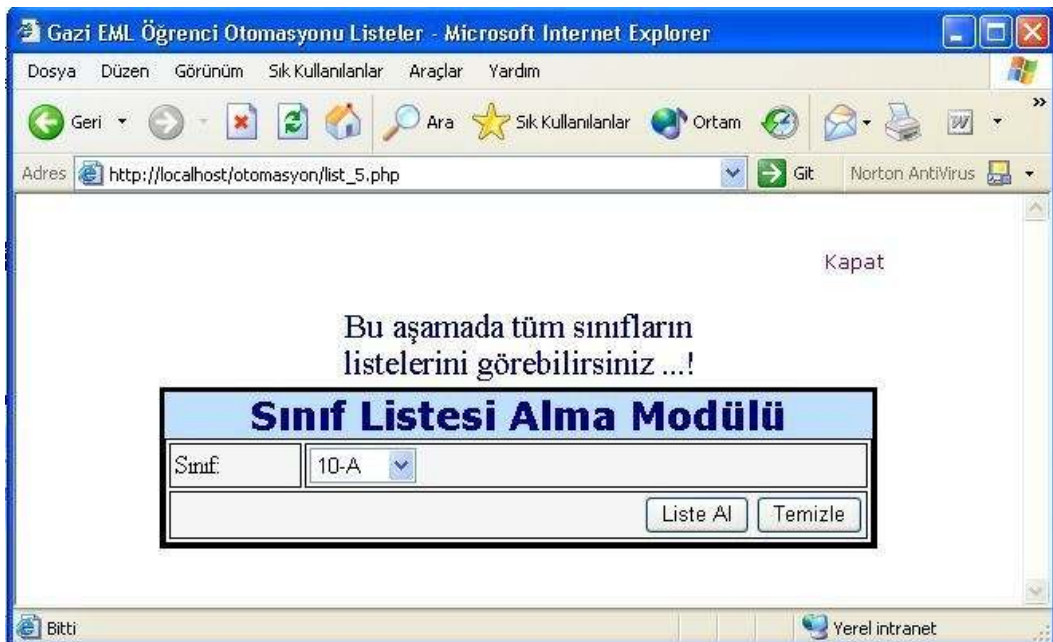
Şekil 5.19’da gösterilen “Ders Yetkilendirme” sayfasında dersin kodu, dersin adı, haftalık ders saat sayısı, okutulduğu sınıf ve derse girecek öğretmenin programa kayıtlı numarası girilerek işlem tamamlanır.

9. Listeler: Bu aşamada, “Öğrenci İşleri Otomasyonu Programının” uygulanacağı okulda mevcut tüm sınıf listelerinin alınabilmesi hedeflenmiştir. Bu noktada “Öğrenci İşleri Otomasyonu Programının” uygulanacağı okulun bir meslek lisesi olması durumunda meslek alanlarına göre liste alabilmek için Şekil 5.20’de görüldüğü bölüm listesi modülü de eklenmiştir.

a. Sınıf listesi: “Öğrenci İşleri Otomasyonu” programına yönetici haklarıyla giren kullanıcı; programa kayıtlı sınıflardan birini seçerek, Şekil 5.21’de görüldüğü gibi “Liste Al” butonunu tıklamasıyla seçilen sınıftaki öğrencilerin listesi ekranda görüntülenir. Bu işlem için, öğrenci bilgilerinin tutulduğu veritabanı dosyasını kullanıp, sınıf bilgisinin kaydedildiği alan üzerinde bir sorgulama işlemi gerçekleştirilir.



Şekil 5.20 Listeler

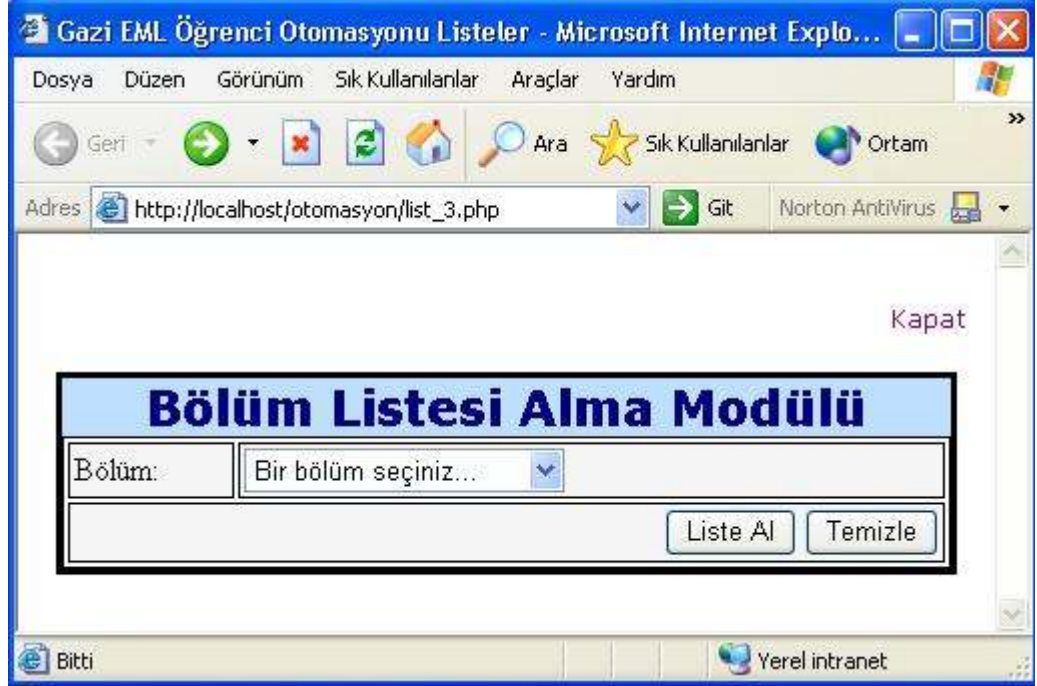


Şekil 5.21 Sınıf Listeleri

b. Bölüm listesi: Yukarıda anlatılanlardan farklı olarak liste sınıfa göre değil, bölüme göre hazırlanır. Bunun için Şekil 5.22’de görüldüğü gibiorgulama işleminde bölüm bilgisinin tutulduğu alan kullanılır.

5.1.2 Öğretmen Kullanıcılar

Sisteme öğretmen haklarıyla bağlanan kullanıcı; sistem tarafından o öğretmene ait kişisel sayfasına yönlendirilir.



Şekil 5.22 Bölüm Listeleri

Öğretmen kişisel sayfasında Şekil 5.23'te görüldüğü gibi yapılabilecek 9 farklı işlem vardır. Bunlar:

1. **Kişisel Sayfam:** Öğretmenin kişisel bilgilerinin bulunduğu sayfadır. Şekil 5.24'te görüldüğü gibi; öğretmenin resim, kullanıcı adı, ad, soyad, ünvan, e-posta, yayınlamış çalışmaları, özgeçmiş ve hobi ile zevkler gibi kişisel bilgiler bulunur.



Şekil 5.23 Öğretmen Sayfası

2. **Bilgi Güncelleme:** Bu aşamada sisteme kullanıcı adı ve parolasını giren öğretmen haklarına sahip kullanıcı; Şekil 5.25'te ki gibi ad, soyad, unvan ve e-posta bilgilerini güncelleyebilir. Bunların dışında kalan bilgiler ise yönetici tarafından değiştirilebilir.

Şekil 5.24 Öğretmen Kişisel Bilgileri

Şekil 5-25 Öğretmen Kişisel Bilgileri Bilgi Güncelleme Ekranı

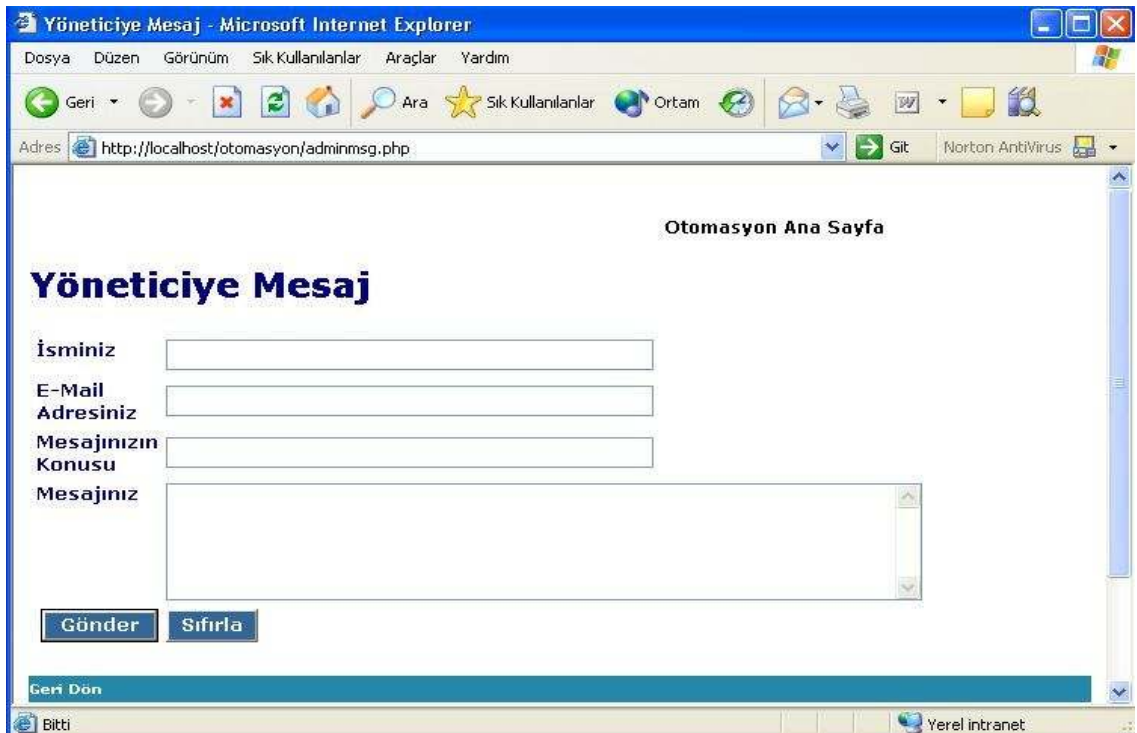
3. **Şifre Değiştirme:** Güvenlik sebeplerinden dolayı izlenecek bir yoldur.

Her üç kullanıcı grubunda da tanımlanmış olan şifre değiştirme işlemi için Şekil 5.26’da görülen eski şifre girildikten sonra yeni şifre iki kez girilerek işlem tamamlanır.

4. **Yöneticiye Mesaj:** Programın yöneticisine sistem ile ilgili görüş ve soruların iletilmesi amacıyla eklenmiştir. Kullanıcı “Yöneticiye Mesaj” sayfasında Şekil 5.27’de görülen ad, e-posta adresi, mesaj konusu ve mesajınz bilgilerini girip, sayfada ki “Gönder” butonunu tıklayarak işlemi tamamlar.



Şekil 5.26 Öğretmen Şifre Değiştirme Ekranı



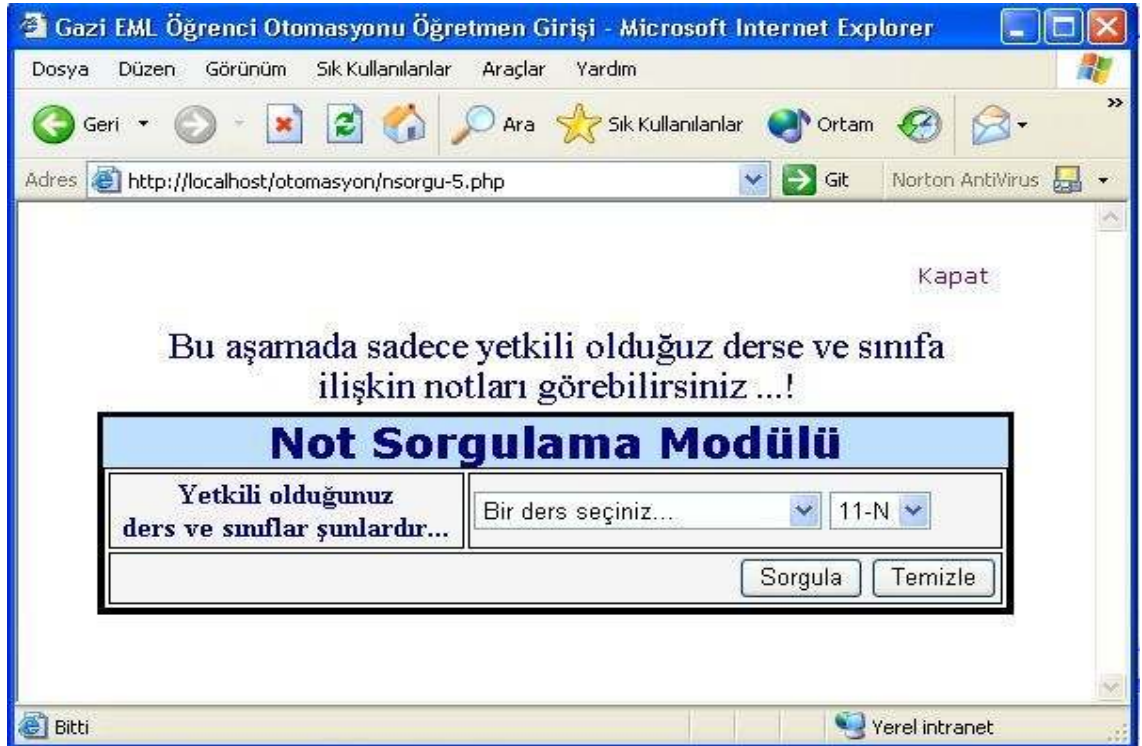
Şekil 5.27 Yöneticiye Mesaj Gönderme Ekranı

5. Yılsonu Ders Ortalamaları: Bir öğretim yılı boyunca her öğrencinin her bir dersi için farklı sayıda yazılı ve sözlü notları oluşacaktır. Her öğrenci en az bir dersten yıllık ödev notu alacaktır. Bu notlara göre yılsonu notu belirlenerek öğrencinin her bir derste ki yıllık başarı durumu belirlenir.

Şekil 5.28’de görüldüğü gibi “Yılsonu Ders Ortalamaları” modülünde her bir öğretmen sadece yetkilendirildiği derslere ilişkin yılsonu ortalamalarını görebilir.

6. Sınıf Not Bilgileri: Yukarıda anlatılan “Yılsonu Ders Ortalamaları” modülünden farklı olarak öğretmen, yetkilendirildiği derlere ait notları görebilir.

Şekil 5.29’da görüldüğü üzere öğretmen haklarına sahip kullanıcı; yönetici tarafından yetkilendirildiği derslerden birini seçerek yıl içerisinde öğrenciye verilen notları görebilir.



Sekil 5.28 Yılsonu Ders Ortalamaları

7. **Sınıf Listeleri:** Herhangi bir okulda bir öğretmen birden fazla sınıfın dersine girmektedir. Tabi ders öğretmenine, farklı işlemlerde kullanmak üzere sınıf listelerini gerekecektir. “Öğrenci İşleri Otomasyonu” programının uygulandığı okulda öğretmen haklarıyla programa giren kullanıcı sadece dersine girdiği sınıflara ait sınıf listesini Şekil 5.29’da görüldüğü gibi alabilir.

8. **Devamsızlık:** Devamsızlık modülü ile Şekil 5.31’de görüldüğü gibi öğrenci ve sınıf tabanlı devamsızlık sorgulama işlemleri yapılabilir.

Kapat

Bu aşamada sadece yetkili olduğunuz derse ve sınıfa ilişkin notlarını görebilirsiniz ...!

Not Sorgulama Modülü

Yetkili olduğunuz ders ve sınıflar şunlardır...

Bir ders seçiniz... 11-N

Not Türü...

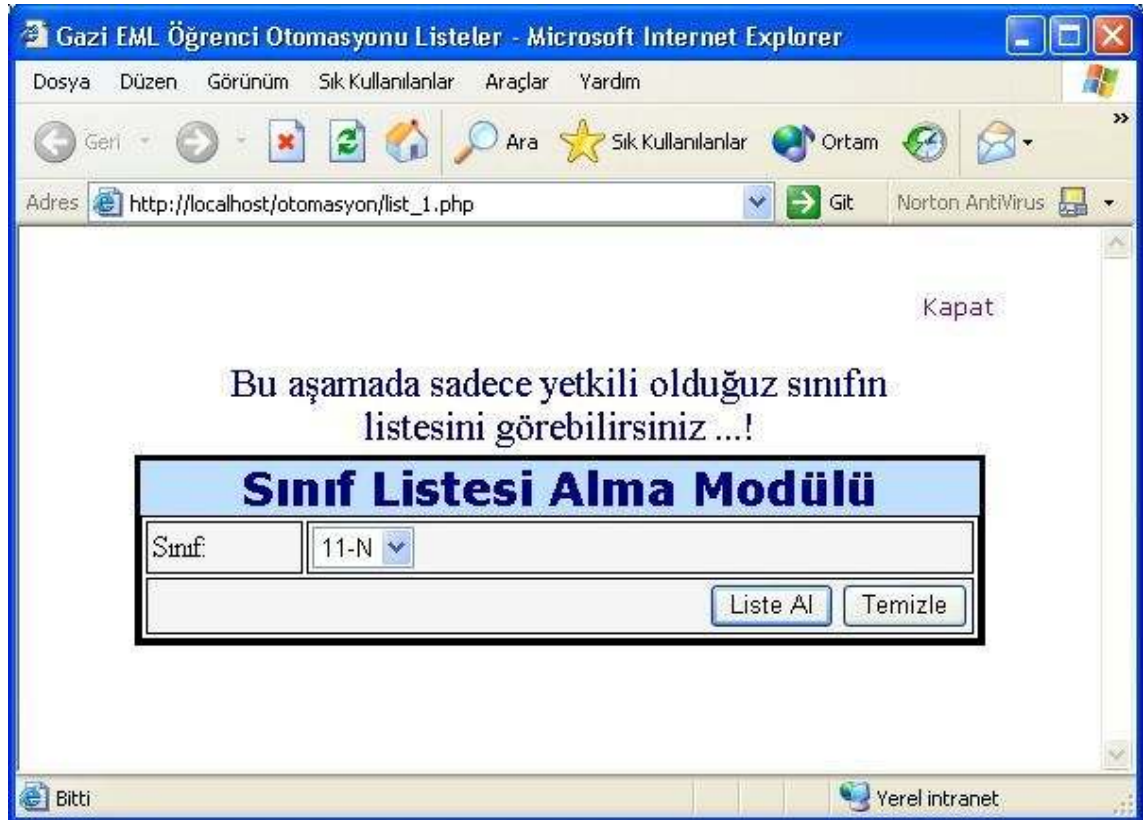
Sorgula Temizle

Bitti Yerel intranet

Şekil 5.29 Sınıf Not Bilgileri

a. Sınıf bazında devamsızlık: Sadece yönetici haklarıyla programa giriş yapanlar için, kullanıcı tarafından girişi yapılan devamsızlık bilgilerinin öğretmen tarafından incelenebilmesi amaçlanmıştır.

Şekil 5.32’de görüldüğü gibi öğretmen haklarına sahip kullanıcı, sadece dersine girdiği sınıfların listesinin bulunduğu açılır kutu içerisinde ki sınıflardan birini seçip “Sorgula” butonunu tıklayınca, seçilen sınıfa ait devamsızlık bilgileri listelenir.



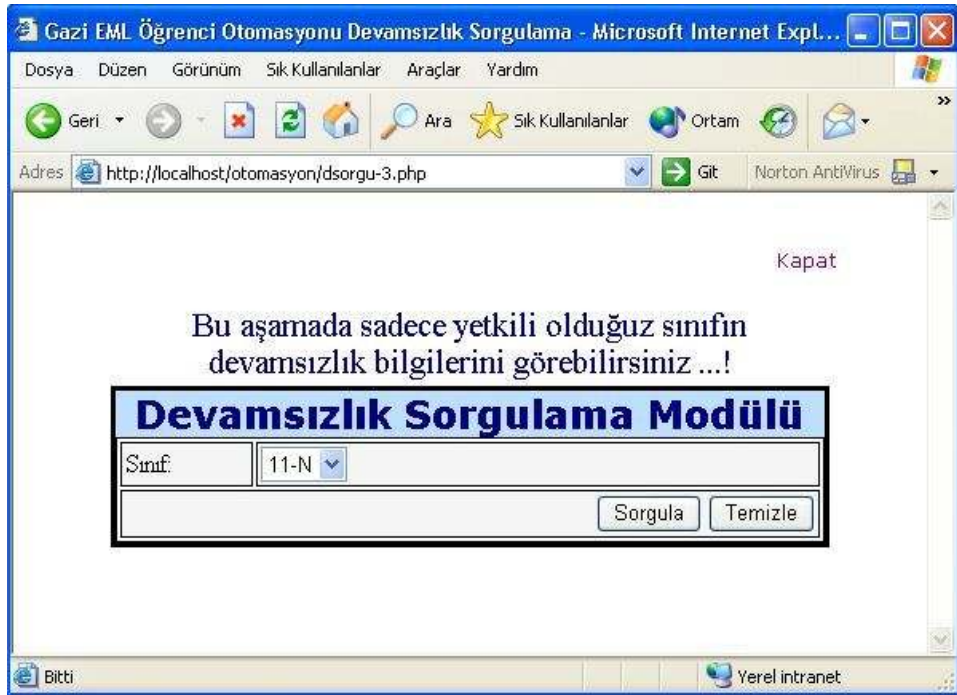
Şekil 5.30 Sınıf Listeleri



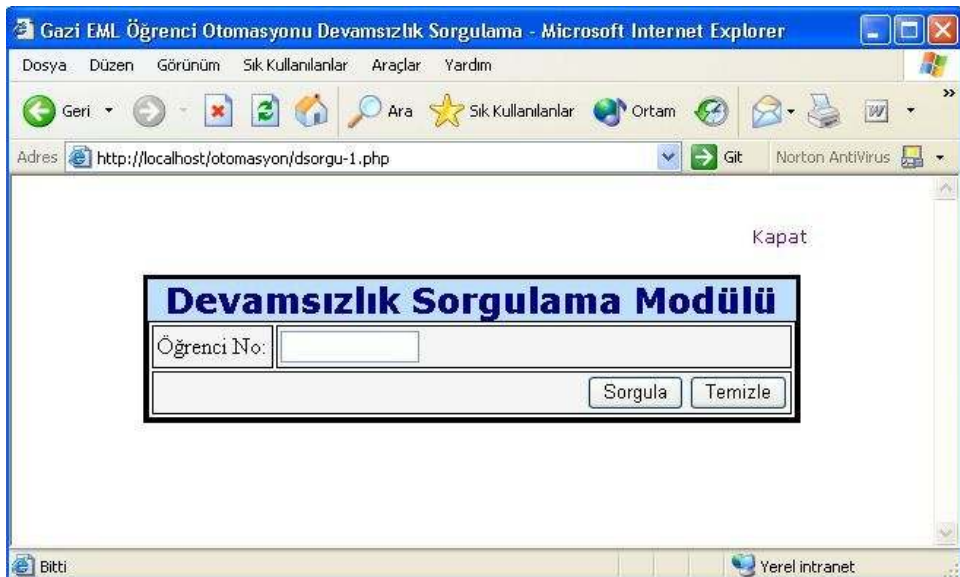
Şekil 5.31 Devamsızlık Sorgulama Ekranı-1

b. Öğrenci bazında devamsızlık: Yukarıda anlatılan “Sınıf bazında devamsızlık” modülünden farklı olarak tek bir öğrenciye ait devamsızlık bilgileri görüntülenir.

9. **Not Girişi:** Burada öğretmenlerin girdikleri derse ilişkin öğrenci notlarını girmesi amaçlanmıştır. Öğretmen, kişisel sayfasında “Not Girişi” bağlantısını tıklayınca; sadece yönetici tarafından yetkilendirildiği derslerin isimlerinin olduğu bir sayfa açılır. Şekil 5.34’te “Bir ders seçiniz” açılır kutusu içerisinde öğretmenin yetkilendirildiği dersler, hemen yanında ki açılır kutuda ise sınıflar listelenir. Öğretmen bu derslerden not girişi yapacağı dersi seçerek işlemi tamamlar.



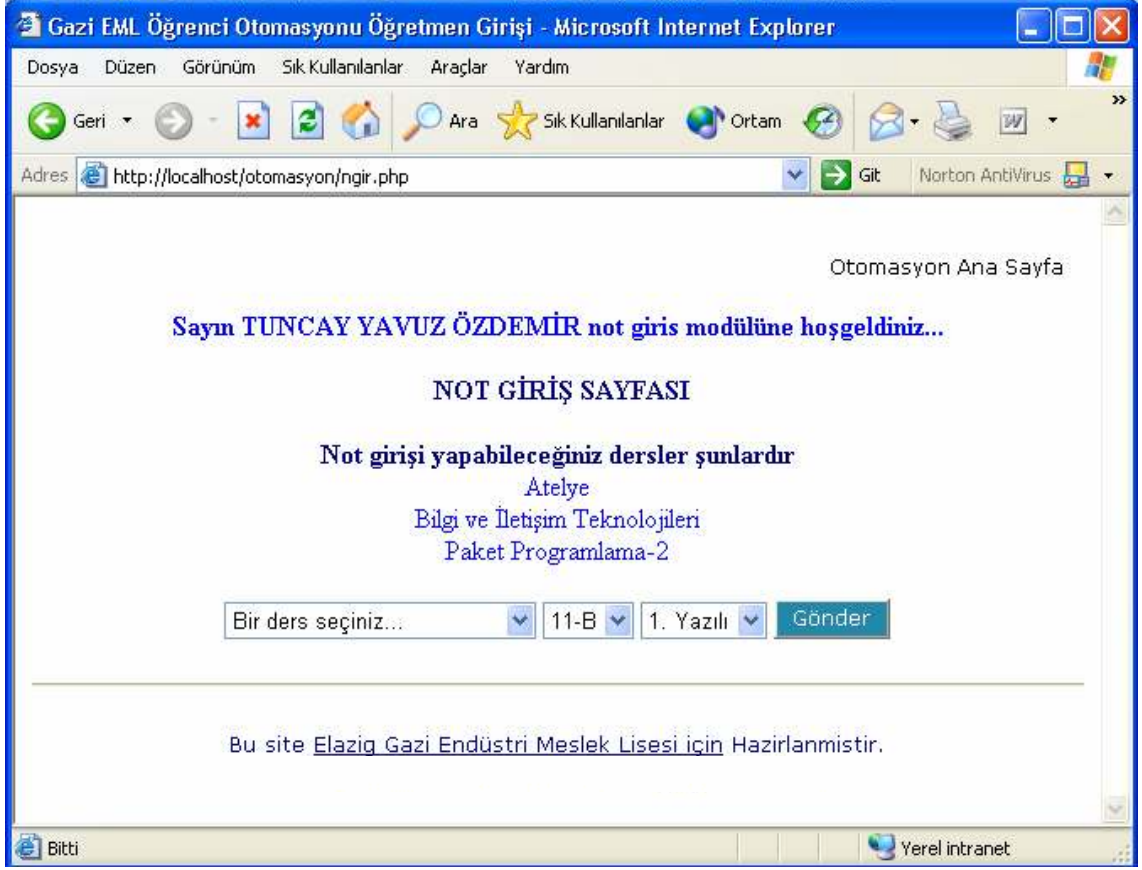
Şekil 5.32 Devamsızlık Sorgulama Ekranı-2



Şekil 5.33 Devamsızlık Sorgulama Ekranı-3

5.1.3 Öğrenci Kullanıcılar

Otomasyon programına öğrenci yetkileriyle bağlanan kullanıcı Şekil 5.35'te görüldüğü gibi; kişisel bilgilerini, not ortalamalarını, şifre değiştirme işlemlerini, kişisel sayfası ile devamsızlık bilgilerini görebileceği ve yöneticiye mesaj gönderebileceği “Öğrenci Sayfası”na yönlendirilir.



Şekil 5.34 Öğretmen Not Giriş Ekranı

Öğrenci kişisel sayfasında Şekil 5.35'te görüldüğü gibi yapılabilecek altı farklı işlem vardır. Bunlar:

1. **Kişisel Bilgilerim:** Şekil 5.36'te görüldüğü gibi “Kişisel Bilgilerim” sayfasında; öğrenci numarası, kullanıcı adı, adı, soyadı, resim, e-posta adresi, doğum yeri ve yılı, anne-baba adı, ev telefonu, cep telefonu, T.C. kimlik numarası, özgeçmiş ve hobi/özel zevkler gibi kişisel bilgilere ulaşılabilir. Sayfanın üst tarafında bulunan “Bilgilerimi Güncelle” bağlantısı tıklanarak öğrenci bilgileri güncellenebilir.



Şekil 5.35 Öğrenci Giriş Sayfası

Elazığ Gazi EML Öğrenci Otomasyonu - Microsoft Internet Explorer

Dosya Düzen Görünüm Sık Kullanılanlar Araçlar Yardım

Geri Ara Sık Kullanılanlar Ortam

Adres http://localhost/otomasyon/ogrnkisi.php

ÖĞRENCİ KİŞİSEL BİLGİLERİ

Bilgilerimi Güncelle | Kapat

Öğrenci Numarası		
Kullanıcı Adı		
Adınız		
Soyadınız		
Resminiz	Sisteme yüklenmiş resim yok	
E-Posta Adresi		
Doğum Yılı ve Yeri		
Ana Adı		
Baba Adı		
Ev / İş Telefonu		
Cep Telefonu		
T.C. Kimlik Numarası		
Adres		
Özgeçmiş		
Hobiler, Özel Zevkler		

Bu site [Elazığ Gazi Endüstri Meslek Lisesi için](#) Hazırlanmıştır.

Şekil 5.36 Öğrenci Kişisel Bilgiler Penceresi

2. **Yöneticiye Mesaj:** Programın yöneticisine sistem ile ilgili görüş ve sorunların iletilmesi amacıyla eklenmiştir.

Kullanıcı “Yöneticiye Mesaj” sayfasında Şekil 5.37’de görülen ad, e-posta adresi, mesaj konusu ve mesajınız bilgilerini girip, sayfada ki “Gönder” butonunu tıklayarak işlem tamamlanır.

3. **Şifre Değiştirme:** Daha çok güvenlik sebeplerinden dolayı programa eklenen modül Şekil 5.38’de görüldüğü gibi kullanıcı; eski şifre girildikten sonra yeni şifre iki kez girerek işlemi tamamlar.

Şekil 5.37 Yöneticiye Mesaj Gönderme Ekranı

Şekil 5.38 Öğrenci Şifre Değişirme Ekranı

4. **Not Ortalaması Alma:** İşlemin adından da anlaşılacağı üzere öğrencinin not ortalamaları bilgilerine ulaşabileceği bir modüldür.

5. **Devamsızlık Bilgilerim:** Sadece yönetici haklarıyla programa giren kullanıcı tarafından girilebilen devamsızlık bilgilerinin öğrenci tarafından görülebilmesi amaçlanmıştır.

6. **Kişisel Sayfam:** Şekil 5.35'te görüntülenen öğrenci bilgileri görüntülenir. Şekil 5.32'den farklı olarak bilgi güncellemesi yapılamaz.

7. **Karnem:** “Öğrenci İşleri Otomasyonu Programının” uygulandığı okulda, sisteme öğrenci modunda giren kullanıcının, o döneme ait tüm derslere ilişkin not ortalamaları ile toplam devamsızlık bilgileri görüntülenir.

6. SONUÇ

Bu tez çalışmasında web tabanlı öğrenci işleri otomasyonu programını hazırlayabilmek için öncelikle dil seçimi yapılmıştır. Bu amaçla günümüzde yaygın kullanım alanı bulmuş olan CGI, ASP ve PHP web programlama dilleri incelenmiştir. Bu diller arasında bir seçim yapabilmek için bir performans analizi yapılmıştır. Performans değerlendirmesi amacıyla; web sitesi üyelik sistemi seçilerek adı geçen bu üç web programlama dili ile ayrı ayrı örnek uygulamalar geliştirilmiştir. Seçilen uygulamada, en çok kullanılan bölümler olduğu için, her bir uygulamada ki giriş sayfası, üye girişi ve üye kaydı bölümleri üzerinde performans değerlendirmesi yapılmıştır. Her bir bölümün çalıştırılması durumunda, CPU'yu meşgul etmesi % cinsinden ölçülerek elde sonuca göre en iyi performansı PHP web programlama dilinin sergilediği görülmüştür.

Bu nedenle tez çalışması kapsamında geliştirilen “Öğrenci İşleri Otomasyonu Programının” hazırlanması için Web programlama dillerinden PHP tercih edilmiştir.

Dil tercihi yapıldıktan sonra öğrenci işlerinin ana unsurları belirlenerek, bu işlemlerden hangilerinin web ortamına taşınmasıyla olası zaman ve işgücü kayıplarının önüne geçilebileceği tespit edilmiştir.

Bu karşılaştırma sonucu ile PHP web programlama dili kullanılarak öğrenci işlerinin web ortamında yapılabilmesini sağlayan bir otomasyon programı geliştirilmiştir. Geliştirilen otomasyon programı ile okullarda idareciler tarafından özenle ve zamanında yapılması gereken işlemlerden önemli ve zaman alıcı olanlarını web ortamına taşıyıp, zaman ve mekândan bağımsız hale getirerek, olası zaman ve işgücü kayıplarının önüne geçilmesi amaçlanmıştır.

Geliştirilen otomasyon programı ile idareciler; sisteme giriş yapacak kullanıcılar ile ilgili gerekli tanımlamaları yaparak, tüm öğrenci ve öğretmenleri sisteme dahil eder. Bununla birlikte her bir ders için öğretmenlerin not girişi yapabilmesi amacıyla gerekli izinleri verir. Böylece; öğrencilerin ders notlarını, sadece o derse giren öğretmenin girmesi sağlanır. İdareciler tarafından yapılan “Kullanıcı Tanımlama” ve “Ders Ekleme” işlemleriyle birlikte her bir öğrencinin yıl içerisinde yaptığı devamsızlık bilgileri sisteme kaydedilir. Ayrıca sisteme kayıtlı olan öğrenci ile ilgili her türlü bilgiye ulaşım izni idarecilere verilmiştir.

Öğretmenler; öğrencilerin yazılı, sözlü ve ödev notları ile öğrenciler hakkında görüşlerini girebilmektedir. İdareciler tarafından verilen izinle sadece dersine girdikleri öğrencilere ait ders notlarını ve devamsızlık bilgilerini, sınıf bazında ve öğrenci bazında görebilmektedir.

Öğrenciler ise; kullanıcı adı ve parolalarını kullanarak web ortamında kendilerine ait not ve devamsızlık bilgilerini görebilir. Aynı şekilde öğrenci velileri de kullanıcı adı ve parola bilgileri ile öğrencilerine ait her türlü bilgiye ulaşabilmektedir.

Geliştirilen bu program Elazığ Gazi Endüstri Meslek Lisesinde kullanıma sunulmuştur.

7. KAYNAKLAR

- [1] HOLZSCHLAG, M. E.” H TML 4”, Sistem Yayıncılık, İstanbul, 2004.
- [2] Stanek R William, Microsoft SQL Server 2000 Administrator’s Packet Consultant, Microsoft Pres, 2000
- [3] UYSAL, M, PHP 4 ile Web Tasarımı, Beta Yayınevi, 2001
- [4] İnternet: Programlama.Com, Programlama.Com, http://www.programlama.com/sys/c2html/view.php3&varList=DocID_D4016 (Erişim tarihi: Mart 2005)
- [5] Mustafa Hadi Dilek, Mustafa Hadi Dilek, http://hadi.ulakbim.gov.tr/myworks/internethaftasi2001/classic/php_3.myhtml (Erişim tarihi: Nisan 2004)
- [6] ACAR, Ö. ”Perl/Cgi”, Pusula Yayıncılık, İstanbul, 2001.
- [7] İnternet: Oksinet İnternet Çözümleri, CGI Programlama http://www.oksinet.com/cgi_programlama.htm, (Erişim tarihi: Temmuz 2004)
- [8] ÖCAL,H, Ö, CGI El Kitabı, PcNet Dergisi,2000
- [9] ASP Standarts of the İnternet Information Server Resource Kit, Microsoft Pres,1988
- [10] KUZU, A, ASP ile Web Programcılığına Giriş, Beta Yayınevi, 2001
- [11] İnternet: Türkçe Asp Merkezi, ASP’ye Giriş, http://www.maxiasp.com/konu_oku.asp?knno=74&ort=63,7931034482759, (Erişim tarihi: Mayıs 2005)
- [12] Varsha, A, Tony, H, Paul R, “Performance comparison of dynamic Web platforms”, A Department of Computer Science and Engineering, Indian Institute of Technology-Bombay, Mumbai 400 076, India, 2002.
- [13] İnternet: Programlama.Com, Programlama.Com, <http://www.programlama.com/sys/chtml/view.php3?DocID=3> (Erişim tarihi: Nisan 2005)
- [14] İnternet: Türkiye PHP Grubu, Dosya Arşivi, http://www.php.org.tr/index.php?option=com_dn&d=32&Itemid=36 (Erişim tarihi: Nisan 2005)
- [15] ÇAYCI, Ö, PHP ve MySQL, Seçkin Yayınları, 2002
- [16] İnternet: Zend The PHP Company, Zend PHP Education, <http://www.zend.com/store/products/zend-optimizer.php> (Erişim tarihi: Mayıs 2005)
- [17] Ullman, L. “PHP for the World Wide Web Visual Quickstart Guide”, Peachpit Press - 2001.

- [18] Özdemir, T.Y., “Web Programlama Dilleri”, Yüksek Lisans Semineri, Fırat Üniversitesi Fen Bilimleri Enstitüsü, 2004.
- [19] Özdemir, T.Y., Türkoğlu, İ., Türel, Y.K., “Performance Analysis of Web Program Languages”, 1st INTERNATIONAL VOCATIONAL AND TECHNICAL EDUCATION TECHNOLOGIES CONGRESS, P311, 5-7 September 2005, Istanbul.

8. ÖZGEÇMİŞ

1977 yılında Elazığ'da doğdu. İlk, orta ve lise öğrenimini Elazığ'da tamamladıktan sonra 1995 yılında Fırat Üniversitesi Teknik Eğitim Fakültesi Elektronik-Bilgisayar Eğitimi Bölümü Bilgisayar Öğretmenliği Bölümünü kazandı. 2001 yılında Malatya Mehmet Topsakal İlköğretim Okulu'nda Bilgisayar Öğretmeni olarak göreve başladı. 2003 yılı Şubat ayında Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Ana Bilim Dalı yüksek lisans eğitimine başladı. 2003 yılı Mart ayında Elazığ Gazi Endüstri Meslek Lisesi Bilgisayar Bölümüne Bilgisayar Öğretmeni Olarak atandı. Halen Elazığ Gazi Endüstri Meslek Lisesi'nde Bilgisayar Bölüm Şefi olarak görev yapmakta olup evli ve bir çocuk babasıdır.