

**KOCAELİ ÜNİVERSİTESİ \* FEN BİLİMLERİ ENSTİTÜSÜ**

**PHP, MySQL VE XML TABANLI TÜRKÇE DİNAMİK  
WEB SİTESİ İÇERİK YÖNETİM SİSTEMİ: DyNA**

**YÜKSEK LİSANS TEZİ**

**İsmail KIRBAŞ**

**Anabilim Dalı: Elektronik ve Bilgisayar Eğitimi**

**Danışman: Yrd. Doç. Dr. Mehmet YILDIRIM**

**KOCAELİ, 2007**

**KOCAELİ ÜNİVERSİTESİ \* FEN BİLİMLERİ ENSTİTÜSÜ**

**PHP, MySQL VE XML TABANLI TÜRKÇE DİNAMİK  
WEB SİTESİ İÇERİK YÖNETİM SİSTEMİ: DyNA**

**YÜKSEK LİSANS TEZİ**

**İsmail KIRBAŞ**

**Tezin Enstitüye Verildiği Tarih: 28 Aralık 2007**

**Tezin Savunulduğu Tarih: 23 Ocak 2008**

**Tez Danışmanı**

**Yrd. Doç. Dr.  
Mehmet YILDIRIM**

(.....)

**Üye**

**Yrd. Doç. Dr.  
Hakan KAPTAN**

(.....)

**Üye**

**Yrd. Doç. Dr.  
Celal ÇEKEN**

(.....)

**KOCAELİ, 2007**

## **ÖNSÖZ VE TEŞEKKÜR**

Tüm dünyada internetin hızlı gelişimi ve yaygınlaşması, başlangıçta durağan sayfalar şeklinde hazırlanan web sitelerinin artık ihtiyaçları karşılayamaz hale gelmesine yol açmıştır. Bu çalışmada, çok fazla sayıda ve tipteki web sayfalarını sistematik bir biçimde üretmek ve yönetmek için geliştirdiğimiz ve DyNA olarak adlandırdığımız dinamik web sitesi içerik yönetimi sistemi tanıtılmaktadır. DyNA, web sayfası hazırlama tecrübesi olmayan kişilerin dahi kolaylıkla ve kullanıcı yetkileri doğrultusunda, herhangi bir başka programa ihtiyaç duymaksızın, site içeriği ile ilgili temel işlemleri rahatlıkla yapabileceği bir geliştirme ve yönetim ortamı sunmaktadır.

Tez çalışmam süresince değerli zamanlarını ayıran, bilgi ve deneyimlerini paylaştan, mesai kavramı gözetmeksizin kendisine ulaşabildiğim, çalışmalarımı yönlendiren ve her zaman destek olan tez danışmanım Yrd. Doç. Dr. Mehmet YILDIRIM'a ve tez aşamasında deneyimlerinden ve bilgilerinden faydalandığım değerli dostum Arş.Gör. Mevlüt KARAÇOR'a teşekkürlerimi bir borç bilirim.

Beni bugünlere getiren anne ve babama, tez çalışması sırasında yardımlarını esirgemeyen tüm dostlarıma çok teşekkür ederim. Tez çalışması süresince yoğun çalışmalarına sabreden, beni her zaman anlayışla karşılayan ve desteğini hiçbir zaman esirgemeyen değerli eşim Yasemin KIRBAŞ'a teşekkür ederim.

## İÇİNDEKİLER

ÖNSÖZ .....	i
İÇİNDEKİLER .....	ii
ŞEKİLLER DİZİNİ .....	v
TABLolar DİZİNİ .....	vii
SİMGELER DİZİNİ .....	viii
ÖZET .....	ix
İNGİLİZCE ÖZET.....	x
1. GİRİŞ .....	1
2. WEB SİTESİ İÇERİK YÖNETİM SİSTEMLERİ .....	5
2.1. İçerik Kavramı .....	5
2.2. İçerik Yönetim Sistemi .....	6
2.3. İçerik Yönetim Sistemi Türleri .....	7
2.4. Web Sitesi İçerik Yönetim Sistemi .....	8
2.5. Web Sitesi Üretimi ve Proje Yönetim Süreci .....	10
2.5.1. Web projesinin başlangıç aşaması .....	10
2.5.2. Web projesinin tanımlanması .....	11
2.5.3. Web projesinin yapısı .....	11
2.5.4. Tasarım aşaması ve ilk örnek web sitesi üretimi .....	11
2.5.5. Web sitesinin üretimi ve test aşaması .....	12
2.5.6. Web sitesinin sunumu .....	12
2.5.7. Değerlendirme ve sürdürülebilirlik .....	12
2.6. Web Sitesi Üretiminde ve Yönetiminde WİYS Kullanımı .....	13
2.6.1. İçeriğin oluşturulması .....	14
2.6.2. İçeriğin yönetilmesi .....	15
2.6.3. İçeriğin yayınlanması .....	15
2.6.4. İçeriğin sunulması .....	16
2.7. WİYS'in Bileşenleri .....	16
2.7.1. İçerik yönetim uygulaması (İYU) .....	17
2.7.2. Meta-data yönetim uygulaması (MYU) .....	19
2.7.3. İçerik gönderim uygulaması (İGU) .....	20
2.8. WİYS'lerde İçerik Yönetim Çemberi .....	21
2.8.1. Hedefler ve ihtiyaçlar .....	22
2.8.2. Ziyaretçiler .....	23
2.8.3. Yayınlama .....	23
2.8.4. İçerik tipleri .....	24
2.8.5. Yazarlar .....	24
2.8.6. Edinim kaynakları .....	24
2.8.7. İş akışı ve iş bölümü .....	24
2.8.8. Erişim yapıları.....	25
2.8.9. Meta-data: içeriğin paketlenmesi .....	25
2.9. İşleyiş Bakımından WİYS .....	26
2.9.1. Toplama .....	26
2.9.2. Yönetim .....	27
2.9.3. Yayınlama .....	27

2.10. WİYS’de İş Akışı .....	28
2.10.1. Plan .....	30
2.10.2. Geliştirme .....	31
2.10.3. Yönetim .....	31
2.10.4. Yayınlama .....	31
2.10.5. Değerlendirme .....	31
2.10.6. Saklama .....	31
2.10.7. İçerik yaşam döngüsü .....	32
2.11. WİYS’de Kullanıcılar .....	33
2.11.1. Genel kullanıcı (ziyaretçi) .....	33
2.11.2. Üyeler .....	34
2.11.3. Yazarlar ve editörler .....	34
2.11.4. Yöneticiler .....	35
3. WEB SİTELERİ ve WEB SİTELERİNDE ŞABLON KULLANIMI .....	36
3.1. Yapılarına Göre Web Sitesi Türleri .....	36
3.1.1. Durağan (Statik) web siteleri .....	36
3.1.2. Değişken (Dinamik) web siteleri .....	37
3.1.3. Statik web sitesi üreten WİYS’ler .....	37
3.1.4. Statik ve dinamik web sitesi üreten WİYS’ler .....	38
3.2. Web sitelerinde şablon kullanımı .....	38
3.2.1. Gösterim mantığı ve şablonlar .....	41
3.2.2. İşaretçi stilleri .....	43
3.2.2.1. Kuralcı HTML yapısı .....	43
3.2.2.2. Özel etiketler .....	43
3.2.2.3. HTML açıklamaları .....	44
3.2.2.4. Özelleştirilmiş yazım .....	44
3.2.2.5. Betik etiketleri .....	45
3.2.3. Verinin şablona itilmesi ve çekilmesi .....	45
3.2.3.1. İtme metodu .....	46
3.2.3.2. Çekme metodu .....	46
4. DyNA’DA DİNAMİK ŞABLON KULLANIMI .....	48
4.1. İçerik Bileşeni Kavramı .....	49
4.2. DyNA’da Kullanılan Bileşik Görünümlü, Ayrışık ve Parçalı Şablon Sistemi Yaklaşımı .....	50
4.2.1. Bileşik görünüm .....	51
4.2.2. Ayrıştırma .....	52
4.2.3. Parçalama .....	52
4.3. DyNA’da Dinamik Şablon Bileşenleri .....	53
4.4. Sayfa Tipi ve Görünüm Tipi Tanımları .....	54
4.5. Sayfa Gösterimi ve Şablon-Bileşen İş Akışı .....	57
4.6. Sayfa Parçalama .....	60
4.7. Düzenleme Şablonları Kullanarak Sayfa Düzenleme .....	62
4.8. İçerik Girişi ve Düzenleme Şablonları .....	65
5. DyNA KURULUM VE ÖZELLİKLERİ .....	67
5.1. DyNA’da Sayfa Hiyerarşisi .....	68
5.2. DyNA’da Veritabanı Yapılandırması .....	70
5.3. DyNA’da Kullanılan Şablon-Bileşen Dosya Örnekleri .....	77
5.4. DyNA’da Dizin Yapılandırması .....	78
5.5. DyNA’da Site Sabitlerinin Tanımlanması .....	80

5.6. DyNA’da Sayfa Oluřturma ve İliřkilendirme Yaklařımı .....	80
5.7. DyNA ile Kullanıcı Tarafından Talep Edilen Bir Sayfanın Dinamik Olarak Oluřturulması .....	81
5.8. Web Tabanlı Çevrimiçi HTML Editörler .....	82
6. DyNA YÖNETİM PANELİ .....	84
6.1. DyNA Yönetim Panelinde Site Haritası Bölümü .....	86
6.1.1. Sayfa ekleme .....	88
6.1.2. Sayfa silme .....	89
6.1.3. Sayfaların sıralamasını deęiřtirme .....	89
6.1.4. Sayfaları düzenleme .....	89
6.1.5. Sayfa düzenlemede hızlı menü kullanımı .....	90
6.1.6. Sayfa içerięini taşıma .....	91
6.1.7. Sayfa içerięini kopyalama .....	92
6.2. Kullanıcı İşlemleri .....	93
6.2.1. Yeni rol tanımlama işlemi .....	93
6.2.2. Kullanıcı ekleme .....	94
6.2.3. Kullanıcı silme .....	95
6.2.4. Rol düzenleme .....	95
6.3. DyNA’da Yedek Alma Sistemi .....	96
6.4. DyNA Dosya Yöneticisi .....	97
6.5. DyNA İşlem Takibi – LOG .....	98
6.6. Silinen Sayfanın Geri Alınması .....	99
6.7. Gösterim Adlarının Düzenlenmesi .....	100
6.8. Sayfa Tiplerinin Düzenlenmesi .....	101
6.9. Tip İzinlerinin Düzenlenmesi .....	102
6.10. Sunucu Tarafına Toplu Dosya ve Resim Yükleme .....	103
6.11. DyNA İçerisindeki Sunucu Tarafı Çevrimiçi Resim Araçları.....	104
6.11.1. Resim boyutlandırma .....	105
6.11.2. Resim kırpma .....	106
7. SONUÇLAR .....	108
KAYNAKLAR .....	110
ÖZGEÇMİř .....	115

## ŞEKİLLER DİZİNİ

Şekil 2.1. İçerik yönetim sistemleri .....	8
Şekil 2.2. Web sitesi içerik yönetim sistemi genel özellikler çemberi .....	9
Şekil 2.3. WİYS temel iş akış diyagramı .....	17
Şekil 2.4. İçerik yönetim uygulaması (İYU) .....	18
Şekil 2.5. Meta-data yönetim uygulaması (MYU) .....	20
Şekil 2.6. İGU ve farklı biçimlere dönüştürülmüş sonuç sayfaları .....	21
Şekil 2.7. İçerik yönetim çemberi .....	22
Şekil 2.8. İşleyiş bakımından içerik yönetimi yapısı .....	27
Şekil 2.9. Web sitesi içerik yönetim sisteminde iş akışı .....	30
Şekil 2.10. İçerik yaşam döngüsü .....	33
Şekil 3.1. Durağan (statik) web sitesi yapısı .....	36
Şekil 3.2. Değişken (dinamik) web sitesi yapısı .....	37
Şekil 3.3. Statik web sitesi üreten WİYS yapısı .....	37
Şekil 3.4. Statik ve dinamik web sitesi üreten WİYS yapısı .....	38
Şekil 4.1. Web sayfası ve sayfayı oluşturan içerik bileşenleri .....	50
Şekil 4.2. Sayfa tipleri .....	54
Şekil 4.3. Makale sayfası (Sayfa tipi 1) .....	55
Şekil 4.4. Site haritası sayfası (Sayfa tipi 2) .....	56
Şekil 4.5. İletişim sayfası (Sayfa tipi 3) .....	56
Şekil 4.6. DyNA çevrimiçi sayfa üretimi .....	59
Şekil 4.7. Beş bölgeye ayrılmış bir sayfa örneği .....	60
Şekil 4.8. DyNA yönetim paneli ekran görüntüsü .....	63
Şekil 5.1. DyNA sayfa hiyerarşisi .....	69
Şekil 5.2. DyNA temel izin yapısı .....	79
Şekil 5.3. HTML zengin editör normal görünüm .....	82
Şekil 5.4. HTML zengin editör HTML kod görünümü .....	83
Şekil 6.1. DyNA yönetim paneli yönetici giriş ekranı .....	85
Şekil 6.2. DyNA yönetim paneli genel görünümü .....	86
Şekil 6.3. DyNA yönetim paneli sol menü görüntüsü .....	87
Şekil 6.4. DyNA yönetim paneli sayfa ekleme işlemi .....	88
Şekil 6.5. DyNA yönetim paneli sayfa ekleme işlemi başarılı mesajı .....	88
Şekil 6.6. DyNA yönetim paneli sayfa düzenleme işlemleri .....	90
Şekil 6.7. DyNA yönetim paneli hızlı erişim menüsü .....	91
Şekil 6.8. DyNA yönetim paneli sayfa taşıma işlemi .....	91
Şekil 6.9. DyNA yönetim paneli sayfa taşıma işlemi .....	92
Şekil 6.10. DyNA yönetim paneli sayfa kopyalama işlemi .....	92
Şekil 6.11. DyNA kullanıcıları yönetim bölümü ekran görüntüsü .....	93
Şekil 6.12. DyNA yönetim paneli yeni rol tanımlama bölümü ekran görüntüsü .....	94
Şekil 6.13. DyNA yönetim paneli kullanıcı ekleme bölümü ekran görüntüsü .....	95
Şekil 6.14. DyNA yönetim paneli kullanıcı silme bölümü ekran görüntüsü .....	95
Şekil 6.15. DyNA yönetim paneli rol düzenleme bölümü ekran görüntüsü .....	96
Şekil 6.16. DyNA yedekleme bölümü ekran görüntüsü .....	97

Şekil 6.17. DyNA dosya yöneticisi ekran görüntüsü .....	98
Şekil 6.18. DyNA işlem takibi ekran görüntüsü .....	99
Şekil 6.19. DyNA yönetim panelinde silinen sayfanın geri alınması .....	100
Şekil 6.20. DyNA yönetim panelinde gösterim adlarının düzenlenmesi .....	101
Şekil 6.21. DyNA yönetim panelinde sayfa tiplerinin düzenlenmesi .....	102
Şekil 6.22. DyNA yönetim panelinde sayfa tip izinlerinin düzenlenmesi .....	103
Şekil 6.23. Sunucu tarafına toplu dosya ve resim yükleme .....	104
Şekil 6.24. DyNA yönetim paneli resim işleme bölümü ekran görüntüsü .....	105
Şekil 6.25. DyNA resim boyutlandırma bölümü ekran görüntüsü .....	106
Şekil 6.26. DyNA resim kırpma bölümü ekran görüntüsü .....	107



## **TABLolar DİZİNİ**

Tablo 4.1. Örnek görünüm tipleri .....	57
Tablo 4.2. Örnek veri ile doldurulmuş gösterim tablosu .....	62
Tablo 5.1. Gösterim tipi adları ve değerleri .....	71
Tablo 5.2. Örnek tipler tablosu içeriği .....	71
Tablo 5.3. Örnek tip_ izinleri tablosu içeriği .....	72
Tablo 5.4. Örnek gösterim tablosu içeriği .....	73
Tablo 5.5. Örnek roller tablosu içeriği .....	74
Tablo 5.6. Örnek kullanıcılar tablosu içeriği .....	75
Tablo 5.7. Örnek içerik tablosu içeriği .....	76
Tablo 5.8. Geçmiş tablosunun alanları .....	77

## SİMGELER

### Kısaltmalar

AJAX	:Asynchronous JavaScript and XML
API	:Application Programming Interface
CGI	:Common Gateway Interface
CMS	:Content Management System
CRM	:Customer Relation Management
CSS	:Cascading Style Sheets
DTD	:Document Type Definition
FTP	:File Transfer Protocol
GIF	:Graphics Interchange Format
HTML	:Hyper-Text Mark-up Language
HTTP	:Hyper-Text Transfer Protocol
IP	:Internet Protocol
IIS	:Internet Information Server
İYS	:İçerik Yönetim Sistemi
İYU	:İçerik Yönetim Uygulaması
JPEG	:Joint Photographic Expert Group
MGK	:Model Görünüm Kontrolörü
MIME	:Multipurpose Internet Mail Extensions
MVC	:Model View Controller
MYU	:Meta-data Yönetim Uygulaması
PDF	:Portable Document Format
PHP	:Hypertext Pre-Processor
RD	:Resource Description Framework
SGML	:Standard Generalized Markup Language
SQL	:Structured Query Language
TCP/IP	:Transmission Control Protocol / Internet Protocol
UPA	:Uygulama Programlama Arayüzü
URL	:Uniform Resource Locator
URN	:Uniform Resource Name
W3C	:World Wide Web Consortium
WAP	:Wireless Application Protocol
WCMS	:Web Content Management System
WİYS	:Web Sitesi İçerik Yönetim Sistemi
WML	:Wireless Markup Language
WYSIWYG	:What You See Is What You Get
WWW	:World Wide Web
XHTML	:eXtensible Hyper Text Markup Language
XML	:eXtensible Markup Language
XSL	:eXtensible Style Language

# PHP, MySQL VE XML TABANLI TÜRKÇE DİNAMİK WEB SİTESİ İÇERİK YÖNETİM SİSTEMİ: DyNA

**İsmail KIRBAŞ**

**Anahtar kelimeler:** Web Sitesi İçerik Yönetim Sistemi, Web Sitesi Tasarımı, Dinamik Şablon, XML, PHP, MySQL

**Özet:** Tüm dünyada İnternetin hızlı gelişimi ve yaygınlaşması, başlangıçta kullanılan ve durağan sayfalar şeklinde hazırlanan web sitelerinin artık ihtiyaçları karşılayamaz hale gelmesine yol açmaktadır. Bu sorunun çözümüne ilişkin olarak, birçok çalışma yapılmış ve halen de yapılmaktadır. Hazırlanan bu tez çalışması ile, “DyNA” ismini verdiğimiz yeni bir dinamik web sitesi içerik yönetim sistemi geliştirilmiş ve problemin çözümüne yeni bir yaklaşım sunulmaya çalışılmıştır. DyNA, web sitesi yazarları, yöneticileri ve geliştiricileri için hazırlama ve yayınlama aşamasında pek çok kolaylığı beraberinde getirmektedir. Web sayfası hazırlama tecrübesi olmayan kişilerin dahi, kullanıcı yetkileri doğrultusunda, herhangi bir başka programa ihtiyaç duymaksızın, site içeriği ile ilgili temel işlemleri rahatlıkla gerçekleştirebilecekleri bir yönetim ortamı sunmaktadır.

Web sitelerinde görsel ve yapısal bütünlüğü sağlamak, kurumsal görünüme zarar vermemek amacıyla şablonlar kullanılmaktadır. DyNA web sitesi içerik yönetim sisteminde de şablon bileşenler kullanılmakta, böylece sitede görsel bütünlük sağlanırken, site içeriğinin kolaylıkla üretilmesi ve yönetilmesi mümkün hale gelmektedir.

Hazırlanan web sitesi içerik yönetim sisteminin özellikleri arasında; farklı yayın ortamları için birden fazla dilde, çoklu ve kişiselleştirilebilir görünümlere izin vermesi, sayı sınırı olmaksızın rol ve yetkiler tanımlayarak kullanıcılar eklenmesi, web sitesi planlama ve yayına konma safhalarında belirtilmemiş dahi olsa yeni sayfa tipleri, yeni diller, hatta yeni yayın platformları için geliştirilebilmesi ve değiştirilebilmesi yer almaktadır.

# **PHP, MySQL AND XML BASED TURKISH DYNAMIC WEB SITE CONTENT MANAGEMENT SYSTEM: DyNA**

**İsmail KIRBAŞ**

**Keywords:** Website Content Management System, Dynamic Template, Web Site Design, XML, PHP, MySQL

**Abstract:** Improvement and widespread usage of the Internet doesn't meet the requirements of web sites, which had been used and prepared as static web sites in the beginning. So many studies take place in the literature to overcome this problem. In this thesis, a new dynamic web site content management system, called as DyNA, is developed and introduced to provide solutions with new approaches. DyNA offers a lot of facility in preparation and publishing of web site to authors, administrators and developers. It provides a management environment to make basic operations easily, without the requirement of another third-party software. Therefore, it can be used by even un-experienced person with respect to authority level.

Templates are used in web sites to keep visual and constitutional integrity and to protect institutional appearance from damages. DyNA web site content management system also uses templates, hence while the integrity of web site is provided easily production and management of the content is becoming possible.

The presented web site content management system has many advantageous features. It provides multi-language, customized and multiple appearances for different publication environments. It is able to append unlimited actors by defining roles and authorities. Even though it is not described in the planning and publishing phase, a new page-type, a new language and even a new publishing platform can be developed and transformed.

## 1. GİRİŞ

İçerik veriye benzerdir ve bilgi olarak da algılanabilir. Burada önemli bir ölçüt bilginin bir değer taşıyor olmasıdır. İşlenmemiş veri içerik olarak adlandırılmaz. Örneğin, bir veritabanında tutulan kayıtlar ve değerler ilk etapta bir içerik arz etmezler, sadece veri formunda tutulmaktadırlar. Bu veriler, değişik metotlarla incelenip raporlandığında insanlar tarafından anlaşılabilir, işlenebilir, değerlendirilebilir bir hale geldiğinde içerik olarak tanımlanabilir [1-3].

Bilgisayar bilimleri açısından içerik yönetimi, içeriğin yaşam devri ile ilgili eylem, işlem ve araçların tümüdür. Bu bağlamda içerik, sayısal ortama geçirilmiş ya da geçirilmesi mümkün; düz metin, veritabanı girdileri gibi yapısal olan, veya metin işleme belgeleri, resim, film, ses kaydı, faks gibi yapısal olmayan ortamlarda var olan bilgilerdir [4,5]. Gündelik kullanımda ise içerik kelimesi bilgi, belge, veri, enformasyon kelimeleri ile benzer anlamları ifade eder. İçerik yönetimini modelleyen bilgisayar yazılımlarına da içerik yönetim sistemi (İYS) adı verilir [6,7].

İnternet üzerinde devasa miktarlara ulaşmış bilginin derlenmesi, kontrol edilip etkili bir şekilde sunumu, erişimi ve değerlendirilebilmesi amacıyla web sitesi içerik yönetim sistemleri (WİYS) geliştirilmektedir. Günümüzde web düzenleme araçları küçük çaplı sitelerin oluşturulmasında iyi bir ortam sağlarlar da ölçek büyüdüğünde, yazarlar arttığında ve aynı içeriği kullanan birden fazla yayın organı söz konusu olduğunda büyük ölçekli bir çözüm ihtiyaç haline gelir ve bu da WİYS'i gerekli kılar [7].

Bir WİYS, içeriği oluşturan büyük miktarlardaki verilerin toplanması, depolanması ve sonrasında yayınlanması amacıyla kullanılan büyük veritabanları, dosya sistemleri ve diğer ilgili yazılım parçalarının bir bileşimidir [8]. Bu web yayınlama sistemleri, metinleri, ses dosyalarını, video dosyalarını veya resimleri indeksleme özellikleriyle diğer veritabanlarından ayrılırlar.

Gerçekte WİYS bir üründen çok bir kavram ve projedir. Bu kavram bir işlemler kümesini kapsar. WİYS'in temel amacı etkili ve verimli web yayınına sağlayan işlemlerin bütünleştirilmesi ve otomasyonunun artırılmasıdır. WİYS, yazarlık aşamasından yayınlama aşamasına kadar geçen yolda, bir akış şeması dahilinde gerçekleştirilen tüm işlemleri yönetir, içerik depolama ve bütünleştirme işleri için bir sistem sağlar [9].

Web siteleri genellikle düzenli içerik güncellemelerine ve görünümün standartlaştırılmasına ihtiyaç duyarlar. Standart bir stil kullanmadan farklı türlerde içeriğin gösterilmesi kullanıcıda kaybolmuşluk hissi uyandırabilir. Standardizasyon aynı zamanda web sitesinin kimliğinin de önemli bir parçasıdır.

Günümüzde İnternet ortamında yayınlanan pek çok site görsel ve yapısal bütünlüğü sağlamak amacıyla şablonlar kullanmaktadır. Şablon sistemi, standardizasyonu otomatikleştirirken sitil sayfaları ve seyir çubukları kullanarak site içerisindeki yazı tipi, seyir yapısı bütünlüğünü sağlar. Web geliştiricilerini ve tasarımcılarını sitenin bakım ve içerik ile ilgili işlemlerinden kurtarır.

İçerik ve kod ayrımı tekniği; kodun daha açık yazılması, bileşenlerin tekrar kullanılabilmesi, bakımının kolaylığı, güvenliğin kontrol edilebilmesi, değiştirilebilir görünüm özellikleri, tek bir noktada yapılacak değişikliğin tüm sitede aynı şekilde görüntülenmesi, daha küçük parçacıklarla çalışma kolaylığı, iş yükünün dağıtılması gibi avantajlar ve üstünlükleri sebebiyle programcılar ve tasarımcılar arasında tercih edilmektedir [10].

Bu tez çalışmasında, bir web sitesinin görsel bölümünü oluşturan şablonlar ele alınmış, bileşen-şablon yaklaşımı kullanılarak farklı bir WİYS olan DyNA geliştirilmiş ve geliştirilen sistem açıklanmaya çalışılmıştır.

Şablonlar genellikle içine veri tabanından çekilen verilerin yerleştirildiği, böylece web sitesinin görsel ve yapısal bütünlüğünü sağlayan statik dosyalardan meydana gelirler [11,12]. DyNA'da ise, şablonlar statik dosyalar yerine aktif bileşenlerden oluşurlar ve birbirlerini çağırıp çalıştırabilirler.

DyNA, PHP (Hypertext Pre-Processor) [13] dilinde, endüstri standardı olarak kabul edilen eXtensible Markup Language (XML) [14,15] yapısı kullanılarak yazılmıştır. DyNA, UNIX, LINUX ve Windows işletim sistemlerinin farklı sürümlerinde çalışabilecek bir yapıya sahiptir.

DyNA’da, geliştirilebilir ve esnek bir WIYS yapısı oluşturabilmek için; PHP, MySQL [16] ve APACHE [17,18] kullanımı tercih edilmiştir. PHP esnek kodlama imkanı sağlarken MySQL veritabanı; güvenilirlik, kullanım kolaylığı ve hızlı çalışması sebebiyle seçilmiştir.

HTML dilinin yetersiz kaldığı pek çok konuda yapısal eksiklikleri tamamlamak, esnek veri yapıları kullanarak kayıt ve karşılıklı veri alışverişi işlemlerini yürütmek amacıyla XML dilinin kullanımı tercih edilmiştir. HTML verinin biçimi ile ilgilenirken, XML yapı ile ilişkili bilgiyi tutmaktadır [19].

DyNA kullanıcılar ve geliştiriciler için karmaşık web sayfalarını yönetmek amacıyla dinamik şablon-bileşenler kullanımıyla kolay metotlar sunmaktadır. Bileşik görünüm, içeriğin sayfa yapısından ayrılması, sayfa görünümünün parçalar halinde ele alınması gibi pek çok özelliğe sahiptir. Aynı anda birden fazla kullanıcının veri girişine ve yetkileri doğrultusunda işlemler yapmasına olanak tanır. Dahası rol tabanlı kullanıcı sistemi sayesinde farklı yetki seviyelerdeki farklı kullanıcıları destekler. XML, DyNA’nın çekirdeğini oluşturmaktadır. Bu yaklaşım sayesinde web sitesi içerisinde veya farklı web siteleri arasında veri transferi ve sorgulaması yapılabilmektedir. Aynı zamanda çeşitli platformlar için çeşitli biçimlerde yayın yapılması sağlanmaktadır.

Bölüm 2’de; içerik, içerik yönetim sistemi, web sitesi içerik yönetim sistemi gibi temel kavramlar açıklanmaktadır. Bir web projesi geliştirme aşamaları, web projesi üretiminde web sitesi içerik yönetim sistemlerinin kullanımı, içerik yönetim çemberi, içerik yönetim sistemi bileşenleri hakkında literatür araştırmalarına dayalı bilgiler verilmektedir.

Bölüm 3’de, bir sonraki bölüme zemin hazırlamak amacı ile, web sayfalarının durağan ve dinamik türleri, web sayfalarında şablonların kullanımı, şablonlarda sunulacak olan içeriğin alınıp verilmesi konularına değinilmektedir.

Bölüm 4’te, bu tez çalışması ile gerçekleştirilen, Türkçe dinamik web sitesi içerik yönetim sistemi olan DyNA’nın bileşik görünümlü, ayrışık ve parçalı şablon sistemi yaklaşımı tanıtılmaktadır.

Bölüm 5’te, DyNA’nın kurulumu, sayfa hiyerarşisi, kullanılan veritabanı gibi bazı temel özellikleri hakkında bilgiler verilmektedir.

Bölüm 6’da, DyNA’nın yönetim paneli kullanılarak; yeni sayfalar oluşturulması, düzenlenmesi ve silinmesi, kullanıcı ve rol tanımlamaları, dosya yönetimi, resim düzenleme işlemleri gibi yönetsel işlerin nasıl gerçekleştirileceği anlatılmaktadır.

Bölüm 7’de, tez çalışmasının sonuçları değerlendirilmektedir.



## 2. WEB SİTESİ İÇERİK YÖNETİM SİSTEMLERİ

### 2.1. İçerik Kavramı

Bilgisayarlar zaman alıcı ya da karmaşık matematiksel hesaplamaları gerçekleştirmek üzere geliştirildiler ve pek çok yönden insan iş gücünün yerini aldılar. Eğer bir problem, sayılar ve mantıksal varlıklarla ilgili basit işlemler serisine indirgenebilirse sonuç bir bilgisayara buldurulabilir [20].

En alt seviyedeki bilgisayarlar verileri işlerler. Bu bilgisayarlar tarafından işlenen veri insanlar tarafından doğrudan doğruya okunamaz ve anlaşılabilir, çünkü sadece bilgisayarlar tarafından anlaşılabilir üzere üretilirler. Veri işlemek üzere üretilmiş olan bilgisayarların tanımı gereği, sadece veriden ibaret olmayan içerik üzerinde işlem yapabilmeleri oldukça zordur. Teknoloji yıllar boyunca gelişmeye devam etti ve artık günümüzde bilgisayarların içeriği insan gibi anlamlandırarak üzerinde işlemler yapması ihtiyacı doğdu [20].

İçerik veriye benzerdir ve bilgi olarak da algılanabilir. Burada önemli bir ölçüt bilginin bir değer taşıyor olmasıdır. İşlenmemiş veri içerik olarak adlandırılmaz. Örneğin bir veritabanında tutulan kayıtlar ve değerler ilk etapta bir içerik arz etmezler, sadece veri formunda tutulmaktadırlar. Bu veriler, veri madenciliği esaslarıyla incelenip raporlandığında insanlar tarafından anlaşılabilir, işlenebilir, değerlendirilebilir bir hale geldiğinde içerik olarak tanımlanabilir.

İçeriğin en büyük karakteristik özelliği insani yönüdür. İçerik insanlar tarafından sezilebilir ve anlaşılabilir olma özelliğine sahiptir. Veri insanlar için doğrudan bir anlam ifade etmez iken içerik anlamlıdır ve içerisinde bilgi taşır.

## 2.2. İçerik Yönetim Sistemi

Bilgisayar bilimleri açısından içerik yönetimi, içeriğin yaşam devri ile ilgili eylem, işlem ve araçların tümüdür. Bu bağlamda içerik, sayısal ortama geçirilmiş ya da geçirilmesi mümkün; düz metin, veritabanı girdileri gibi yapısal olan veya metin işleme belgeleri, resim, film, ses kaydı, faks gibi yapısal olmayan ortamlarda var olan bilgilerdir. Gündelik kullanımda ise içerik kelimesi bilgi, belge, veri, enformasyon kelimeleri ile benzer anlamları ifade eder. İçerik yönetimini modelleyen bilgisayar yazılımlarına da İçerik Yönetim Sistemi (İYS) adı verilir [20].

İçerik yönetimi; bilgi mimarisi, iş yönetimi, yazılım ve ağ mühendisliği, içerik oluşturulması ve yayın geliştirme gibi farklı disiplinlerin dinamik bir birleşimi olarak tanımlanmaktadır [20]. İYS'ler teknik tecrübesi bulunmayan yazarların ve editörlerin çalışabilmesini mümkün kılan teknikler ile doküman veya benzeri içeriklerin yaratılmasına ve organize edilmesine yardımcı olan yazılım sistemleridir.

İYS için çoğu zaman, bir web sitesi hazırlanmasını ve web sayfalarında bulunan bilgilerin arzu edilen şekilde görüntülenmesini sağlayan yazılımlar olarak dar anlamlı bir tanımlama yapılmaktadır. Oysaki uluslararası platformda faaliyet gösteren birçok organizasyon için modern bir varlık yönetimi tekniği olarak zorunluluk haline gelmektedir. Günümüzde İYS'ler kendi içerisinde bölünmüş bir yazılım endüstrisi haline gelmiştir. Sayısız yazılım firmasının ürünleri ile birlikte birçok açık kaynak kod projesini de içerisinde barındırmaktadır.

İYS tanımı ilk olarak web sitesi hazırlama ve yönetme işlevi için kullanılmıştır. Önceleri birçok organizasyon kendi içerisinde ve amaçları doğrultusunda yazılımlar geliştirmiştir. Web sitelerini düzenleyen bu küçük sistemlerin bir pazar oluşturabileceği fikri ilk kez 1995 yılında, hala en kalabalık içerik grubuna sahip olan CNET firmasının, kendi içerik yönetim sisteminin geliştirilmesi için Vignette firması ile anlaşması sonucu ortaya çıkmıştır. Böylelikle Vignette firması İYS kavramını bir sektör haline getirmiştir. Zamanla kendi içerisinde portal sistemleri, wiki sistemleri, web-tabanlı grup yazılımları gibi alt dallara ayrılan endüstri günümüzde halen gelişimini sürdürmektedir [21].

İYS terimi kökeninde, web sitesi yayını ve yönetim sistemleri için kullanılmaktadır. Fakat İYS pazarı gelişirken, içerik yönetim sistemleri olarak anılan ürünlerin ölçekleri son derece büyümüş ve terimin anlamı parçalanmaya başlamıştır. Orijinal web sitesi yayınlama yönetim sistemi tanımının tersine portal sistemleri, wiki sistemleri, web-tabanlı grup yazılımları sıklıkla içerik yönetim sistemleri olarak tanımlanmıştır [21].

### **2.3. İçerik Yönetim Sistemi Türleri**

İYS'ler sadece bir ürün yada teknoloji değildirler. İYS, orta ve büyük ölçekli yeni nesil web sitelerinin temelini teşkil eden, geniş çaplı işlemleri tanımlayan soysal bir terimdir [22].

İYS'ler tek bir arayüz yada uygulamaya sahip değildirler. Her bir iş amacının yerine getirilmesi için verimli olarak tasarlanmışlardır. İYS'lerin uygulamaya alınması veya işletilmesi web tabanlı yada sunucu taraflı uygulamalardan farklılık gösterebilmektedir.

İYS'ler genel özellikler olarak şu türler altında gruplandırılabilirler [5]:

- Belge yönetim sistemleri,
- Bilgi yönetim sistemleri,
- Kurum veya kuruluş ölçekli uygulama bütünleştirme sistemleri,
- E-ticaret çözümleri,
- Web içerik yönetim sistemleri,

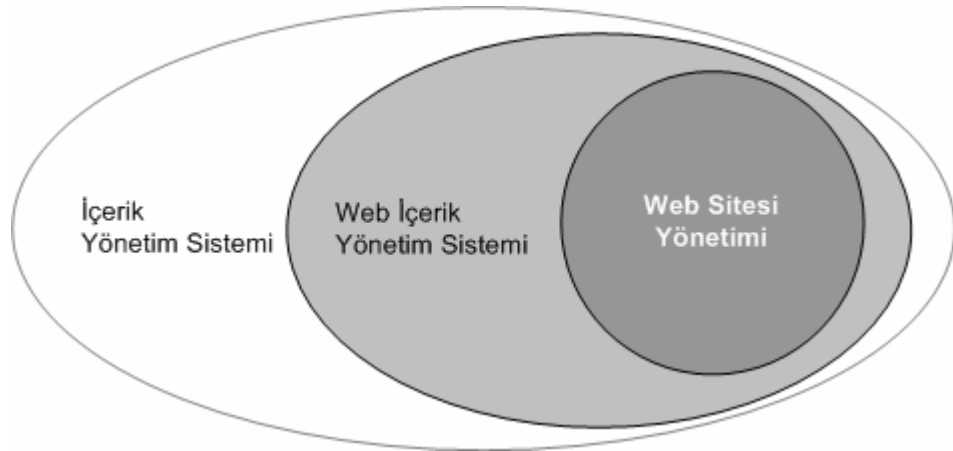
Bir web portalı diğer geleneksel türlere göre daha kısıtlı özellikler içermesine rağmen, içeriğin web üzerinden işlenip aktarıldığı bir İYS çözümdür. Bu kısıtlamalara rağmen, günümüzde İnternet'in yaygınlaşması ile, içerik dağıtımında web üzerinden yapılan uygulamalar tercih edilir hale gelmiştir.

## 2.4. Web Sitesi İçerik Yönetim Sistemi

Bir web sitesi içerik yönetim sistemi (WİYS), içeriği oluşturan büyük miktarlardaki verilerin toplanması, depolanması ve sonrasında yayınlanması amacıyla kullanılan büyük veritabanları, dosya sistemleri ve diğer ilgili yazılım parçalarının bir bileşimidir. Bu web yayınlama sistemi metinleri, ses dosyalarını, video dosyalarını veya resimleri indeksleme özellikleriyle diğer veritabanlarından ayrılır.

Günümüzde web düzenleme araçları küçük çaplı sitelerin oluşturulmasında iyi bir ortam sağlasalar da ölçek büyüdüğünde, yazarlar arttığında ve aynı içeriği kullanan birden fazla yayın organı söz konusu olduğunda büyük ölçekli bir çözüm ihtiyaç haline gelir ve bu da WİYS'i gerekli kılar [20].

Bir WİYS, web içeriğini ve web sitelerini yayınlamak için gereken ek araçlara sahip olan içerik yönetim sistemidir (Şekil 2.1). Temel olarak, dinamik ve büyük web belgeleri topluluğunu kontrol etmek ve yönetmek amacıyla kullanılan bilgisayar sistemidir. Bir WİYS genellikle yönettiği web sitesinin sunucusu üzerinde çalışır. Pek çok sistem bir web tarayıcı uygulaması üzerinde erişilebilecek şekilde bazıları da dosya transfer protokolü (FTP) desteği verecek şekilde tasarlanmıştır. İçerik ve içerikle ilgili bütün diğer bilgi genellikle sunucu tabanlı ilişkisel veri tabanında tutulur. WİYS genellikle web sitesinin önceki ve üzerinde çalışmakta olan sürümlerini de kayıtlarında tutar.



Şekil 2.1: İçerik yönetim sistemleri [20]

WİYS üzerinden kontrol edilerek yayınlanmış olan sayfalar ziyaretçilerin web tarayıcıları tarafından görüntülenebilirler. İçerik oluşturucular belgelerini sisteme girerler. Yetkili editörler gerekli kontrollerden ve son düzenlemelerden sonra ya onay verirler yada iptal ederler. Görünüm editörleri sitenin görünümünü düzenler. Şef editör sitedeki tüm yayınlardan sorumlu kişidir ve iş akışının tüm adımlarında, yayınlara ilgili tüm teknik detaylarla ilgilenerek bir yada daha fazla web sitesinde yayınlanacak şekilde kontrol eder.

WİYS kullanan bir web sitesi, sadece bünyesinde hazırladığı içeriği ziyaretçilerine sunan bir yapı değildir. Aynı zamanda diğer sayısal ortamlardan ve web sitelerinden içerik alabilir, elde ettiği içeriği işleyerek başka bir web sitesine veya sayısal ortama aktarabilir. Şekil 2.2’de bir WİYS’in genel özellikleri gösterilmektedir. Tipik bir WİYS’in çekirdeğini sürümlendirme, iş akışı ve bütünleştirme çalışmaları meydana getirirken bunların etrafında veri kaynakları, kullanıcı yönetimi, kullanıcı arayüzü, uygulamalar ve yayınlama bölümleri yer almaktadır. Her WİYS’in Şekil 2.2’de verilmiş olan katman yapısını ve özellikleri taşıma şartı olmamakla birlikte gelişmiş bir WİYS’de bunların tamamına rastlanmaktadır.



Şekil 2.2: Web sitesi içerik yönetim sistemi genel özellikler çemberi [2]

## **2.5. Web Sitesi Üretimi ve Proje Yönetim Süreci**

Bir web sitesi üretilir iken; içerik, kapsam tasarımı, site haritasının meydana getirilmesi, teknik özelliklerin seviyesinin belirlenmesi, işlevsellik, görsel tasarım, sitenin albenisi ve kullanım kolaylığı gibi unsurlar üzerinde itina ile durulması ve buna uygun yapılandırmalara gidilmesi gerekmektedir. Bu durum, web site inşasında farklı uzmanlık alanları ve farklı perspektiflerin ortak bir zemin üzerinde buluşturulduğu proje yönetimi odaklı çalışmayla mümkün olabilmektedir. Web sitesi üretim yönetimi, proje yönetimi çerçevesinde süreç odaklı düşünülen ve geliştirilen bir faaliyetler bütünüdür.

Web sitesi üretim yönetiminin, proje planlama teknikleri çerçevesinde yedi temel aşamaya dayandırılması mümkündür [23]. Bu aşamalar:

- 1) Projenin başlangıç aşaması
- 2) Projenin tanımlanması
- 3) Projenin yapısı
- 4) Tasarım aşaması ve ilk örnek web sitesi üretimi
- 5) Web sitesinin üretim ve test aşaması
- 6) Web sitesinin sunumu
- 7) Değerlendirme ve sürdürülebilirliktir.

### **2.5.1. Web projesinin başlangıç aşaması**

Web projesinin başlangıç aşamasının ana hatları; endüstri analizi (sektör rekabet analizi, endüstri ve pazar araştırması, vb.) ve hedef kullanıcıların analizinden (kullanıcı profillerinin çıkarılması, ön testler ve araştırmalar, tüketici pazar araştırması, vb.) oluşmaktadır. Proje keşif süreci olarak da adlandırılan bu aşama, zaman, emek, bütçe ve web sitesine ait temel iş stratejisinin planlanması ile sitenin nihai kullanıcılar için taşıyacağı değerin anlaşılmasına yardımcı olmaktadır. Proje takımının uzmanlaşması açısından da önem taşıyan bu süreç, zaman ve bütçeye bağlı olarak diğer aktivitelerle paralel gerçekleştirilmektedir.

### **2.5.2. Web projesinin tanımlanması**

Bir web projesinin tanımlanmasındaki birincil adımlar; genel amaç ve hedeflerin tanımlanması (öncelikli hedeflerin ve amaçların belirlenmesi, müşteri anketlerinin dağıtılması), proje planının yaratılması (proje takımının oluşturulması, programın belirlenmesi, bütçe planı ve ayarlaması, teslimi planlanan nihai site yapısına karar verilmesi, işlevsellik test planı, kalite güvenlik, ihtiyaçların belirlenmesi (yaratıcı ve teknik özet hazırlanması), site içi düzenleme (proje sitesinin başlatılması, iç geliştirme sahasının yaratılması, sunucu enformasyonunun doğrulanması, üretim yönetimi) takibinin başlatılması olarak belirlenebilir. Faaliyet alanı ve programı belirlenmiş bir site yaratılmasında, kullanıcı odaklılık ve site içi yönetimde idari aşamalar ile ihtiyaçların belirlenmesi ön planda yer almalıdır.

### **2.5.3. Web projesinin yapısı**

Web projesinin yapısını teşkil eden ana elemanlar; web sitesi kapsamı (içeriğe ait ana hatların belirlenmesi, içerik teslim planının yapılması), web sitesi görünümü (site haritasının ve isimlendirme politikasının oluşturulması, fonksiyonel alanların adreslendirilmesi), ekran görünümü (ekran şemalarının yaratılması, yönlendirmelerin, içerik ve işlevsellik çerçevesinin belirlenmesi) ve kullanıcı görünümüdür (kullanıcı senaryolarının yaratılması, kullanıcı yollarının belirlenmesi, kullanıcılarla birlikte test etme vb.). Bütün bu yapılandırma unsurları; proje müşterisi, proje ekibi ve site ilişkilerini kolaylaştıracaktır. İçerik ve enformasyonun yapısına, site, ekran ve kullanıcı perspektifleri üzerinden yaklaşılmalı, kullanıcıların talep ve beklentilerine yönelik içerik (ürün, hizmet satışı) ve/veya bilgiye kolay erişim sağlanmalıdır.

### **2.5.4. Tasarım aşaması ve ilk örnek web sitesi üretimi**

Tamamlanmış bir ön çalışma üzerinde kavramsal ve üretimsel bir bakış açısı ile yaratıcılığın ön plana çıkarıldığı tasarım aşaması, görsel tasarım yaklaşımı (kavramsal beyin fırtınası, önceliklerin belirlenmesi, grafik şablonlarının yaratılması, tasarım rehberinin hazırlanması, vb.) ve HTML ilk örneği yaratılması (ilk örnek

geliştirme, test etme ve doğrulama, işlevsellik testi, vb.) aşamalarından meydana gelmektedir.

#### **2.5.5. Web sitesinin üretimi ve test aşaması**

Web sitesi gerçek anlamda hayata geçirilmeden önce üretim öncesi planlamalar (faaliyet alanı bütçe ve teknik ihtiyaçların yeniden gözden geçirilmesi, sürüm kontrolü için yöntemlerin oluşturulması, dosya yapılarının düzenlenmesi, site bakım planı vb.), yapısal düzenlemeler (grafiklerin düzenlenmesi, HTML şablonlarının yaratılması, yazılı metinlerin adreslenmesi, sayfa düzenlemeleri, içerik ve üretimin yapılandırılması vb.) ve test yöntemleri (kalite güvenlik, performans test planı, güvenlik yönetimi ile illegal saldırı, virüslerin takibi ve önlenmesi) dikkatle hazırlanmalı, belirlenen proje hedefleri, beklentiler, varsayımlar ve teslimatlar bir kez daha gözden geçirilmelidir.

#### **2.5.6. Web sitesinin sunumu**

Web sitesinin hayata geçirilmesi ile ilgili olan bu aşamada göz önünde bulundurulması gerekenler; site isminin duyurulması, bu amaçla kaynak ve bağlantıların sağlanması, gerek İnternet üzerinde gerekse İnternet dışı olarak kullanıcıların web sitesi hakkında bilgi edinmelerinin sağlanmasıdır. Bir web sitesinin sunumunda yer alan aşamalar; sitenin İnternet üzerinden ziyaretçilere sunumu, nihai kontrollerin tamamlanması, kalite güvenliği, site bakım yönetimi, dokümantasyon takibi, site tutundurma faaliyetleridir (tutundurma planının hazırlanması, arama motorlarına katılımın sağlanması, vb.)

#### **2.5.7. Değerlendirme ve sürdürülebilirlik**

Web sitesi İnternet üzerinde yaşama geçirildikten sonra, site başarısının ölçütü olarak, hem niteleyici hem de niceleyici verilerin elde edilmesi yerinde olacaktır. Web sitesi etkinliği ve gelişimi bir dizi perspektife ve geri beslemenin çeşitli şekillerine dayanmaktadır. Web sitesinin başarısının değerlendirilmesi ve sürdürülebilirliği aşamasında; kullanıcı geri beslemesinin sağlanması ve



kullanıcıların izlenmesi, ölçülebilir sonuçların elde edilmesi ve proje sonrasında gözden geçirilmesini kapsayan sürekli gelişim planı oluşturulmalıdır. Web sitesinin değerlendirilmesinde site ziyaretçilerinin görüşleri önemli bir referans noktasıdır.

## **2.6. Web Sitesi Üretiminde ve Yönetiminde WİYS Kullanımı**

WİYS'leri sadece birer yazılım olarak değerlendirmek yerine birer proje olarak ele almak daha doğru bir yaklaşım olacaktır. Bu nedenle bir WİYS hazırlamadan veya kullanıma geçmeden önce tüm ihtiyaçların açık bir şekilde belirlenerek listelenmesi gerekir. Pek çok organizasyon WİYS'in sadece İnternet veya iç ağ ortamındaki web sayfalarının bakımı için kullanıldığını düşünür. WİYS bu temel ihtiyacın yanı sıra diğer pek çok iş ihtiyacını da karşılamak durumundadır. Sistemin iş ile ilgili ihtiyaçları ve hedefleri kesin olarak belirlenmeden değerlendirmeler yapılması ve başarının ölçülmesi oldukça güçtür. Genellemeler şeklinde ele alınabilecek belli başlı iş amaçları veya ihtiyaçları aşağıda listelenmektedir [23].

- Sitenin esnekliğinin artırılması,
- Bilgi doğruluğunun ve güncelliğinin sağlanması,
- Pazarlama ve tanıtımın desteklenmesi,
- Bilgi tekrarının önlenmesi,
- Kullanıcı tecrübesinin artırılması,
- Satışların desteklenmesi,
- Bilgi güncellemelerinin kolaylaştırılması ve hızlandırılması,
- Web sitesinin büyümesinin desteklenmesi,
- Yayın işleminin geliştirilmesi,
- Bilgi paylaşımının artırılması,
- Aranılan bilgiye ulaşımın kolaylaştırılması,
- Çalışan verimliliğinin artırılması,
- Müşteri destek maliyetlerinin azaltılması,
- Yayın maliyetlerinin azaltılması,
- Web sitesi bakım maliyetlerinin azaltılması,
- Web sitesi ziyaretçi sayısının artırılması,

Yukarıda belirtilen iş amaçlarını gerçekleştirip gerçekleştirmediğini takip edebilmek veya seviyesini ölçebilmek amacıyla aşağıdaki parametreler kullanılabilir [23].

- Web sitesi kullanım istatistikleri,
- Baskı ve dağıtım maliyetleri,
- Çağrı merkezi giderleri,
- Yeni bir materyal basımı veya mevcut materyal üzerinde değişiklik yapma zamanı,
- Müşteri memnuniyeti,
- Yeni sayfaların oluşturulma zamanı,
- Eğitim maliyeti.

Her organizasyonun ihtiyaçları kendi yapısına göre değişmekle birlikte aşağıda temel bazı ihtiyaçlar listelenmiştir. Listelenen ihtiyaçlar hangi WİYS'nin seçilmesi gerektiği sorusuna cevap teşkil etmeye yardımcı olacaktır. Temel ihtiyaçlar yapısal olarak içeriğin oluşturulmasından son kullanıcıya ulaştırılmasına kadar olan dört kategori başlığı altında incelenebilir. Bunlar:

- 1) İçeriğin oluşturulması,
- 2) İçeriğin yönetilmesi,
- 3) İçeriğin yayınlaması,
- 4) İçeriğin sunulması,

#### **2.6.1. İçeriğin oluşturulması**

Bu fonksiyon site yazarları tarafından gerekli bir özelliktir. Verimli bir yazarlık işlemi olmaksızın içerik üretiminde başarı sağlamak oldukça zor olacaktır. WİYS içerisinde WİYS'nin tüm imkanlarını kullanmaya olanak sağlayan bütünleşmiş bir yazarlık ortamı (içerik oluşturma veya içerik girişi bölümü) yer almalıdır. Farklı biçimler ve ortamlarda yayın yapılabilmesi için içerik ve sunumun birbirinden ayrıştırılmış olması gerekmektedir. Yazarlar birden fazla olabilmeli, kullanım arayüzü çoklu kullanıcı ve çoklu yazar desteğine sahip olmalıdır. Bir kez oluşturulmuş içeriğin tekrar kullanımına izin veriliyor olmalıdır. Aynı içerik hem

İnternet hem de iç ağ ortamlarında kullanılabilmelidir. İçerik hakkında detaylı bilgilerin tutulabilmesi ve gelişmiş arama seçenekleri için meta-data yapısı kullanılabilmelidir. Güçlü bir sayfalar arası bağlantı yapısının kurulabilmesi gereklidir. Yeni oluşturulan sayfalara otomatik linkler eklenebilme, kırık link kontrolü yapılabilmelidir. Teknik bilgiye sahip olmayan kullanıcılar için de içerik oluşturulabilmesi sağlanmalıdır. Kullanımı kolay ve verimliliği yüksek olmalıdır.

### **2.6.2. İçeriğin yönetilmesi**

Çözümünün odak noktasını WİYS'lerin içeriği çeşitli araçlar kullanarak yönetmesi oluşturmaktadır. Sade fakat güçlü bir arayüz ile sürüm kontrolü ve arşivleme işlemleri gerçekleştirilebilmelidir. İş akışının planlanmış olması ve sekteye uğramadan gerçekleştirilebilmesi gerekmektedir. Yazarların ihtiyaçlarının belirlenmiş ve mevcut araçlar ile karşılanabiliyor olması gerekmektedir. Giriş ve veri güvenliğinin tam olarak sağlanıyor olması gerekmektedir. Diğer harici sistemlerle bütünleşmesinin sağlanabiliyor olması gerekmektedir. Kullanıcı ve yönetici tabanlı gelişmiş raporlama işlemlerinin kolaylıkla gerçekleştirilebiliyor olması gerekmektedir.

### **2.6.3. İçeriğin yayınlanması**

Yayınlama motoru veriyi depolandığı yerlerden çıkararak kullanıcıya iletilecek sonuç sayfalarının üretilmesini sağlar. Bu işlem dinamik olarak olabileceği gibi ardışık üretim şeklinde de gerçekleşebilir. Son görünüm stil şablonları kullanılarak kontrol edilebilir olmalıdır. Bu özellik esneklik, bütünlük ve genişletilebilme imkanlarını kazandırır. Tüm sayfalar sayfa şablonları tarafından şekillendirilmelidir. İdeal olanı bu şablonların da teknik olmayan kişilerce düzenlenebiliyor olmasıdır. Sürekli geliştirilebilir şekilde bir arayüz ve altyapı sistemine sahip olmalıdır. Yayın için farklı biçim ve ortam (HTML, PDF, WAP, RDF, RSS, XML vb.) imkanlarına sahip olmalıdır. Kullanıcı profilleri doğrultusunda farklı bilgiler gösterilecek şekilde kişiselleştirilebilir olmalıdır. En çok ziyaret edilen sayfalar, günlük kullanım, arama motorlarının kullanımı gibi kullanım istatistiklerini üretebilmelidir.

#### **2.6.4. İeriğın sunulması**

Üretilen sayfalar kullanıcılar için belli bazı standartları sağlamalıdır. Web sitesi kolay kullanımlı ve verimli olmalıdır. Bunun sağlanması için prototip tasarımlar yapılarak gerçek kullanıcılar üzerinde denemeler yapılabilir. Kullanışlılık rehberleri değerlendirilmeli, bu doğrultuda bir tasarım yapılmalıdır. WİYS, W3C Web Ulaşılabilirlik İnisiyatifi tarafından belirlenmiş standartları yerine getiriyor olmalıdır. Sayfalar belli başlı tarayıcıların farklı sürümlerinde (Internet Explorer, Netscape, Opera vb.) hatasız olarak görüntülenebiliyor olmalıdır. Üretilen web sayfalarının geçerli HTML kodlarına sahip olması gerekir. Bu özellik tarayıcılar ve platformlar arasında maksimum uyumu sağlar. Etkili bir site içi gezinme (navigasyon) yapısının kurulup kullanılıyor olması gereklidir. İndeksleme ve arama işlemlerinin daha başarılı olabilmesi için meta-data kullanılabiliyor olmalıdır.

WİYS'nin toplam maliyeti sadece para cinsinden değil yapılacak işin gerektirdiği kaynak ve yeteneklerin kullanımı ile ölçülebilir.

#### **2.7. WİYS'in Bileşenleri**

Tipik bir WİYS üç temel bileşenden oluşur (Şekil 2.3). Bunlar bileşenler; içerik yönetim uygulaması (İYU), meta-data yönetim uygulaması (MYU) ve içerik gönderim uygulamasıdır (İGU) [24]. Bazı WİYS türleri daha fazla elemana sahip olabilirler, fakat genelde bu üç şekilde gruplanabilirler. İYU, WİYS'in içerik bileşenlerini yönetir. Diğer taraftan, MYU içerik bileşenleriyle ilgili bilgiyi yönetir. Son olarak da İGU, içerik bileşenlerinin site kullanıcısı tarafından görüntülenme yollarıyla ilgilenir.



Şekil 2.3: WİYS temel iş akış diyagramı

### 2.7.1. İçerik yönetim uygulaması (İYU)

Bir içerik yönetim uygulaması, içerik bileşenlerinin başlangıçlarından silinmelerine kadar tüm yaşam döngüsünü yönetir. Bir İYU, içerik bileşenlerinin depodan oluşturulması, bakımı ve silinmesi işlerini yapar [24].

Depo bir veritabanı olabileceği gibi, bir grup dosya da olabilir yada her ikisinin bir kombinasyonu yapılabilir. Yönetim işlemi doğal olarak sıralıdır ve iş akışı kullanılarak gerçekleştirilir. İYU genelde WİYS'in yönetim bölümü olarak düşünülür.

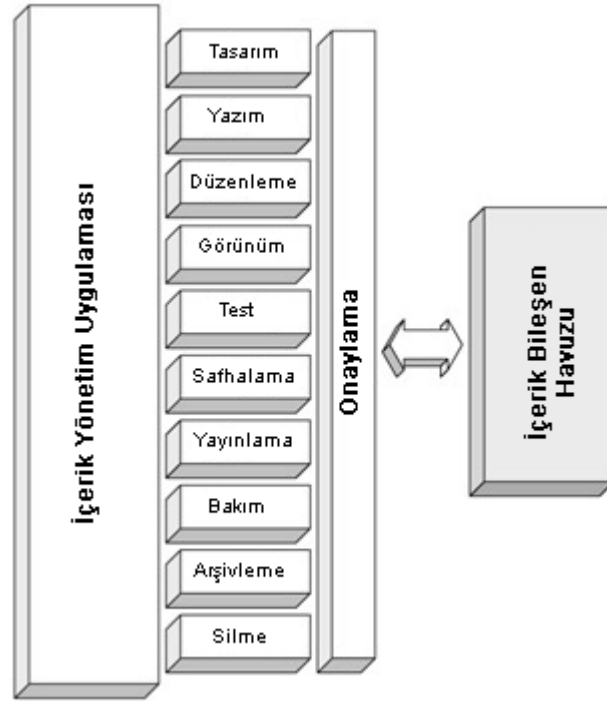
İYU içerik yazarının HTML bilmesine yada altta yatan web mimarisini bilmesine gerek bırakmaksızın içerik bileşenleri oluşturmaya izin verir. Bu bir web sitesinin günlük bakım işleri için, sabit bir web Webmaster'a olan ihtiyacı ortadan kaldırır.

Neredeyse bütün İYU'lar çoklu kullanıcıyı desteklerler. Her bir kullanıcı içerik bileşeninin yaşam döngüsü üzerinde bir yada daha fazla role sahiptir.

Pek çok İYU rol tabanlı güvenlik sistemine sahiptir, yani kullanıcılar rollerle tanımlanmış yetkileri kullanabilirler. Küçük ölçekli bir web sitesi için az sayıda kişi ve rol söz konusu iken, her rol birbirinden farklı görevleri yada fonksiyonları tanımlar. Daha büyük kapasiteli web siteleri için daha fazla rol ve daha kısıtlı yetkiler söz konusudur.

Kullanıcı rolleri İYU kurulurken ayarlanır. Rol tanımları yapılırken görev listesi ve fonksiyonlar ortaya çıkarılır. Özel görev ve fonksiyonlar belirlenerek bunlar için ayrı onay mekanizmaları oluşturulur. Bazı gelişmiş İYU'larda yeni rollerin eklenmesine ve daha önce yapılmış tanımlar üzerinde değişiklikler yapılmasına izin verilir. Böylece daha dinamik ve web sitesinin değişen ihtiyaçlarına daha uygun bir yapı elde edilmiş olur.

İYU'nun amacı içerik bileşenlerinin yaşam döngülerini mümkün olduğunca çabuk ve etkili bir şekilde yerine getirmektir. Yaşam döngüsünün sonunda, içerik bileşenleri daha olgun ve kararlı bir halde olmalıdırlar. Aşağıdaki şekil İYU'nun yerine getirmesi gereken bazı yaygın, üst seviye yaşam döngüsü evrelerini göstermektedir.



Şekil 2.4: İçerik yönetim uygulaması (İYU) [24]

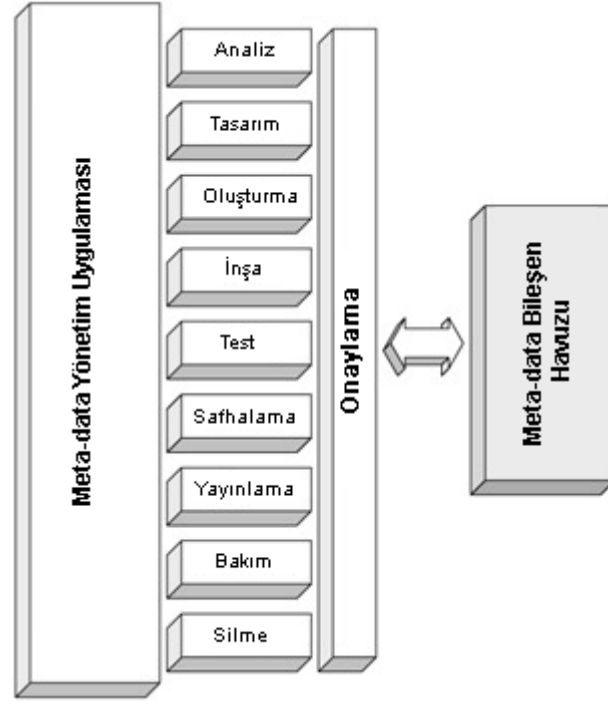
### 2.7.2. Meta-data yönetim uygulaması (MYU)

İdeal bir WIYS’de, içerik bileşenleri ve bunların yayınlanması tamamen ayrıştırılmış olmalıdır. İçerik ve içeriğin ulaştırılması işinin bir birinden ayrılmasının temel sebebi tamamen farklı iş akış düzenlerine sahip olmaları ve bunları hazırlamakla görevli kişi ve grupların birbirinden farklı olmasıdır. Editörler ve yazarlar içerik üretimi ve yönetimi işleriyle ilgilenirlerken içerik yönetim uygulamasını kullanırlar bu esnada içeriğin nasıl ve nerelerde görüntüleneceği ile ilgili herhangi bir düzenleme yapmaları gerekmez.

Meta-data, bir nesne yada olguyu herhangi bir şekilde tanımlayan verilerdir. Bilgi hakkında bilgi, web kaynakları hakkında açıklayıcı bilgi, web yada diğer kaynaklar hakkında bilgisayarlarca anlaşılabilir bilgi, kaynak keşfi ve seçim işlemini destekleyen bibliyografik veri gibi farklı tanımlara sahiptir. “Üst-veri” veya “bilgi bilgisi” şeklinde de kullanılmaktadır [26].

WIYS’lerde meta-datanın kullanım amacı; sade, kullanıcı dostu, bütünlük içerisinde bir arayüze sahip bir web sitesi oluşturabilmektir. Web sitesi kullanıcısının bir metin, bir PDF dosyası, bir görsel, video, ses dosyası yada web sitesince sağlanan herhangi bir biçimdeki içerik bileşeni ile ilgili işlem yapıyor olması fark etmez. Meta-data üretilirken şablonlar, betikler ve harici programlar kullanılabilir.

MYU, meta-datanın bütün yaşam döngüsünü yöneten bir uygulamadır. Meta-data içerik bileşenleri hakkındaki bilgidir ve özel olarak içerik bileşenlerinin web sitesi içerisinde nasıl yerleşeceğini belirler. MYU'nun amacı meta-datanın bütün hayat döngüsü boyunca işletilmesidir.



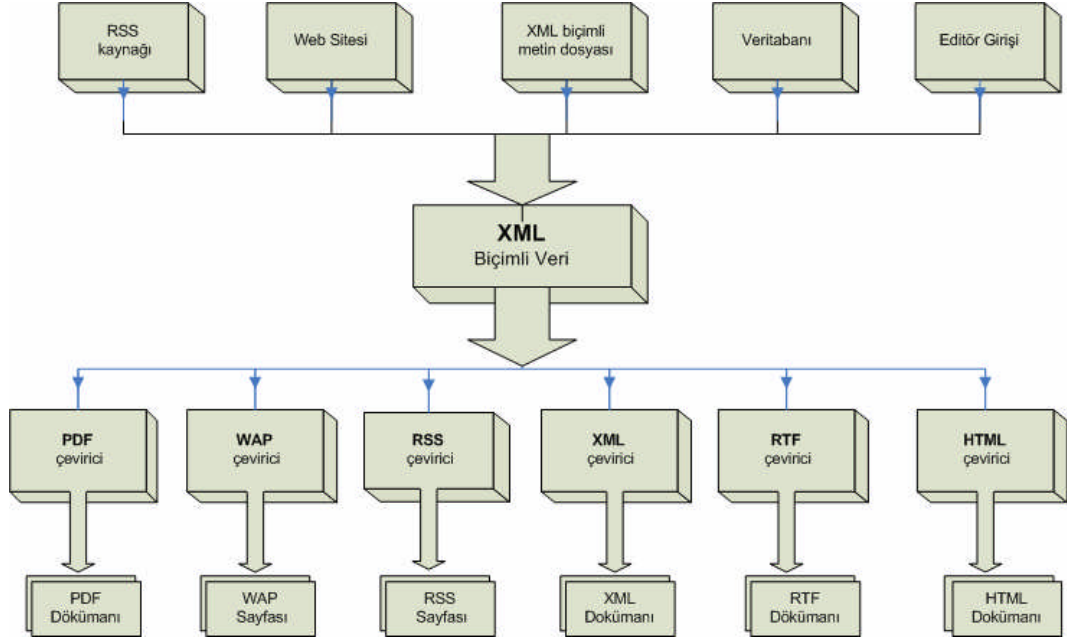
Şekil 2.5: Meta-data yönetim uygulaması (MYU) [24]

### 2.7.3. İçerik gönderim uygulaması (İGU)

İçerik gönderim uygulamasının görevi, içerik bileşenlerini WİYS veri havuzundan çıkarmak ve meta-datayı kullanarak web sitesi kullanıcılarına sunmaktır [24]. WİYS kullanıcıları, İGU'nin kurulması ve ayarlanması dışında yapacak bir göreve sahip değildir.

İyi bir İGU sadece meta-data kullanılarak idare edilebilir. Neyin ne şekilde sunulacağı neredeyse tamamen meta-data tarafından belirlenmektedir. İGU içerisinde gösterime ilişkin herhangi bir kod bulunmadığından sayfa düzeni, renk, boşluklar, yazı tipleri vb. web sayfası özellikleri dinamik olarak meta-data tarafından değiştirilebilir. İGU'lar genellikle sadece web sitesi sayfa seyir sistemi ve sitenin açılış sayfasının nasıl yükleneceğine dair konularını bilme ihtiyacı duyarlar.





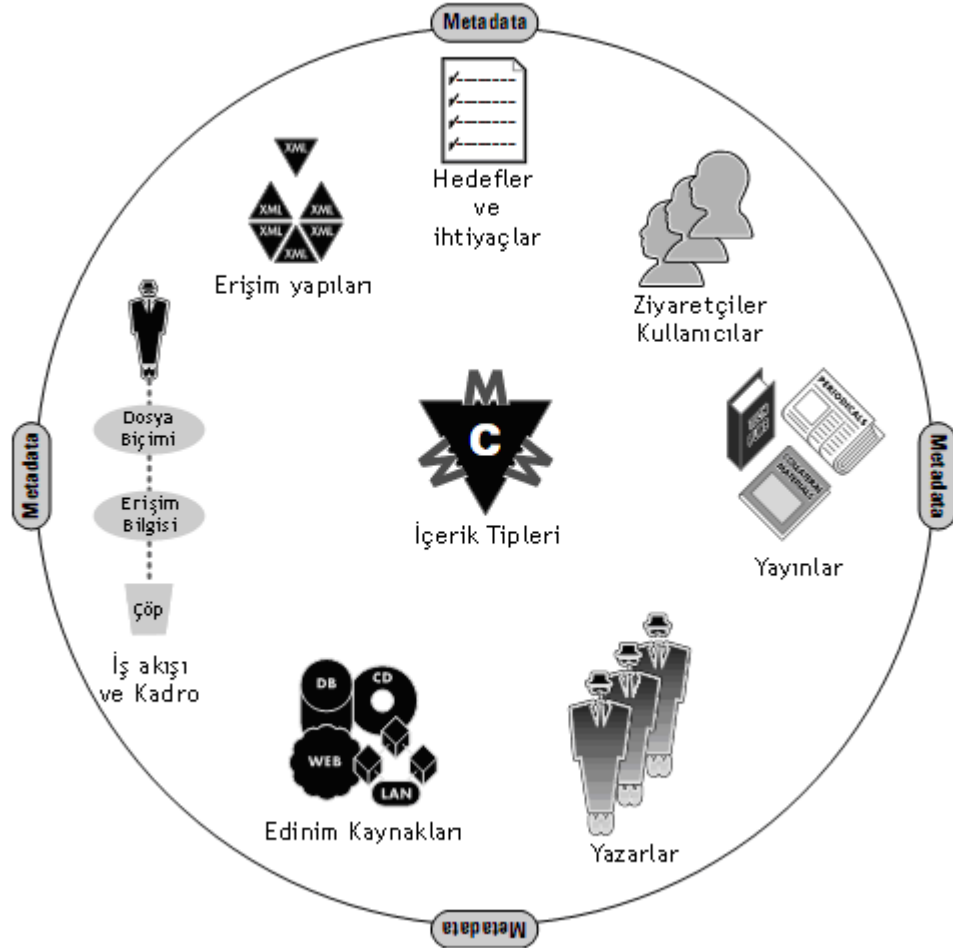
Şekil 2.6: İGU ve farklı biçimlere dönüştürülmüş sonuç sayfaları

İGU'lar aynı zamanda, web sitesi kullanıcılarının depolanan içerik üzerinde derinlemesine ve kolayca arama işlemi yapabilmelerine olanak sağlamalıdır. Sonuç sayfalarının farklı platformlar için farklı biçimlere dönüştürülmesi işleminden de İGU sorumludur. Şekil 2.6'da farklı biçimlere ilişkin örnekler gösterilmektedir.

## 2.8. WIYS'lerde İçerik Yönetim Çemberi

WIYS'ler marka ve sürümden bağımsız olarak sekiz temel bölüm halinde gruplandırılarak incelenebilir [20]. Bu bölümler WIYS'i teşkil eden temel öğelerdir. Bir WIYS oluşturmak için bahsi geçen tüm öğelerin sistemin içerisinde tam olarak yer aldığından emin olunması gerekir. Her bir öğenin kendine has sistem ihtiyaçları olmakla birlikte doğrudan ya da dolaylı olarak birbirleriyle de bağlantılı olmaları gerekmektedir. Öğeler bir yandan birbirlerinin sınırlarını belirlerken diğer yandan sistem genelinde nelerin mümkün olabileceğine karar verirler. WIYS, kendisini meydana getiren öğelerin kuvvetleri arasında bir dengenin sağlandığı meta-data etkileşimidir. Eğer öğeler dengede ise içerik üretimi ve kullanımı için kararlı ve verimli bir sistem oluşturulabilir.

Sistemin çerçevesini meta-data oluşturmaktadır (Şekil 2.7). Meta-data, sistemi bir arada tutar ve şekli belirler. Meta-data ve içerik tipleri birbirlerine oldukça yakın ve ilintilidirler.



Şekil 2.7: İçerik yönetim çemberi [20]

### 2.8.1. Hedefler ve ihtiyaçlar

Hedefler veya amaçlar, WIYS iş yönünden ele alındığında temel olarak kullanılma sebeplerini teşkil ederler. Amaçlar WIYS sistemini meydana getirecek ve kullanacak tüm öğeler tarafından ortaklaşa belirlenmelidir. Amaç bir organizasyonun elde etmeye çalıştığı hedeftir. İhtiyaç amaçlara ulaşabilmek için WIYS sisteminin karşılamak ve yerine getirmek zorunda olduğu şeylerdir. WIYS henüz daha tasarım aşamasında iken, tüm amaç ve ihtiyaçların net olarak belirlenmesi ve yazılı bir şekilde ortaya konması gerekmektedir. Başarılı ve işler bir WIYS oluşturabilmek için projede yer alan tüm birimlere WIYS de yer almasını istedikleri her şey sorulmalı ve

görüşleri alınmalıdır. İhtiyaçlar ve amaçlar, mevcut varlıklarla (ziyaretçiler, yazarlar, yayıncılar vb.) tam olarak dengelenmeli ve aralarında ödünleşmeye gidilmelidir. Ziyaretçiden, yazardan içerik tipinden ve yayından kaynaklanan her ihtiyaç sistemin amaçlarından en az birine karşılık geliyor olmalıdır. Varlıklardan herhangi biri amaçlardan herhangi biri ile çelişiyor olmamalıdır. Örneğin amaçlardan biri maliyetin düşük tutulması iken sitenin bir WAP sürümünün hazırlanması gibi birinci derecede önem arz etmeyecek bir konunun işe başlamadan önce düşünülüp kararlaştırılmış olması gerekmektedir [25].

### **2.8.2. Ziyaretçiler**

Ziyaretçiler varlığı, WİYS'in sağladığı içeriğin ulaştırıldığı katılımcılara ait bilgileri kapsamaktadır. Site hazırlanırken ziyaretçilerin de modellenmiş olması gerekir. Bu doğrultuda ziyaretçi profilleri oluşturulmalı ve mevcut davranış şekilleri, demografik özellikleri, siteyi takip dereceleri, siteye bağlanmak için kullandıkları teknolojiler, beklentileri, siteye erişim metotları, ilgi seviyeleri, site ile ilgili düşünceleri, sitenin sahibi olan organizasyonun yaptıklarını doğru tanıyıp tanımadıkları gibi konuların açıklığa kavuşturulması gerekir. Sitede sunulan içeriğin doğru kişiye, doğru zaman da ve doğru kanaldan aktarılması sağlanmalıdır.

### **2.8.3. Yayınlama**

Yayınlama varlığı, ne oluşturmak istendiği, nasıl kişiselleştirilebileceği, şablonlar yardımıyla WİYS'de nasıl oluşturulabileceği gibi yayınlar hakkındaki bilgileri içerir. Her ziyaretçi grubuna ait isim bilgileri, yayınlara ilgilenen ziyaretçileri tanımlayan veriler, yayınların hangi ortamlar aracılığıyla hangi biçimlerde gerçekleştirileceğine dair bilgiler bu bölümde yer almaktadır. Sitenin kişiselleştirilmesiyle ilgili çalışmalar bu bölüm altında gerçekleştirilir. Yayınlama ve güncelleme ile ilgili zaman çizelgeleri burada belirlenir.

#### **2.8.4. İçerik tipleri**

İçerik tipleri varlığı, toplanacak ve yönetilecek tüm içerik hakkındaki bilgiyi içermektedir. İçerikle ilgili bilgiler ve bunlara ait meta-data verileri bu bölümde tespit edilir.

#### **2.8.5. Yazarlar**

Yazar varlığı, aynı türden içeriği üreten, aynı araçları kullanan, WIYS grubu ile aynı şekilde ilişkisi olan herkesten oluşan grup olarak adlandırılabilen bir yada daha fazla kişiyi gösterir. Yazarın adı, tipi, bilgisayar kullanma bilgi ve becerisi, yetkileri, uzmanlık konuları bu varlık altında incelenir. İş akışı ile doğrudan ilgilidirler. İçeriğin üretimi, denetimi, düzenlenmesi, safhalaşması, değerlendirilmesi gibi işlemler genellikle yazarlar tarafından yerine getirilir. İçeriğin temel kaynağı yazarlardır.

#### **2.8.6. Edinim kaynakları**

Edinim kaynakları, bir şekilde önceden var olan içeriğin sisteme dahil edilmesini tanımlar. Kaynağın sahibine dair bilgi, kaynağa yapılan bağlantı şekli, kaynaktan alınan içerik gibi bilgiler bu varlık içerisinde tespit edilmektedir.

#### **2.8.7. İş akışı ve iş bölümü**

İş akışı; WIYS'in herhangi bir bileşeniyle ilgili olarak, yetkili bir kişi veya otomatikleştirilmiş bir işlem tarafından yerine getirilen görevlerin, yapılış sırasının belirlenmesidir. İş akışı bazı olaylar tarafından tetiklenir. İş akışı varlığı belli bir iş akışı için aşamaları, görevlileri, nesneleri ve tetikleyicileri gösteren bir veriler koleksiyonudur. Adımlar iş akışında yer alan sıralı aşamalardır. Aktör bir görevin yerine getirilmesinden sorumlu kişi yada otomatikleştirilmiş işlemdir. Tetikleyici adımın başlangıç aşamasında sinyal üreten/uyaran bir olaydır. İş bölümü, görevleri kesin ve açık olarak tanımlanmış kişiler arasında yapılacak işi tam olarak yerine getirebilecek bilgi ve yeteneği olan kişiler arasındaki görev dağılımıdır. Diğer

varlıkların işlem ihtiyaçları ile iş gücünün bir araya getirilmesi bu bölümde gerçekleştirilir [27].

#### **2.8.8. Erişim yapıları**

Bir erişim yapısı; içerik tipleri yada belli bir içerik tipine erişebilmek için yayınlanan sayfadan bir başka sayfaya yada bunların dışında bir yere yapılan bağlantıyla ilişkilidir.

Erişim yapıları; hiyerarşiler (ana hatlar), indeksler (anahtar kelimeler), karşılıklı bağlantılar ve sıralamalar (ileri-geri) şeklinde dört tipte olabilir. Bir erişim yapısı varlığı, yapı ve nasıl oluşturulmak istendiği ile ilgili bir veri kümesidir. Sitenin seyir yapısı, bilgiye erişim kolaylığı, WIYS veritabanında etkili ve kolay arama yapabilme gibi konular bu bölüm kapsamında incelenir.

#### **2.8.9. Meta-data: içeriğin paketlenmesi**

İnsanların anlayıp yorumlayabildiği içerik üzerinde işlemler gerçekleştirmek üzere çok sayıda metot geliştirilmiştir. Bu metotlar veri üzerinde işlem yapmayı mümkün hale getirmek için insanın bakış açısıyla içerik ve bilgiyi paketlemeye çalışırlar [26]. Bilgisayar işlemekte olduğu içeriğin barındırdığı ince anlamları kavrayamayabilir, fakat insanın istediği sonuçları üretebilecek işlemleri yerine getirebilir. Elde edilecek sonuçlardaki soyut anlamalar ancak bir insan tarafından yorumlanabilir.

Verinin bilgi ile tanımlanması ve bunun içeriğe yerleştirilmesi günlük olarak yerine getirdiğimiz ve kullandığımız yöntemlere benzetilebilir. Örneğin, kütüphanede kitap aramak işlemi ele alındığında, bir başka bilgi için gerekli bilgi ihtiyacı ortaya çıkar. Bir kütüphane, kitap adı ve yazar adı gibi kategoriler altında arama yapmamıza izin verir. Odalar dolusu kitap veri yığını olarak görülebilecekken, aynı oda içinde yer alan arama motoru gerçek içerik olarak görülebilir.

İçeriğin açıklanması meta-data olarak adlandırılır [26]. Meta-data bir nesne yada olguyu herhangi bir şekilde tanımlayan veri veya verilerdir. Bilgi hakkında bilgi, web

kaynakları hakkında açıklayıcı bilgi, web yada diğer kaynaklar hakkında bilgisayarlarca anlaşılabilir bilgi, kaynak keşfi ve seçim işlemini destekleyen bibliyografik veri gibi farklı tanımlara sahiptir. “Üst veri” veya “bilgi bilgisi” şeklinde de kullanılmaktadır. Veri hakkındaki veridir. Örneğin bir müzik dosyasında sesi oluşturan bilgiler veri olarak nitelendirilirse, bu ses dosyasının boyutları, süresi, kayıt kalitesi, şarkıcı adı, şarkı türü gibi bilgiler meta-datayı teşkil eder. Meta-data öncelikle, işlenmesi hatta anlaşılması imkansızlaşan çok büyük çapta verilerin web üzerinde barındırılmaya başlamasıyla ortaya çıkmıştır.

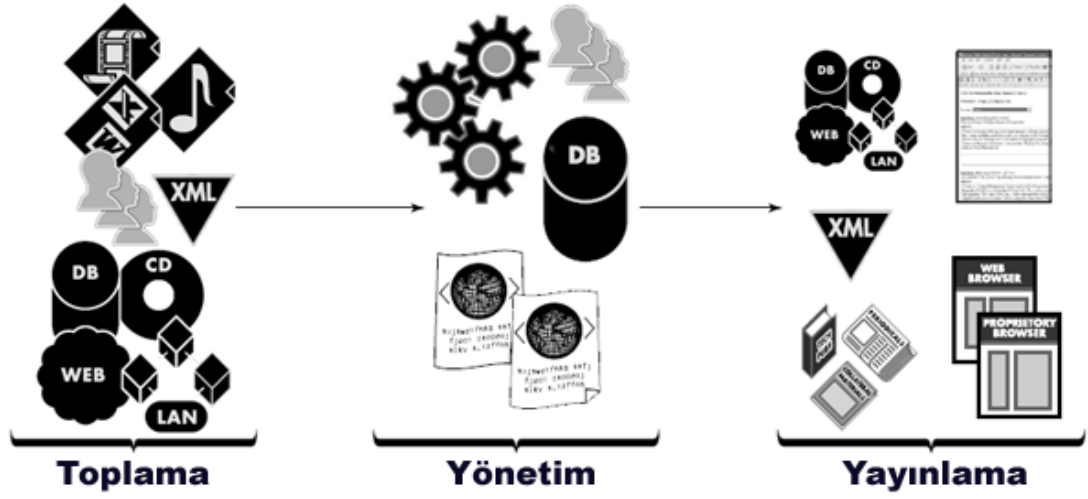
Meta-data standartları İnternet standartları üzerinde yayınlanırlar. Bu metot, mevcut bir standart üzerine yeni bir standart oluşturulmak istendiğinde, içerik alışverişi konusunda son derece kullanışlıdır. Önde gelen bir standart teknoloji, kaynak tanımlama iskeleti’nin Resource Description Framework-(RDF) yazımını belirleyen, W3C tarafından tavsiye edilen, genişletilebilir işaretleme dilidir (XML). Meta-data ve XML, uçtan-uca ağ bağlantısı (Peer To Peer), anlamsal web (Semantic Web), şebekesel hesaplamalar (Grid computing) ve yardımcı ortam (Hypermedia) gibi çok sayıda gelişmiş teknolojide kullanılmaktadır [5,27].

## **2.9. İşleyiş Bakımından WİYS**

İçerik yönetimi, işleyiş bakımından üç temel unsur ile ele alınabilir [20]. Bunlar:

### **2.9.1. Toplama**

İçerik toplama işlemi daha önceden mevcut bir kaynaktan elde edilebileceği gibi sıfırdan oluşturmak şeklinde de yapılabilir. Kaynağa bağlı olarak, bilgi ihtiyaçları doğrultusunda biçimlendirmeyi gerektirebilir (örneğin, XML formatına dönüştürme). Sonuçta toplama, bilgiyi üzerinde düzenlemeler yaparak yada bileşenlerine ayırarak ve doğru meta-data bilgisini ekleyerek sistemde biriktirme işlemidir. Web araçları günümüzde, çok az bir HTML bilgisi ile sayfalar üzerinde temel düzenlemeleri yerine getirebilecek ölçüde temel yönetim şablonlarına, stillere, yardım araçlarına sahiptirler. Bu araçlar site düzenleme işini yapanlara, sağlam ve iyi düzenlenmiş içerik oluşturmalarında yardımcı olurlar.



Şekil 2.8: İşleyiş bakımından içerik yönetimi yapısı

### 2.9.2. Yönetim

İçerik bileşenlerini ve yönetimle ilgili (örneğin sistem kullanıcıları bilgisini içeren) verileri taşıyan veritabanı kayıtlarından yada dosyalardan oluşan bir bilgi havuzu oluşturma işlemidir. Web araçları durum takibi ve yorumlama özellikleriyle temel iş akışını gerçekleştirirler. Yönetimin çekirdeğini teşkil eden bölüm, temelde belli derecelerde kullanıcı kısıtlamaları uygulayan bir dosya sistemi, site taslakları veya diyagramlar ve bağlantı yöneticilerinden oluşur.

### 2.9.3. Yayınlama

Web siteleri, yazılı medya, e-posta gönderimleri gibi bilgi havuzunun dışında kalan hedeflenmiş kanallara, içeriği taşıyarak sunma işidir. Yayınlama, yazılım mühendisliği olarak ele alındığında son ürünün kullanıcıya takdimi şeklinde ifade edilebilir [5]. Yayınlar, doğru şekilde düzenlenmiş bileşenleri, fonksiyonelliği ve standart olarak hazırlanmış etraflı bilgiyi ve yönlendirme sistemini kapsar. Web araçları, standart sayfalar üretmeyi ve bunları sayfalar üzerinde biçimlendirmeyi sağlayan sayfa şablonlarını barındırırlar. Sayfalar üzerindeki resimler, standart metin blokları, otomatik üretilmiş temel seyir sistemi gibi kaynakları paylaşabilir. Site yayınlama yöneticileri, bilgisayarda yerel olarak hazırlanan siteyi genel kullanıma açmak için web sunucusuna yüklemeyi sağlarlar.

Web sitesine ilişkin her tür varlığın, projenin hazırlandığı ortamdan “canlı sunucu” tabir edilen web sitesinin yayınının yapıldığı ortama alınması şeklinde gerçekleşir. Genellikle, web sitesinin HTML biçiminde sunulması yoluyla olur. Yayın biçimi bir e-posta mesajı olabileceği gibi bir Adobe PDF dosyası, RTF dosyası, RSS, RDF veya WML biçiminde de olabilir.

## **2.10. WİYS’de İş Akışı**

Gerçekte WİYS bir üründen çok bir kavram ve projedir. Bu kavram bir işlemler kümesini kapsar. WİYS’in temel amacı etkili ve verimli web yayınına sağlayan işlemlerin bütünleştirilmesi ve otomasyonunun artırılmasıdır. WİYS, yazarlık aşamasından yayınlama aşamasına kadar geçen yolda, bir akış şeması dahilinde gerçekleştirilen tüm işlemleri yönetir ve içerik depolama ve bütünleştirme işleri için bir sistem sağlar [28].

İş akışı, yazarlık ve yayınlama arasında içeriğin attığı adımların yönetimidir. İş akışı, geliştirme işinden bakım işine kadar olan işlemler arasında yapılan bir iş birliğidir. Düzenli ve sık aralılarla tekrarlanan işlemlerin, hatasız ve hızlı bir şekilde yerine getirilebilmesi için, geliştirme aşamasında iken işlemlerin akış sıralamalarının ve hangi öğelerce yerine getirileceğinin önceden belirlenmiş olması gerekir. İş akışı içerisinde bağlantıların kontrolü, içeriğin değerlendirilmesi ve yetkililer tarafından onaylanması tipik adımlar olarak değerlendirilebilir. İş akışı, en uygun şekle sokma metotları kullanılarak verimliliğin artırılmasını, zaman kayıplarının önlenmesini, organizasyonun temel iş mantığının otomatikleştirilmesini ve hatalı işlemlerin en aza indirilmesini sağlar. WİYS içerisinde iş akışının planlanması içeriğin üretimi, gözden geçirilmesi ve onaylanması aşamalarında bilgi akışını otomatikleştirme imkanı sunar.

İş akışı, aşağıdaki ifadelerle tanımlanabilecek çalışma prosedürünün işletimsel yönüdür [29].

- Hangi görevler var olacak,
- Görevleri kimler yerine getirecek,
- Birbirleriyle ilişkileri ve sıralamaları nasıl olacak,



- Eşzamanlılık nasıl sağlanacak,
- Görevlerin gerçekleşmesi için bilgi akışı nasıl olacak,
- Görevler nasıl takip edilecek.

İş akışı, WIYS’de yerine getirilmesi gereken sıralı veya birbirine paralel görevlerin akışının oluşturulması sürecidir [20].

WIYS’in içerdiği varsayılan iş akışları kullanılabilir veya değiştirilebilir özelliktedir. Bazı WIYS’lerde sınırsız sayıda iş akışı tasarlanabilmektedir. Eğer iş akışı belirlenmemiş bir içerik tipine bağlı yeni bir içerik maddesi oluşturmak istenirse, öncelikle iş akışı seçimi yapılmalıdır. Genel olarak bir WIYS’de üç farklı tipte iş akışı vardır [20].

WIYS’in kurulumu ve ilk başlangıcı için iş akışı aşağıdaki şekilde tanımlanabilir:

- Sistem kullanıcıları doğru roller ve sorumluklar ile ilişkilendirilmeli.
- Üretim iş akışı, ortama ve ihtiyaçlara göre oluşturulmalı.
- Üretim iş akışına farklı yaşam döngüleri eklenmeli.
- Ön tanımlı şablonlar kullanılmadan önce ayarlamaları yapılmalı ve kontrol edilmeli.
- Farklı kullanıcılara bu iş akışını kullanmak üzere izinler tanımlanmalı.
- İş akışı süreci boyunca, bir sistem hareketi olduğunda gerekli kişilere e-posta uyarıları gönderilmeli.

WIYS’in çekirdeği kurulduktan sonra, WIYS’in çeşitli bölümleri iş ihtiyaçlarını karşılamak üzere özelleştirilebilir:

- İçerik tipinin kullanıcıdan kullanıcıya değişeceği fikri doğrultusunda içerik kategori ağaçlarının farklı tipleri üretilir.
- Gerekirse site için içerik üretilir.
- WIYS’e içeriğin eklenebileceği formlar tasarlanır

WIYS kullanıcılar ve yöneticiler için ayarlandıktan sonra, bir sonraki adım WIYS’in kullanım iş akışının oluşturulmasıdır:

- Belirli içerik tipleri tanımlanarak yeni içerik maddeleri oluşturulur.
- İçerik maddelerine yeni özellikler ve içerik eklenir.
- İçerik maddeleri belli dizinlerde tutulur.
- Tüm içerik maddeleri yönetici tarafından kontrol edilir ve onaylanır.
- Son olarak içerik maddeleri yayınlanır.

Bir WİYS’de iş akışı genellikle, sırasıyla planlama safhasıyla başlar, geliştirme, yönetim, yayınlama, değerlendirme safhalarıyla devam eder ve saklama, arşivleme yada yok etme adımıyla sonlanır (Şekil 2.9). Bu aşamalar daha ayrıntılı bir biçimde aşağıda incelenmektedir [28-30].



Şekil 2.9: Web sitesi içerik yönetim sisteminde iş akışı

### 2.10.1. Plan

Hangi bilginin değerli olduğu, nasıl elde edileceği, ne zaman, kimler tarafından, hangi kanallar üzerinden, hangi teknolojiler kullanılarak, kimlere aktarılacağına tespit edildiği aşamadır. Üzerinde en çok düşünülmesi ve çalışılması gereken safha olduğu söylenebilir. Öncelikle amaçlar ve hedefler tam bir kesinlikle belirlenmeli sonrasında bunların hangi metotlar kullanılarak kimler tarafından ne şekilde gerçekleştirileceği kararlaştırılmalıdır. Sorumluluklar ve roller kesin çizgilerle belirlenmeli iş akışının nasıl gerçekleşeceği tüm yönleriyle ortaya konmalıdır. Farklı rollerdeki kullanıcıların farklı ihtiyaçlar doğrultusunda içeriğe erişmesi gerekliliği göz önünde bulundurulmalıdır. Genel amaç; doğru içeriğin, doğru kanaldan, doğru zamanda, doğru kişiye ulaştırılması ve bunun ölçülebilmesidir.

### **2.10.2. Geliştirme**

Planlama safhasının ardından alınan kararlar doğrultusunda, içerik üretimine ve üretim için gerekli araçların hazırlanmasına geçilir.

### **2.10.3. Yönetim**

Tüm iş akışının yönetiminin yapıldığı aşamadır. Burada yetkiler ve roller doğrultusunda onaylama veya reddetme, düzeltme veya bir sonraki safhaya geçiş yapma işlemleri gerçekleştirilir.

### **2.10.4. Yayınlama**

Üretilen ve yayınlanması onaylanan içerik bu aşamada yayınlanacak ortamlara sevk edilir. Bu ortamlar bir web tarayıcı olabileceği gibi bunun dışında herhangi bir sayısal ortam veya basılı bir materyal de olabilir.

### **2.10.5. Değerlendirme**

Yapılan tüm çalışmalar bu aşamada değerlendirmeye tabi tutulur. Planlama safhasında belirlenen hedef ve amaçlara ne derece ulaşıldığı varsa aksaklıklar ve eksiklikler tespit edilmeye çalışılır. Buradan elde edilecek bilgiler yeni hedeflerin belirlenmesini veya mevcut hedeflerde yenilemeye gidilmesini gerektirebilir. Değerlendirmede dikkate alınacak en önemli ölçülerden birisi de kullanıcılardan alınacak tepki, istek ve önerilerdir.

### **2.10.6. Saklama**

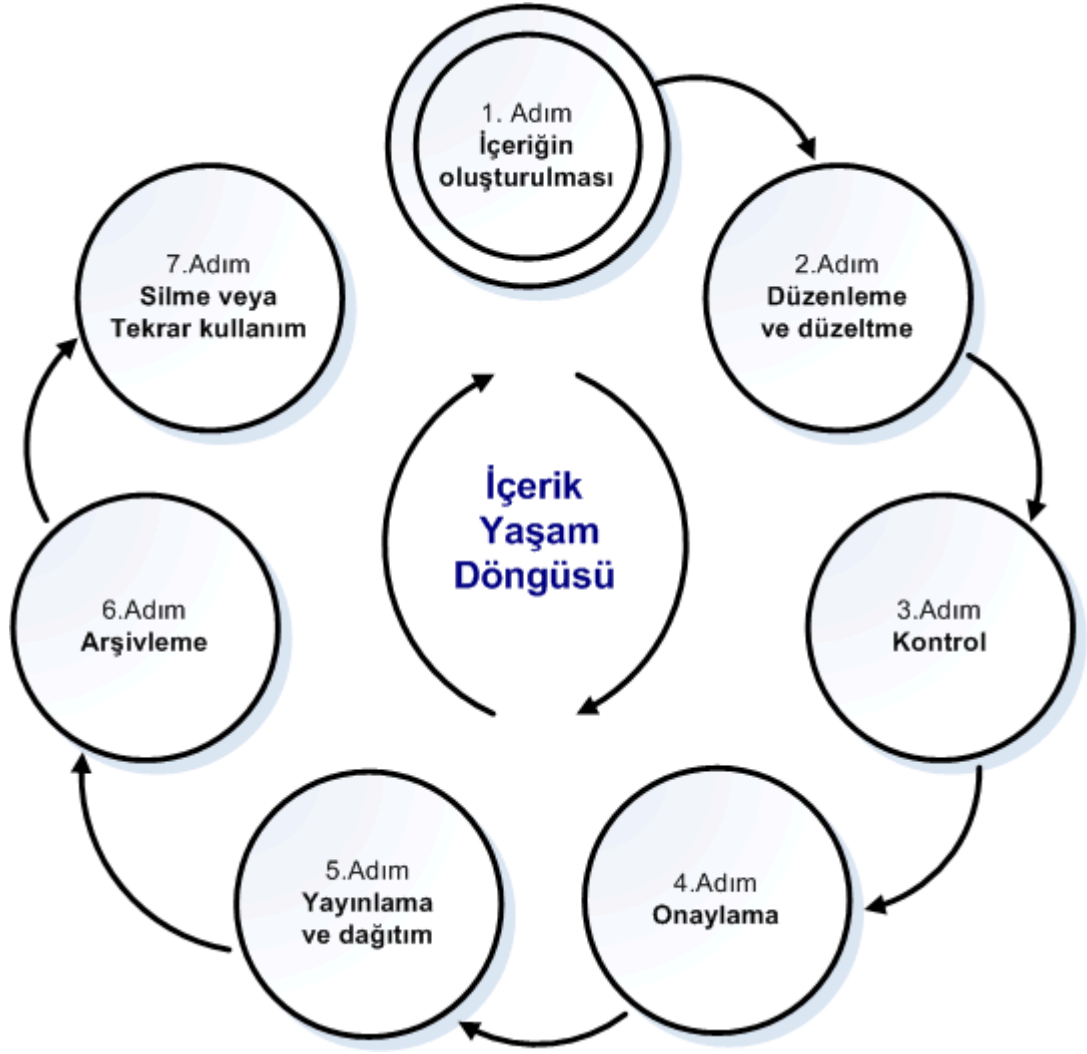
Üretilen ve yayınlanmış olan içerik, değerlendirme safhasından sonra ya yok edilir yada yeniden kullanılacağı göz önüne alınarak sürüm bilgisi eklenerek farklı ortamlarda arşivlenir. Saklama, hazırlanmış ve onaylanmış içeriğin bir havuza yerleştirilmesi işlemidir. Bu işlem içerisinde birden fazla yazarın bulunduğu projelerde, aynı içeriğe müdahale edildiğinde oluşabilecek çakışmaları önlemek

amacıyla, içeriğe sürüm numarasının verilmesi de vardır. Aynı zamanda, içerik yapısal parçalara bölünerek anlamlı bileşenlere ayrılabilir ve ayrı birimler şeklinde saklanabilir. Bu bilgiler bir veri tabanı içerisinde kayıtlar şeklinde tutulabileceği gibi bir XML dosyası şeklinde de tutulabilirler.

#### **2.10.7. İçerik yaşam döngüsü**

Sayısal ortamda içeriğin yaşam devri şu yedi aşamadan oluşur (Şekil 2.10):

- 1) İçeriğin oluşturulması ve kaydedilmesi,
- 2) Düzeltilmesi ve düzenlenmesi,
- 3) Kontrol edilmesi, gerekirse tekrar düzeltilmesi,
- 4) Onaylanması,
- 5) Yayımlanması ve dağıtılması,
- 6) Arşivlenmesi,
- 7) Bir süre sonra kullanımdan kaldırılması yada tekrar kullanılmak üzere güncelleme amaçlı kontrol edilmesi [31-33].



Şekil 2.10. İçerik yaşam döngüsü

## 2.11. WİYS’de Kullanıcılar

WİYS’ler genellikle dört farklı seviyede kullanıcıya izin verirler [20]. Bunlar:

### 2.11.1. Genel kullanıcı (ziyaretçi)

Genelde içeriğin sadece yayımlanmış yada yayımlanmak üzere olan halini görme hakkına sahip, içerikten faydalanan son kullanıcıdır. Bunlar, sitede yayında olan tüm kaynaklara erişebilen, site içerisindeki içerik içinde dolaşabilen normal kullanıcılarıdır. Bu tip kullanıcılar için en önemli gereklilikler; içeriğin düzgün bir şekilde organize edilmiş olması, kullanıcı dostu bir görünümünün olması ve kullanıcı tercihleri doğrultusunda kişiselleştirilebilir olmasıdır. Organizasyonların

çoğunda site üzerinde büyük miktarda bilgi bulunmaktadır ve kullanıcı için faydalı bilgiye erişim oldukça güçtür. Farklı WİYS araçları kullanılarak yararlı bilgi çıkartılıp özetlenebilir ve düzgün bir biçimde sunulabilir. Temel sorun, genel kullanıcıların ihtiyaçları olan bilgileri nasıl bulabilecekleridir. Binlerce sayfalık içerik eğer herhangi bir yapıya ve seyir sistemine sahip değilse değersizleşir. Burada arama motorları devreye girerek kısmi bir çözüm sağlasalar da kullanıcılar bütünlük ve karşılıklı bağlantılara ihtiyaç duyacaklardır. Çalışır bir yapının oluşturulması için bir bilgi mimarı ve profesyonel bir indeksçinin kullanılması gerekebilir. WİYS içerisinde verilen bir arama motoru, kullanıcıların web sayfaları içerisinde anahtar kelimeler kullanarak arama yapmalarını sağlar. WİYS kullanarak bütün yazılı belgeler arama motorunda indekslenerek, ziyaretçilerin İnternet tarayıcıları vasıtasıyla arama yapabilmeleri sağlanabilir.

### **2.11.2. Üyeler**

Bunlar içerik bölümünde kendi iş akışlarını seçebilen kullanıcılardır. Yetkili üyeler içerik ve diğer kullanıcılar üzerinde görüntüleme, onay, değiştirme, iptal etme, ihraç etme gibi farklı olayları gerçekleştirebilirler ve olayları raporlayabilirler.

### **2.11.3. Yazarlar ve editörler**

Yazarlık, pek çok kullanıcının tek satırlık bir metinden dahi oluşsa, web içeriği üretebildiği yetkilendirilmiş ve yönetilen bir ortam içerisinde gerçekleştirilen bir işlemdir. WİYS'in sağlaması gereken avantajlardan birisi de özel yeteneklere ve bilgi gerektirmeden hızlı ve kolay bir şekilde içerik üretimine olanak veriyor olmasıdır. Bunun yanında, “bir kez yaz ve her yerde tekrar kullan” yaklaşımı gereğince bir defa oluşturulacak içerik farklı biçimlerde ve farklı yayın ortamlarında tekrar kullanılabilir olmalıdır [29].

Yazarlar içeriğin oluşturulması ve bazen de düzeltilmesinde yer alırlar. Editörler içeriğin son düzenlemesinin yapılması ve yayım şeklinin belirlenmesiyle uğraşırlar. Sıradan bir WİYS, yazarlar için güçlü, verimli ve kullanımı kolay bir ortam sunar. Yazarların HTML veya başka bir teknik bilgiye sahip olmaları gerekmez. WİYS

binlerce sayfalık içeriği yönetecek, farklı biçimlerde ve platformlarda yayınlayabilecek araçlar sunar. Bir WIYS'in yazım desteği, ekstradan araçlar gerektirmeyecek şekilde güçlü olmalıdır. Materyaller hazırlanıp oluşturulduktan sonra içerik sağlayıcılar bunları önceden hazırlanmış tasarım şablonlar içerisinde görüntüleyebilirler. Teknik olmayan bir (içerik üretici) yazar da içeriği güncelleyebilmeli, bir başka kademelendirme ortamına veya web sitesinde yayına geçişini sağlayabilmelidir.

#### **2.11.4. Yöneticiler**

Yöneticiler web sitesinin tüm biçim ve içeriğini yönetirler. Bu rolün ayrıcalıkları; yönetsel roller, iş akışı, yaşam döngüleri, sınıflandırma ve içerik tiplerini belirlemesidir. Yöneticiler tüm web yayım sistemi üzerinde yönetim yetkisine sahiptirler. WIYS üyeleri içerisinde, yönetici grubu en geniş yetkilere sahip olanlardır. WIYS yöneticileri sitelerin oluşturulmasından, yeni kullanıcıların eklenmesinden, kullanıcı gruplarının tanımlanmasından, kullanıcı yetkilerinin belirlenmesinden ve sitelerin güvenlik ayarlarından sorumludurlar. WIYS yöneticisi yapının bakımıyla, sitenin bütünlüğünün sağlanmasıyla, sitenin istek ve ihtiyaçlarını farklı uygulamalar kullanarak karşılamakla görevlidir. Yönetici aynı zamanda tüm WIYS kullanıcılarının iletişim kuracağı ilk kişidir.

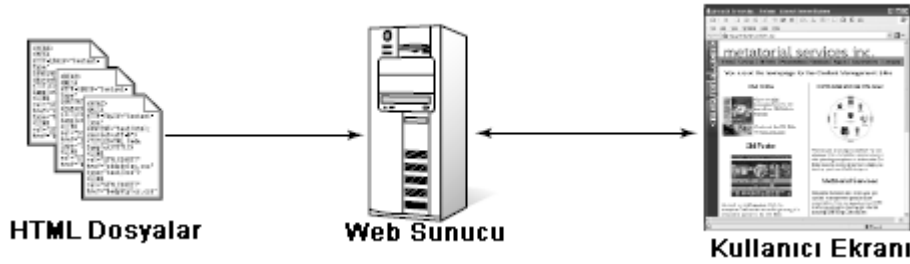
### 3. WEB SİTELERİ ve WEB SİTELERİNDE ŞABLON KULLANIMI

Web sitelerinin üretiminde kullanılan içerik yönetim sistemleri, yazılım ve donanımın bir birleşimi olmakla birlikte aslen bilgisayar altyapısından daha öte bir yaklaşımı beraberlerinde getirirler. Bu bölümde işleyiş yapılarına göre web sitesi türleri ele alınmakta, web sitesi üretiminde şablon kullanım teknikleri ve gerekliliği üzerinde durulmaktadır.

#### 3.1. Yapılarına Göre Web Sitesi Türleri

Web siteleri işleyiş yapıları bakımından temel olarak dört şekilde gruplanabilir [20]. Aşağıda, bu dört işleyiş yöntemi ele alınmakta ve avantaj ve dezavantajları belirtilerek değerlendirmeleri yapılmaktadır. Web sitesi yönetim sistemi olarak hangi yapının kullanılacağı, gerçekleştirilecek projenin ihtiyaçları doğrultusunda değişmekte olup kurulum ve kullanım maliyetleri farklıdır.

##### 3.1.1. Durağan (Statik) web siteleri



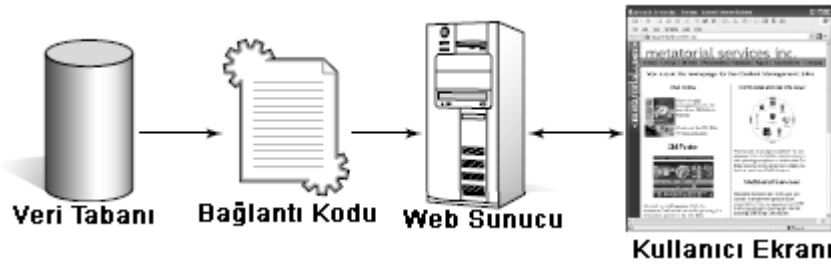
Şekil 3.1: Durağan (statik) web sitesi yapısı

Statik web sitesi yapısında, yayınlanacak ve yayında olan tüm içerik dosyalar halinde web sunucusunda barındırılır. Ziyaretçinin İnternet tarayıcısı isteği üzerine, web sunucusu tarafından yayınlanır. Statik web siteleri, içeriğin sıkça değişmediği, ziyaret sayısının yüksek olduğu durumlarda tercih edilirler, hızlı çalışmaları ve sunucu sisteme daha az iş yükü bindirmeleri sebebiyle avantajlıdır.



### 3.1.2. Değişken (Dinamik) web siteleri

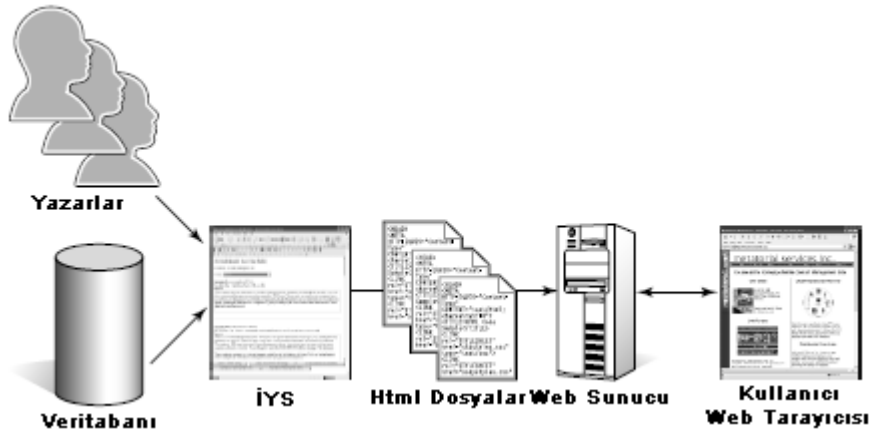
Dinamik web sitesi yapısında, içeriğe ilişkin bilgiler bir veya daha fazla veritabanında tutulmaktadır. Veritabanı ile bağlantı yapılmasının destekleyen dinamik bir web dili vasıtasıyla, ziyaretçinin İnternet tarayıcısının isteği üzerine, o anda veritabanı sorgulanarak üretilir ve yayınlanır. Dinamik web sitelerinde içerik son derece günceldir, çünkü her sayfa ziyaretçinin isteği ile o anda hazırlanmaktadır [34].



Şekil 3.2: Değişken (dinamik) web sitesi yapısı

### 3.1.3. Statik web sitesi üreten WİYS'ler

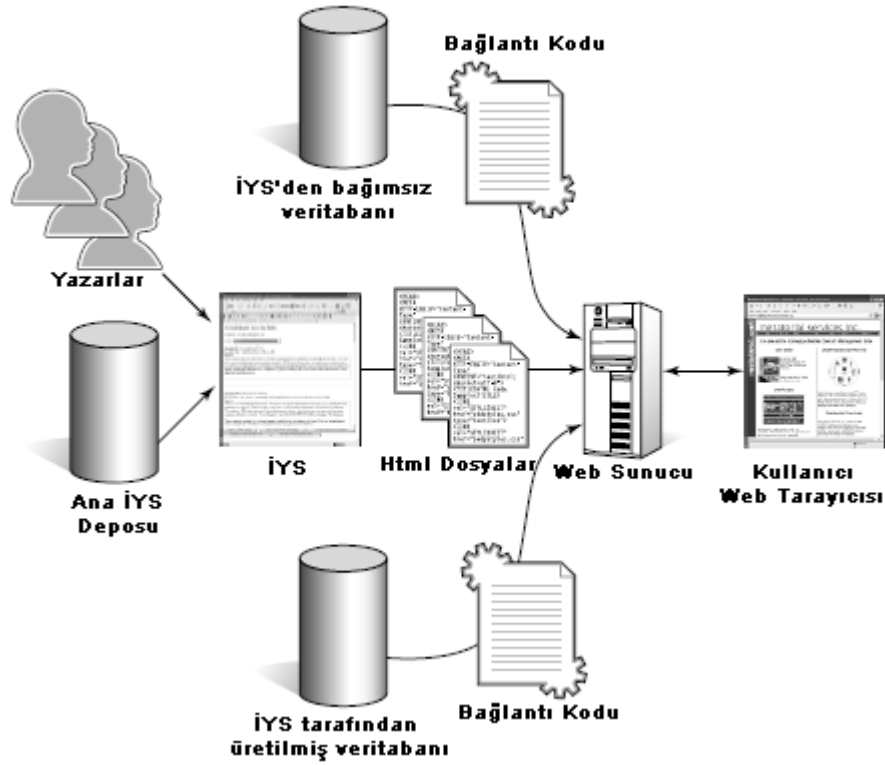
Statik web sitesi üreten dinamik yapılarda, içeriğe ilişkin veriler veritabanlarında tutulurken, WİYS aracılığıyla statik web sayfaları üretilir. Statik sayfaların üretimi dinamik olarak gerçekleşirken, yayında olan sayfalar statik yapıdadır. Bu yöntemin avantajı, hem dinamik hem de statik web sitelerinin avantajlı yönlerini birleştirmesidir.



Şekil 3.3: Statik web sitesi üreten WİYS yapısı

### 3.1.4. Statik ve dinamik web sitesi üreten WIYS'ler

Statik ve dinamik web sitesi üreten sistemler yukarıda değinilen sistemlerin bir karmasını teşkil etmektedir. Sitenin hızlı çalışması gereken ve içeriği sık güncellenmeyen bölümleri statik yapıda tutulurken, içeriğin güncelliğinin önemli olduğu bölümlerde dinamik sayfa üretimi tercih edilebilir. Birden fazla uygulamanın ve veritabanının kullanıldığı geniş ölçekli uygulamalar için tercih edilebilir.



Şekil 3.4: Statik ve dinamik web sitesi üreten WIYS yapısı

### 3.2. Web sitelerinde şablon kullanımı

Şablonlar, çıkış dokümanını elde etmek amacıyla herhangi bir şablon motoru kullanarak; değiştirilmek, işletilmek, üzerinde oynamalar yapmak yada çalıştırılmak için içerisinde işaretçilerin yer aldığı tipik HTML belgeleridir [35]. Bir başka tanıma göre şablon, basit işlemler sonucunda veriyle doldurulabilen boşluklar içeren bir HTML dokümanıdır [10]. Şablonlar tasarımcılar tarafından da programcılar tarafından da doğrudan okunabilirler. Şablon görünümü, model görünüm kontrolörü (model view controller) yapısının görünüm kısmına karşılık gelmektedir [10,36]. Bir

web şablon sistemi, web sayfalarının ve web dokümanlarının web sitelerinde yayını ve İnternet üzerinden ulaştırılması amacıyla kullanılan yazılım ve yöntemleri tanımlar [37].

Bir şablon motorunun kullanılmasının ardındaki asıl amaç, gösterim mantığı ile veri kompozisyonlarını gösterim aşamasında birbirinden ayırmaktır. Web sitesi geliştirilmesi sırasında HTML içerisinde kod kullanmamak, kodun içerisinde de HTML kullanmamak olarak özetlenebilir [10,11].

Bir web şablon sistemi aşağıda verilen üç ana bölümden meydana gelmektedir.

- 1) Şablon Motoru: Sistemin birincil işlem elemanıdır.
- 2) İçerik Kaynağı: Giriş veri akımlarının her türünü kapsar. Örneğin, ilişkisel veritabanı, XML dosyaları, LDAP dizini, yerel ve genel ağ üzerinde erişilebilir tüm veriler.
- 3) Şablon kaynakları: Bir şablon diline özgü olarak hazırlanmış web şablonları.

Şablon ve içerik kaynakları şablon motoru tarafından işlenip birleştirilerek web dokümanlarının üretilmesi sağlanır. Yayımlanacak ortama göre, HTTP protokolü veya bir başka İnternet protokolü kullanılarak birbirinden farklı çıkış biçimlerinde oluşturulabilir.

Şablon kullanımındaki amaç, web tasarımcısının tekrarlı ve rutin işlerinin otomatik hale getirilmesidir. Böylelikle; görünüm bütünlüğü, yapısal bütünlük, üretim hızının artması, içerik üzerinde etkin kontrol, içerik ile yapının birbirinden ayrılması sağlanır ve toplam iş yükü azaltılmış olur.

Web siteleri genellikle düzenli içerik güncellemelerine ve görünümün standartlaştırılmasına ihtiyaç duyarlar. Standart bir stil kullanmadan farklı türlerde içeriğin gösterilmesi kullanıcıda kaybolmuşluk hissi uyandırabilir. Standardizasyon aynı zamanda web sitesinin kimliğinin de önemli bir parçasıdır.

Şablon sistemi, standardizasyonu otomatikleştirirken sitil sayfaları ve seyir çubukları kullanarak site içerisindeki yazı tipi, seyir yapısı bütünlüğünü sağlar. Web

geliştiricilerini ve tasarımcılarını sitenin bakım ve içerik ile ilgili işlemlerinden kurtarır.

İçerik ve kod ayrımı tekniği; kodun daha açık yazılması, bileşenlerin tekrar kullanılabilmesi, bakımının kolaylığı, güvenliğin kontrol edilebilmesi, değiştirilebilir görünüm özellikleri, tek bir noktada yapılacak değişikliğin tüm sitede aynı şekilde görüntülenmesi, daha küçük parçacıklarla çalışma kolaylığı, iş yükünün dağıtılması gibi avantajlar ve üstünlükleri sebebiyle programcılar ve tasarımcılar arasında tercih edilmektedir [38].

PHP'nin ortaya çıkışının temel sebebi, verilerin bağlanması işinin C/C++ uygulamaları tarafından gerçekleştirilerek HTML formatına dönüştürülmesidir. Fakat son zamanlarda gelişerek veritabanlarının birbirine bağlanması ve web servisleri gibi görevlerde de kullanılmaktadır. Bulanıklaşan bu ilk çıkış amacı, temel yapısı web şablon dili olan PHP uygulamalarının uygulama ve görünüm mantık katmanlarında kullanımını ortaya çıkarmıştır. Şablon motorlarının büyük bir çoğunluğu kendi şablon yazım stillerini kullanırlar. PHP bu dili parçalar halinde yorumlayan bir araç haline gelmiştir.

PHP ile hazırlanmış web uygulamaları temel olarak beş katmanda ele alınabilir [11].

- 1) Gösterim Katmanı: Web tarayıcısının bulunduğu katmandır.
- 2) Gösterim Mantık Katmanı: HTML kodunu üreten PHP kodu. Gösterim ve kontrol modellerinin bulunduğu katmandır.
- 3) Alan Mantık - Uygulama Mantık Katmanı: Üstteki katmanlar için alttaki katmanlardan bilgi dönüştürülmesi ile ilgilenen PHP kodunun yer aldığı katmandır.
- 4) Veri Erişim Katmanı: Veritabanından veya dosya sisteminden veri getirmeye ilgili katmandır.
- 5) Veri Katmanı: Veritabanı (Ör: MySQL) veya bilgi depolayan dosya sisteminin yer aldığı katmandır.

### 3.2.1. Gösterim mantığı ve şablonlar

Web şablon sistemleri konusunda üzerinde karar verilmesi gereken anahtar konulardan birisi de gösterim mantığının şablonlar kullanılarak ne şekilde gerçekleştirileceğidir. Bu konuda yapılmış ve halen uygulanmakta olan beş tür metot bulunmaktadır. HTML dilinin kendisi ardışık bir gösterim mantığını desteklemez, bu nedenle ya HTML tekrarlanarak çoğaltılır veya tekrarlı çoğaltma yapısı programın içerisine yerleştirilir.

Gösterim mantığını anlatabilmek için genellikle iki kavramdan bahsedilmektedir. Bunlardan ilki bildirimsel ikincisi ise buyurgandır. Bildirimsel yapıda, işlenecek öğelerin ne olduğu tanımlanır. Buyurgan yapıda ise işlemin nasıl olacağı belirtilir. PHP, C ve Java gibi en popüler programlama dilleri buyurgan biçimi kullanırken, SQL ve HTML ise bilinen bildirimsel dillerdendir [11].

Bunun yanında şablonlardaki gösterim mantığında çok güçlü bir dile ihtiyaç duyulmaz. Güçlü bir dil kullanmanın getirisi esneklik olurken, dezavantajı karmaşıklıktır. Şablon tasarımcısı ile uygulama tasarımcısı arasında bir denge söz konusudur. Birinin kazandığı yerde diğeri kaybetmektedir.

Programlama dillerinin güç dereceleri çözüm getirdikleri problemlerin türlerine ve sayılarına göre farklılık göstermektedir. HTML gerçek bir programlama dili sayılamayacak kadar güçsüzdür [11].

Bildirimsel ve tanımlayıcı doğasıyla HTML, gösterim mantığı için iyi bir uyum sağlar. Gösterim mantığı için, düşük güçlü ve açıklayıcı yazım tekniği ile HTML tasarımcısı açısından çok daha kolaydır.

Buyurgan yazım tekniğinde, HTML içerisine serpiştirilmiş şablon motoru komutları yer almaktadır. Örnek olarak, buyurgan bir yazımın kullanıldığı bir projeden yapılan alıntı aşağıdadır:

```
{if $prev_offset >= 0}
```

```

[<a class="prevnext" href="tiki-
webmail.php?section=contacts&find={$find}&offset={$prev_offset}&sort_mode={$sort_mode}">{tr}prev{/tr}</a>]&nbsp;
{/if}
{tr}Page{/tr}: {$actual_page}/{ $cant_pages}
{if $next_offset >= 0}
&nbsp;[<a class="prevnext" href="tiki-
webmail.php?section=contacts&find={$find}&offset={$next_offset}&sort_mode={$sort_mode}">{tr}next{/tr}</a>]
{/if}
{if $direct_pagination eq 'y'}
{section loop=$cant_pages name=foo}
{assign var=selector_offset
value=$smarty.section.foo.index|times:$maxRecords}
<a class="prevnext" href="tiki-
webmail.php?section=contacts&find={$find}&offset={$selector_offset}&sort_mode={$sort_mode}">
{$smarty.section.foo.index_next}</a>&nbsp;
{/section}
{/if}

```

Aynı kod parçasının bildirimsel şekli aşağıdaki gibi olabilir:

```

<page:navigator id="pagenav" items="20">
  <page:first>First</page:first> <page:prev>Prev</page:prev>
  <page:list>
    <page:number>
    <page:elipses>...</page:elipses>
    <page:separator> </page:separator>
  </page:list>
  <page:next>Next</page:next> <page:last>Last</page:last>
</page:navigator>

```

Buyurgan tekniğin üstünlüğü yazımının nispeten mümkün ve kolay olması iken, dezavantajı tekrarlı yazımı gerektirmesidir.

DyNA içerisinde kullanılan şablon-bileşen yapısı, her iki tekniğin de üstünlüklerini bir araya getirmeye dayalıdır. Yazımı kolaylaştırırken aynı zamanda tekrarlanan kod bloğu sayısını en aza indirmeye imkan vermektedir. Aşağıda DyNA içerisinde sık kullanılan bir şablon yapısının kod bloğu verilmektedir.

```

echo "Kategoriler <ul>";
$cocuk =cocuklar($id);
for ($i=0;$i<count($cocuk);$i++)
{
echo "<li><a href=\"?id=\".$cocuk [$i].\" \" class=liste>"
.tag_oku(xml_al ($cocuk[$i]),"tr_baslik"). "</a></li>";
}echo "</ul>"

```

### 3.2.2. İşaretçi stilleri

İşaretçiler (marker) şablonlar ve kod arasındaki köprünün yarısını oluşturmaktadır. İşaretçiler HTML şablonu içerisinde görünürler ve şablon içerisinde motor tarafından işlenecek yerleri gösterirler. İşaretçilerin, şablon dosyasının neresinde, ne şekilde yer alacağı ve nasıl yorumlanacağına ilişkin farklı metotlar kullanılmaktadır. İşaretçi stilleri içerisinde genel olarak kabul gören ve kullanılmakta olan beş farklı metot örneklerle birlikte aşağıda yer almaktadır.

#### 3.2.2.1. Kuralcı HTML yapısı

Bu metotta, bütün işaretçiler HTML için geçerlidir. Elemanlar kendi ID öznitelikleri ile tanımlanırlar ve şablonlar XML benzeri parçalayıcılar kullanılarak işlenirler. Bu yaklaşımın avantajı XML ve XHTML olarak standart olarak kabul edilen kurallara uygun olmasıdır. Dezavantajı ise yazımının ayrıca öğrenilmesinin gerekmesi ve HTML yapısına göre tanımlanmamış olmamasıdır. Aşağıdaki örnekte “tal” özniteliği standart dışı olmakla birlikte HTML için geçerli bir yapı oluşturmaktadır.

```
<table>
  <tr tal:repeat="item here/cart">
    <td tal:content="repeat/item/index">1</td>
    <td tal:content="item/description">Widget</td>
    <td tal:content="item/price">1.50</td>
  </tr>
</table>
```

#### 3.2.2.2. Özel etiketler

Java ile popüler olmuş bir metottur. HTML ile karışık, kullanıcının tanımladığı ek etiketler olarak tanımlanırlar. HTML yazım kuralının (eğer kullanıcı ajanı tanıyamadığı bir eleman ile karşılaşırsa elemanın içeriğini yorumlamaya çalışır) avantajını kullanır. Pek çok HTML editörü kullanıcı tanımlı etiketler üzerinde değişiklik yapamasa bile yapıyı bozmazlar. Aşağıdaki örnekte; kullanıcı tarafından tanımlanmış ve standart bir HTML tarayıcı tarafından yorumlanamayacak olan ERRORSUMMARY, list gibi etiketler kullanılmış ve yine bu etiketler için standart olmayan eklemeler yapılmıştır.

```

<ERRORSUMMARY name='Summary'>
<list:LISTITEM>
<FONT COLOR="RED">{ErrorMessage}</FONT><list:SEPARATOR>
<BR>
</list:SEPARATOR></list:LISTITEM>
</ERRORSUMMARY>

```

### 3.2.2.3. HTML açıklamaları

Bu metot şablon gösterim mantığını tanımlamak amacıyla HTML'in yorum satırı yazımını kullanır. Avantajı HTML olarak geçerli, genel olarak basit olmasıdır. Aşağıda buna ilişkin bir kod bloğu verilmiştir. HTML yorumlayıcı için yorum satırları olarak görülen ifadeler şablon motoru için veri yerleştirilecek olan bölgeleri işaretlemektedir.

```

<table>
  <!-- BEGIN_ROW -->
  <tr>
    <td><!-- Username --></td>
    <td><!-- Password --></td>
  </tr>
  <!-- END_ROW -->
</table>

```

### 3.2.2.4. Özelleştirilmiş yazım

Bu metot, şablon programlama diline bazı formların eklenmesi şeklinde yapılır. Avantajı, görünümü model ve kontrolörden ayırtmasıdır. Dezavantajı ise yeni bir programlama dilinin öğrenilmesini gerektirmesi ve HTML uyumluluğunun olmamasıdır. Aşağıda buna ilişkin, kendine has döngü yapısının yer aldığı bir kod örneği verilmektedir. Örnekte her kullanıcı için bir tablo satırında isim ve e-posta adresleri yazdırılmakta ve her satırda satır fon rengi değiştirilmektedir.

```

<table>
{section loop=users}
{strip}
  <tr bgcolor="{cycle values="aaaaaa,bbbbbb"}">
    <td>{users.name}</td>
    <td>{users.email}</td>
  </tr>
{/strip}
{/section}
</table>

```



### 3.2.2.5. Betik etiketleri

Bu metot, programlama dilini şablon dili olarak kullanır. PHP programlama dilinin kendisi, çıktıyı C/C++ kodları altında işleyen bir şablon dili olarak tasarlanmıştır. Avantajı son derece yalın ve hızlı olmasıdır. Şablonda yer alan tüm komutlar PHP motoru tarafından kolaylıkla ve başka herhangi bir ön işleme gerektirmeden tam olarak çalıştırılabilir. Şablon tasarımı sırasında MGK katmanlarının karıştırılmamasına dikkat edilmelidir.

```
<table>
<?php
foreach ( $users as $user ) {
?>
    <tr>
        <td><?=$user->name()?></td>
        <td><?=$user->email()?></td>
    </tr>
<?php
}
?>
</table>
```

DyNA, bu beş model arasından en hızlı ve yalın olan betik etiketleri metodunu esas almaktadır. Dinamik şablon bileşenleri birbirlerini çağırıp işletebilirler. Şablon hazırlanırken, bölümler ve altbölümler ne kadar iyi düşünülüp hazırlanırsa kod tekrarından da o derece kaçınılmış olur. Bu işlem bir program yazılırken hangi altprogramların yazılacağıının tespit edilmesine çok benzeyen bir durum arz eder. Çünkü DyNA için her bir şablon bir fonksiyon gibi icra edilebilmekte ve birbirleri arasında çağırılabilirler.

### 3.2.3. Verinin şablona itilmesi ve çekilmesi

Verilerin, şablonlara yerleştirilmesi için genel olarak iki yaklaşım benimsenmiştir. Genelde iteratif çalışması gereken yapılarda kullanılmakla beraber, yazım ve kullanım kolaylığına göre programcı tarafından birbirlerinin yerine tercih edilebilirler.

### 3.2.3.1. İtme metodu

Şablona yerleştirilecek bütün bilgi, şablon işlenmeden önce şablon API'si tarafından yorumlanır. Çıkış verisi kaynak konumundaki sayfa veya tablolarda yer alan elemanların tiplerine göre şekillenir [39]. Bu yöntemin avantajı büyük çaplı şablon yapılarında kullanıldığında sayfa üretimi ve hata kontrolünü kolaylaştırmasıdır. Dezavantajı ise yazımının zor, kod okunaklılığının az olmasıdır. Aşağıda verilen kod bloğu içerisinde bir döngü yapısı kullanılmış ve kaynaktan çekilen veri şablon içerisine itilmiştir.

```
<TABLE BORDER="1" ALIGN="CENTER">
{section name=i loop=$phpmodule}
<list:LISTITEM>
<TR><TD>{$phpmodule[i].Name}</TD><TD>{$phpmodule[i].Description}</TD>
</TR>
</list:LISTITEM>
{/section}
</TABLE>
```

### 3.2.3.2 Çekme metodu

Bu metoda göre, şablon API'sinin yorumlama aşamasında bağlamalara, geri çağırımlara veya olaylara izin verilebilir. Avantajı şablonun gösterimi için gereken verinin tamamına ihtiyacı olmamasıdır. Şablonun içerisinde yer alan gösterilmeyecek ve atlanacak yerler için şablon API'sinin bu alanları hesaplamadan işleme devam etmesi sağlanmıştır. Çekme metodunda veriye şablon içerisinde ihtiyaç duyulan yerlerde kaynağa erişilerek belli bölgeler için veri çekilmesi işlemi gerçekleştirilir. Aşağıdaki örnekte çekme metodunun kullanımı yer almaktadır. Çekme metodunun avantajı kullanım kolaylığı, kısalık ve okunabilirlik iken dezavantajı da verilerin geliş sırasının programcı tarafından önceden bilinmesi gerekliliğidir.

```
<list:LIST name='ExampleList'>
<TABLE BORDER="1" ALIGN="CENTER">
<list:LISTITEM>
<TR><TD>{Name}</TD><TD>{Description}</TD></TR>
</list:LISTITEM>
</TABLE>
</list:LIST>
```

DyNA kendi içerisinde hem itme hem de çekme metoduna izin vermektedir. Tüm veriler veritabanında tutulmakta, dinamik şablon bileşenlerinde yer alan PHP komutları doğrultusunda işlenmekte ve sonuç kullanıcı tarafına aktarılmaktadır. Dinamik şablon bileşenleri oluşturulurken gereklilikleri ölçüsünde her iki metot da kullanılabilir.

#### 4. DyNA'DA DİNAMİK ŞABLON KULLANIMI

İnternet üzerinde devasa miktarlara ulaşmış bilginin derlenmesi, kontrol edilip etkili bir şekilde sunumu, erişimi ve değerlendirilebilmesi amacıyla web sitesi içerik yönetim sistemleri geliştirilmektedir. Bir WIYS, içeriği oluşturan büyük miktarlardaki verilerin toplanması, depolanması ve sonrasında yayınlanması amacıyla kullanılan büyük veritabanları, dosya sistemleri ve diğer ilgili yazılım parçalarının bir bileşimidir. Bu web yayınlama sistemleri, metinleri, ses dosyalarını, video dosyalarını veya resimleri indeksleme özellikleriyle diğer veritabanlarından ayrılırlar.

Bir web sitesinin görünümünü meydana getiren sayfalar bazen farklı bazen aynı içeriğe sahip olabilirler. Bir kullanıcı sayfalar arasında dolaşırken sayfalar içersindeki veri ve içerik sayfadan sayfaya değişiklik gösterebilir fakat başlık ve yan taraflar gibi pek çok öge aynı şekilde kalmaktadır. Bütün sayfalar göz önüne alındığında her bir sayfanın yapısı ve yerleşim planı görünüm açısından benzerlik arz edebilir veya aynı olabilir. Ve bazı öğeler ya da bölümler bir sayfada bulunurken diğerinde bulunmayabilir. Bu öğelerin veya bölümlerin görüntülenip görüntülenmeyeceği genellikle görünüm uygulamaları içerisine doğrudan kodlanarak gerçekleştirilir. Daha sonradan bu kısımlarda yapılması gereken bir değişiklik veya düzeltme güç olmaktadır. Buna ek olarak site görsel ve yapısal bütünlüğünü sağlamak ve kullanıcının site içerisinde gezinirken kaybolmaması için şablon yapıları kullanılmaktadır [35].

Şablonlar genellikle içine veri tabanından çekilen verilerin yerleştirildiği, böylece web sitesinin görsel ve yapısal bütünlüğünü sağlayan statik dosyalardan meydana gelirler. Şablonlar, çıkış dokümanını elde etmek amacıyla herhangi bir şablon motoru kullanılarak, değiştirilmek, işletilmek, üzerinde oynamalar yapmak yada çalıştırılmak üzere içerisinde işaretçilerin yer aldığı tipik HTML belgeleridir [40]. Bir başka tanıma göre şablon, basit işlemler sonucunda veriyle doldurulabilen boşluklar içeren

bir HTML dokümanıdır [12]. Şablonlar aynı zamanda tasarımcının da programcının da doğrudan okuyabilecekleri HTML dosyalarıdır.

DyNA’da ise şablonlar, statik dosyalar yerine aktif bileşenlerden oluşurlar ve birbirlerini çağırıp çalıştırabilirler. DyNA, bileşik görünüm (composite view) yapısını kullanmaktadır. Bileşik görünüm diğer tekrar kullanılabilir alt görünümünün birlikte kullanımıyla oluşturulan bir görünümdür [41-44]. Alt görünümünde yapılacak tek bir değişiklik bu alt görünümü kullanan tüm sayfaları etkileyecektir. Dahası bileşik görünüm yerleşim düzenini kontrol eden bir şablon gibi bir işlev görmektedir. Sitenin geneli üzerinde tutarlı ve düzenli bir yapının elde edilebilmesi ve site genelinde yapılacak değişikliklerin kolay ve hızlı bir şekilde gerçekleştirilmesini sağlar.

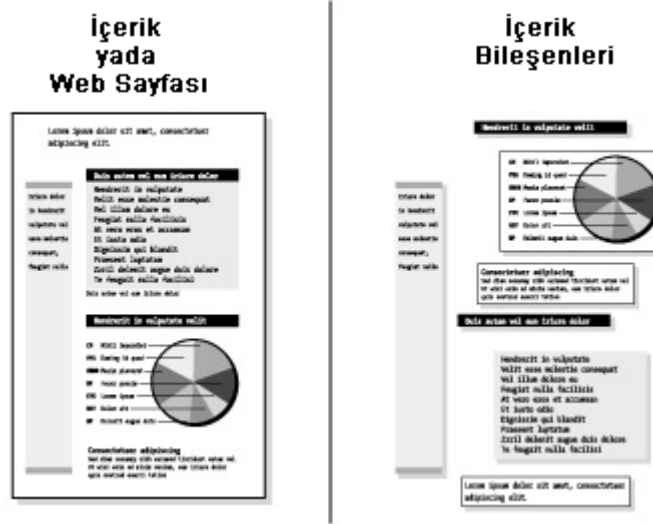
#### **4.1. İçerik Bileşeni Kavramı**

Bir web sitesi yazı, resim, ses ve video gibi çok farklı tiplerde içerikten oluşmaktadır. Bu tip bileşenleri ayırmak üzerlerinde işlem yapmayı kolaylaştıracaktır. Bunu yapmaktaki temel neden içeriğin tipine göre tasarlanmış araçların kullanılabilmesi, farklı parçacıkları birleştirerek özelleştirilmiş ve türetilmiş web sayfalarının üretilmesidir [24,45,46]. Aşağıdaki şekilde bir web sayfasını oluşturan içerik bileşenleri görülmektedir.

WİYS’ler genellikle içeriği oluşturan küçük parçaların işlenerek birleştirilmesi prensibine dayanırlar. Pek çok WİYS’in kullandığı bu küçük parçalar içerik bileşeni olarak adlandırılırlar. İçerik bileşeni bir web sayfasında yer alan bir makale ya da bir öyküyü meydana getiren öğelerden herhangi biri olarak düşünülebilir.

İçerik bileşenin ayrıştırılması; başlıklar, alt başlıklar, gövde metni, seyir, reklam alanları, yayın tarihi, kaynak gibi küçük parçacıklar şeklinde olabileceği gibi bütün bir sayfa içeriği tek bir bileşen olarak da düşünülebilir.

Şekil 4.1’de bir içerik bileşenin ne olduğunu göstermektedir. Şeklin solundaki kısımda tam bir web sayfasının diyagramı gösterilmekte sağ tarafta ise aynı sayfanın içerik bileşenlerine ayrıştırılmış hali verilmektedir.



Şekil 4.1: Web sayfası ve sayfayı oluşturan içerik bileşenleri [24]

DyNA için her bir içerik bileşeni, insan DNA’sını meydana getiren nükleotidler şeklinde düşünülebilir. İçerik bileşenleri, belirlenmiş şablon dosyaları kullanılarak dinamik olarak üretilmekte ve bir web sayfası oluşturacak şekilde birleştirilmektedir.

Aynı zamanda bu metotla, site içeriğinden sorumlu kimseler uzmanlaştıkları işleri daha iyi bir şekilde gerçekleştirme olanağına kavuşurlar. Örneğin, fotoğraf işlerinde uzmanlaşmış birinin bu işle uğraşırken yazı işleriyle ilgilenmesi gerekmez.

## 4.2. DyNA’da Kullanılan Bileşik Görünümlü, Ayrışık ve Parçalı Şablon Sistemi Yaklaşımı

Web servisleri ve çoklu ortam uygulamaları genellikle son derece değişken ve esnek ortamlar üzerinde gerçekleşmektedir. Etkileşimli ve güncel içerik sağlanması zorunluluktur. Mevcut bilgi belli sınırları aştığında bunun idaresi ve üretimi zorlu bir iş halini almaktadır. Görünüm ve işleyişte esnekliğin sağlanması web uygulamaları için temel bir ihtiyaçtır. İçerik ile yapının ayrılmasını sağlayan, bileşik görünümlü, bölünebilir bir şablon sisteminin kullanımı bir gereklilik halindedir [11].

Bir şablon, bir şablon motoru tarafından işlenmesi için içerisine çeşitli işaretler yerleştirilmiş tipik bir HTML dosyasıdır [12]. Şablon, basit işlemler yapılarak içerişi veri ile doldurulabilen boşluklar içeren bir HTML dokümanıdır şeklinde de tanımlanabilir [10]. Şablonun kullanım amacı web uygulamasının programlama dili kodunu HTML kodlarından ayıştırmaktır [11].

#### **4.2.1. Bileşik görünüm**

Bir web sitesinin sayfaları arasında gezinirken farklı sayfalar arasındaki içerik ve görünüm değişmekte fakat başlık yan bar gibi yaygın pek çok bölüm aynen kalmaktadır. Bir sayfanın bazı bölümleri ya da elemanları birkaç farklı sayfada da görüntülenmektedir. Bu elemanlar veya gruplar sayfalara doğrudan kodlandıklarında görünüme ilişkin düzenlemeler ve değişiklikler yapmak zorlaşmakta ve bütünlük bozulmaktadır. Bu ek kodlama, görünümün içine yapıldığında bütüncül bir görünümün elde edilmesi zorlaşırken bakımı da daha güç hale gelmektedir.

Bileşik görünüm, birbirinden ayrı ve yeniden kullanılabilir ayrıca tanımlanmış alt-görünümlerin bir araya getirilmesiyle elde edilen görünümdür. Bir alt görünümde yapılacak bir değişiklik bu alt görünümün kullanıldığı tüm sayfaları birden etkileyecektir. Bu sebeple, bileşik görünüm, web sitesinin görünümünü meydana getiren şablonu oluşturan alt parçacıkları yönetir. Amaç tüm web sitesi üzerinde yönetim ve değişiklik işlerinin görsel ve yapısal bütünlüğü zedelemekten geçektirilmesi [46].

Gerçek dünya uygulamalarında web sayfaları pek çok kaynaktan aldığı içeriği birden fazla alt görünüm kullanarak tek bir sayfa oluşturmak suretiyle sunmaktadırlar. Web sayfaları biçimlendirme kodunun her bir atomik yapının içine kodlanmasıyla meydana getirilirler. Dahası bu görünümler bir ağaç hiyerarşisi şeklinde birleştirilebilirler. Her bir bileşen bütününe içine çağırılabilir ve sayfanın görünümü içerikten bağımsız olarak yönetilebilir [47].

#### **4.2.2. Ayırıştırma**

Ayırıştırma tekniği uygulamaya yönelik kodlarla görünümü düzenlemeye yönelik kodların birbirinden ayrı tutulması mantığına dayanmaktadır. Uygulama kodu içeriği toparlar, şablon motoruna gönderir ve görüntüler. Uygulama kodu bu içeriğin şablonda nasıl görüntüleneceği ile ilgilenmez [48].

Programcıların ve tasarımcıların kod ile içeriğin ayrıştırılmasını istemelerinin pek çok sebebi vardır. Bu yaklaşım; anlaşılabilirlik, iş bölümünün yapılması, bileşenlerin tekrar kullanımı, tek bir noktanın değiştirilmesi, bakım, değiştirilebilir görünüm, güvenlik gibi avantajlar sağlamaktadır [34].

#### **4.2.3. Parçalama**

Pek çok web sitesi kararlı, durağan bir görünüme sahiptir. Bu temel yapı içerik değişse bile nadir olarak değişir. Web sayfasının görünümü içeriğin değişiminden bağımsız olarak aynı kalmaktadır. İçerik olarak zengin olan bu tür sitelerde tematik alanlara ayrılmış bölgeler ayrık alanlar olarak belirlenebilir [44].

Parçalama yaklaşımı kullanılarak dinamik sayfaların ön belleğe alınması işlemiyle uğraşmadan büyük bir performans artışı elde edilebilir. Parçalama temelli yaklaşım, web sayfalarının üretilmesini kolaylaştırırken performansı da artırmış olur. Aynı görünüme sahip bir dizi web sayfasının tasarlanması kolaylaşır. Aynı zamanda genel bilginin birkaç farklı web sayfasına yerleştirilmesi kolayca gerçekleştirilebilir. Sadece bir parça üzerinde değişiklik yapılarak pek çok sayfada birden değişiklik elde edilmiş olur.

Eğer web sayfaları statik bir dosyalama sistemi şeklinde depolanıyorsa bütün sayfaların bir kerede güncellenmesi zordur, fakat parçalama yaklaşımını kullanan dinamik bir sitede, bir parçaya yapılacak müdahale ile bütünlük bozulmadan, anında gerçekleştirilmiş olur [47,49].



### 4.3. DyNA’da Dinamik Şablon Bileşenleri

Web sayfalarının sayıları ne kadar çok olursa bunları üretmek ve yönetmek için yapılacaklar da o derecede sistematik olmalıdır. Bu sebeple çalışmamız, web sayfalarını parçalara bölerek yönetme konusunu ele almaktadır. Web sayfalarında şablon-bileşen kullanılarak yapılan parçalama yaklaşımı pek çok avantajının yanında esneklik ve bütünlüğün sağlanmasına da yardımcı olmaktadır [8].

DyNA, web sayfalarını, sınırlandırılmış belli kurallarla birbirine bağlanmış, farklı özelliklere sahip nesneler olarak ele alır. Farklı özellikler sayfada yer alan uygulamalar şeklinde olabileceği gibi, sadece görünüme ilişkin farklılıkları da içerebilir veya her ikisinin bir birleşimi olabilir.

Şablon bileşenleri, nesneye yönelik programlama yaklaşımında yer alan metot ve özellik işlevlerine benzer hizmetleri tek bir çatı altında yerine getirebilen kod parçacıklarından oluşurlar. Şablon bileşenleri bir web sayfasının meydana gelebilmesi için birleştirilirken sayfanın hem görüntüsünü hem de sayfada yer alacak uygulamaları belirlemiş olurlar. Bir web sitesi sadece uygulamalardan ibaret olmadığı gibi sadece görünümünden de ibaret değildir. DyNA’nın XML, PHP ve MySQL altyapısı, şablon bileşenlerinin ayrıştırma ve birleştirme işlemlerini mümkün kılar.

DyNA’da şablon bileşenleri aşağıdaki görevleri gerçekleştirmek amacıyla kullanılmaktadırlar:

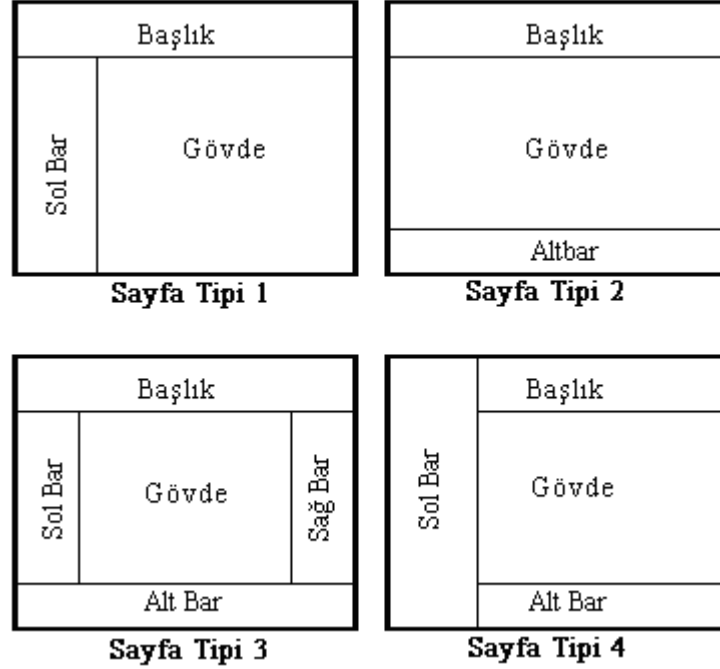
- Sayfaların çalışma biçimlerini ve sayfalarda yer alan uygulamaları belirleme,
- Sayfaların görünümünün belirlenmesi, arayüz nesnelerinin oluşturulması,
- Site içerisinde yer alan seyir sistemi, o anda bulunan sayfanın gösterimi, site haritası gibi otomatik olarak oluşturulan yapıların gerçekleştirilmesi,
- RSS, XML, RTF, PDF, WAP gibi farklı biçimlerde çıktıların üretilmesi,
- Kullanıcılarla ilgili giriş, çıkış, takip, güvenlik işlemleri,
- Yetkilendirme sistemi, yetkiler doğrultusunda seyir, görüntüleme ve diğer web sayfası uygulamalarının gerçekleştirilmesi,
- Aynı veya farklı görünüm seçenekleriyle çoklu dil desteğinin sağlanması,

- Modüler programlamayı destekleme,
- Yeniden kullanılabilirlik,
- Proje bazlı değiştirilebilir, esnek yapıların oluşturulması.

#### 4.4. Sayfa Tipi ve Görünüm Tipi Tanımları

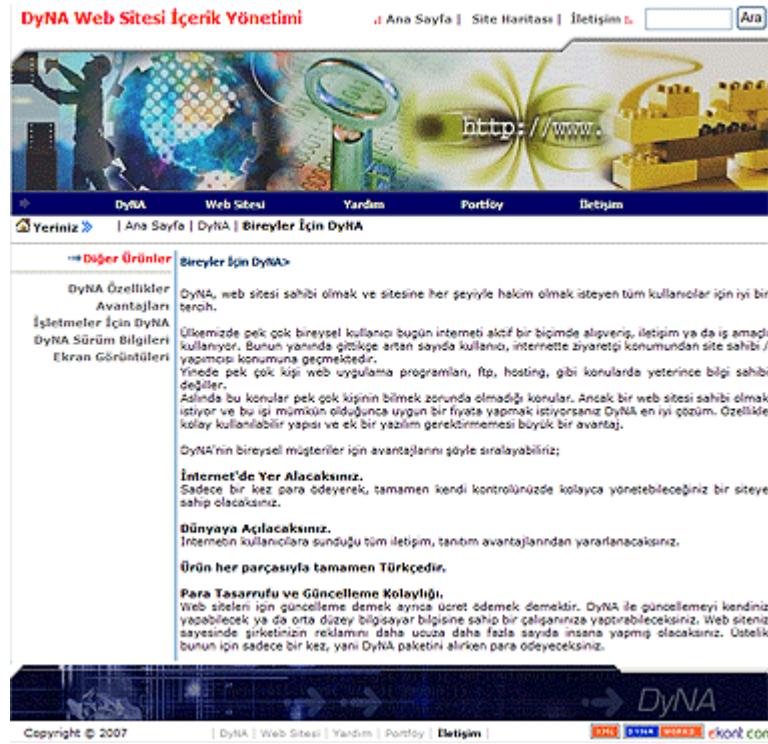
DyNA yapısı itibarı ile çok çeşitli görünüm stillerini desteklemektedir. Geliştiriciler kendi görünüm stillerini de sınırsız olarak gerçekleştirebilirler. Bir web sitesi içerisinde kullanılacak her bir farklı yerleşim stili “sayfa tipi” olarak tanımlanmaktadır. Site yayına konduktan sonra da yeni sayfa tipleri türetilip eklenebilmektedir.

Sayfa tipleri web sayfasının görünümünü, yerleşim düzenini ve iş mantığını belirler. Şekil 4.2’ de görüldüğü üzere sayfa tipi, hangi şablon-bileşenin sayfanın neresinde yer alacağını gösterir.

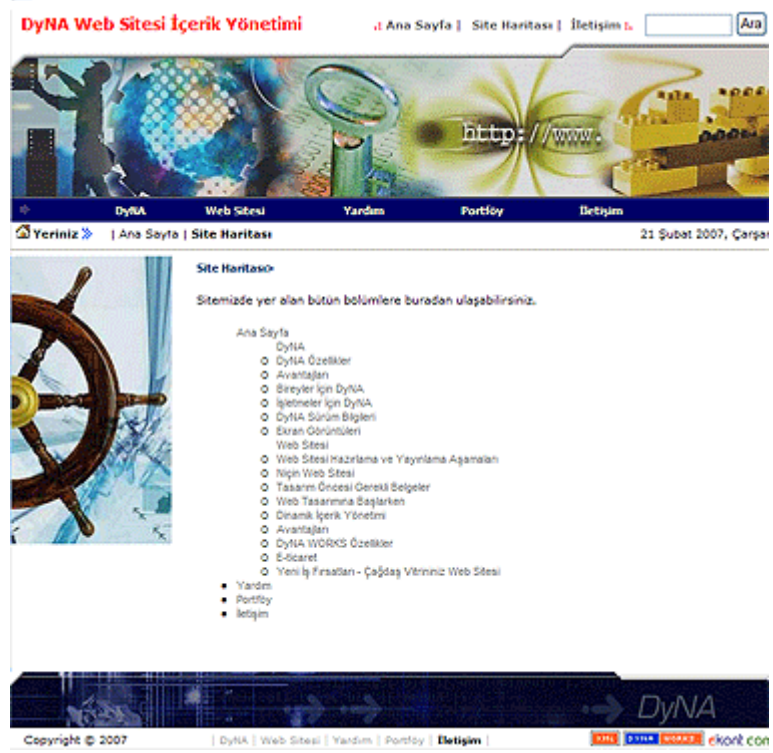


Şekil 4.2: Sayfa tipleri

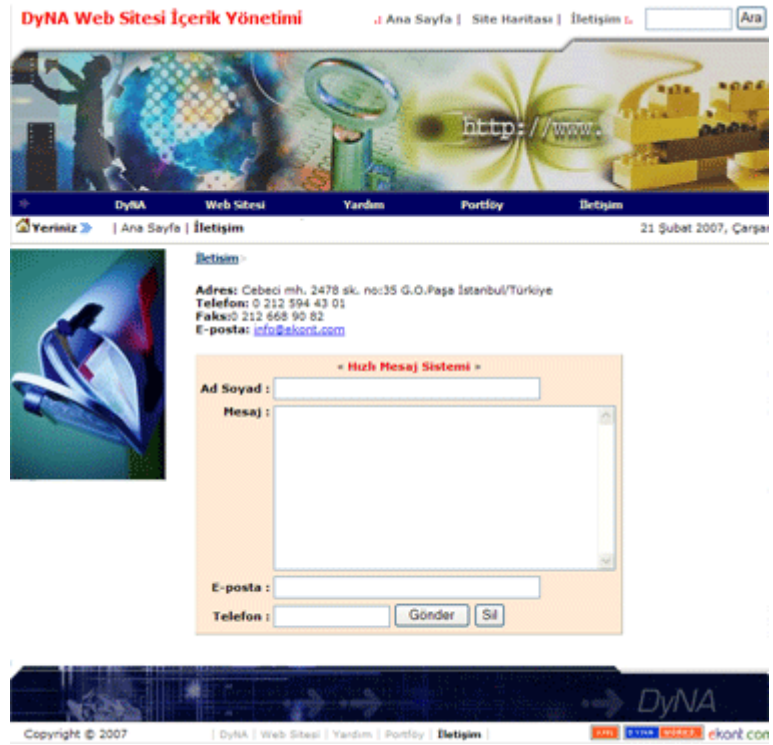
Şekil 4.3, Şekil 4.4 ve Şekil 4.5’de sırası ile site haritası, iletişim ve makale sayfaları yapısal olarak birbirlerine benzemekle birlikte tamamen farklı iş mantıklarına ve parçalarına sahiptirler. Bu nedenle, bütün bu sayfalar ve sayfa tipleri ayrıca tanımlanmalıdırlar. Her sayfa tipi için aynı ada sahip farklı içerikte şablon-bileşenler tanımlanır ve hazırlanır. Örneğin makale sayfasının gövdesini meydana getiren “govde” adlı bileşenle site haritası sayfasının gövdesini meydana getiren bileşen aynı ada sahip olmasına rağmen içerikleri farklıdır ve birbirinden bağımsız alt şablon-bileşenlere sahip olabilirler. Aynı durum iletişim sayfası için de geçerlidir ve burada da ziyaretçi tarafından doldurulan bir formdan elde edilen bilgiler yönetim panelinden belirlenen bir e-posta adresine gönderilmektedir.



Şekil 4.3: Makale sayfası (Sayfa tipi 1)



Şekil 4.4: Site haritası sayfası (Sayfa tipi 2)



Şekil 4.5: İletişim sayfası (Sayfa tipi 3)

DyNA’da kullanılan görünüm tipi parametresi, bir web sayfasının; site dili, yetkili kullanıcı görünümü, metin görünümü, platforma bağlı görünüm gibi genel görünüm ve stil özellikleri ile ilişkilidir (Tablo 4.1).

Şablon bileşenleri genellikle yan bar, başlık, alt bar gibi sayfa parçalarıdır. Bununla birlikte bütün şablon bileşenleri görsel ve görünür olmak zorunda değildirler. İş mantığı da şablon-bileşenler kullanılarak ayrıştırılıp yönetilebilir.

Tablo 4.1: Örnek görünüm tipleri

Görünüm Tipi Değeri	Görünüm Tipi
1	İngilizce (ziyaretçiler için)
2	Türkçe (ziyaretçiler için)
3	Yetkili kullanıcılar
4	Site Yöneticileri
5	Cep Telefonları ve WML
...	...

DyNA web sitesi içerik yönetimi platformu doğrudan sayfa yönetimi yerine, sayfa tipi yönetimine dayanmaktadır. Bu yüzden geliştirici aynı sayfa tipinde yüzlerce sayfada aynı anda değişiklik yapmak isterse sadece ilgili şablon bileşenini değiştirmesi yeterlidir.

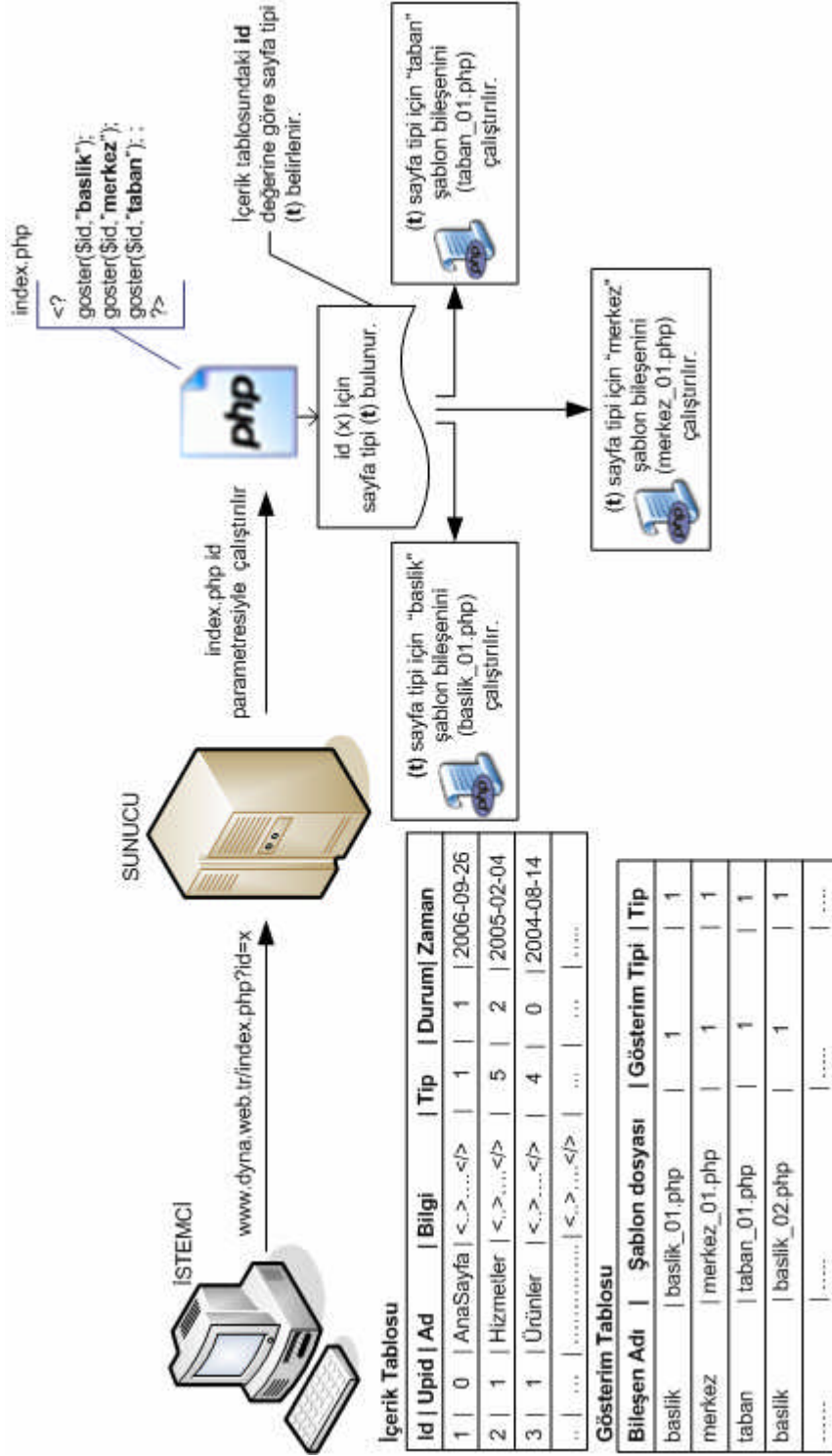
Sayfa tipleri, görünüm tipleri ve şablon-bileşenleri arasındaki ilişkiler DyNA veritabanının gösterim adlı tablosunda tutulmaktadır.

#### 4.5. Sayfa Gösterimi ve Şablon-Bileşen İş Akışı

Kullanıcı tarafından bir sayfa görüntülenme isteği gelmesi halinde aşağıdaki işlem adımları gerçekleşir (Şekil 4.6):

- 1) Kullanıcı, id veya görünüm tipi parametresi ile birlikte bir sayfa isteğinde bulunur (örneğin [www.dyna.web.tr/index.php?id=45&gorunum=2](http://www.dyna.web.tr/index.php?id=45&gorunum=2))

- 2) Bileşen yöneticisi, verilen id değerine karşılık gelen kaydın tip alanındaki değeri okuyarak sayfa tipini belirler.
- 3) Tip ve görünüm tipi değerlerine göre bileşen yöneticisi gösterim adlı tabloda belirtilmiş olan dinamik şablon bileşeni dosyasını seçer.
- 4) Seçilen şablon-bileşen sırayla çalıştırılır. Çalıştırma PHP yorumlayıcısı tarafından gerçekleştirilir. Bu işlemler için ön işleme yapmaya gerek bulunmamaktadır.



Şekil 4.6: DyNA çevrimiçi sayfa üretimi

#### 4.6. Sayfa Parçalama

Sistemin anahtar özelliği karmaşık web sayfalarını basit parçacıklardan meydana getirmesidir. Bu yaklaşım küçük parçacıklarla çalışmayı gerektirdiğinden oldukça verimlidir.

Şekil 4.7’de görüldüğü üzere bir web sitesinin tüm sayfaları beş ana bölüme ayrılmıştır (1-başlık, 2-seyir sistemi, 3-yan bar, 4-gövde, 5-alt bar). Bütün bölümler iş mantığına ve kullanıcı yetkisine göre değiştirilebilmektedir.



Şekil 4.7: Beş bölgeye ayrılmış bir sayfa örneği

Şekil 4.3 ve Şekil 4.5 de aynı bölümlendirme fakat farklı iş mantıkları yer almaktadır. Örneğin makale sayfasının (tip 1, şekil 4.3) yan bar bölümü diğer makalelerin başlıklarına bağlantılar veriyorken iletişim sayfasının (tip 3, şekil 4.5) yan barı e-posta ile ilgili bir resmi göstermektedir. İki sayfa da aynı sayfa yapısında olmasına rağmen (aynı bölümler, farklı şablon bileşenleri) farklı çalışırlar. Eğer bir sayfa tipinin görünümü veya çalışması ile ilgili bir değişiklik yapılacaksa, sadece tip



ve görünüm tipi verilmiş olan ilgili şablon-bileşenin değiştirilmesi yeterli olmaktadır.

www.dyna.web.tr adresindeki örnek web sitemizde kullanıcının gezmek amacıyla çağırabileceği tek bir sayfa (index.php) bulunmaktadır. Diğer tüm sayfalar kullanıcı tarafı istek sonucunda id ve görünüm tipi parametreleri girilerek dinamik olarak üretilmektedir. Sadece id parametresi bir sayfayı bambaşka bir sayfa haline dönüştürmek için yeterlidir. Görünüm tipini belirten başka bir parametre kullanmaya gerek kalmamaktadır.

Gösterim tablosunda parçaların üretiminde hangi şablon-bileşenin kullanılacağını tespit etmek üzere bölüm adı, sayfa tipi, dosya adı ve görünüm tipi olmak üzere dört alan yer alır.

Bölüm adı genellikle “goster” fonksiyonu ile ana sayfa veya şablon-bileşeni içerisinde kullanılmaktadır. Örneğin index.php sayfası aşağıdaki kodları içermektedir:

```
<table><tr><td>
<?
goster ($id,"baslik");
goster ($id,"navigasyon");
?>
</td></tr><tr><td>
<?
goster ($id,"yanbar");
goster ($id,"govde");
?>
</td></tr><tr><td>
<? goster ($id,"altbar"); ?>
</td></tr>
</table>
```

Her sayfa, görünüm stilini ve çalıştırılacak şablon-bileşenini açıklayan bir tip değerine sahiptir. Sayfa tipi olarak “99” değerinin özel bir anlamı vardır. Bu belirtilen bölüm için varsayılan şablon-bileşeni dosyasını gösterir. Eğer bir bölüm için karşılık gelen tip numarası ve bölüm adına ilişkin bir kayıt yoksa varsayılan şablon-bileşen dosyası kullanılır. Örneğin Tablo 4.2’de verilenlere göre; bölüm

adı=gövde, sayfa tipi=4 ve gösterim tipi=1 olan bir kayıt gözükmemektedir dolayısıyla varsayılan şablon-bileşeni dosyası olarak “Govde\_1.php” kullanılacaktır.

Tablo 4.2: Örnek veri ile doldurulmuş gosterim tablosu

Bölüm Adı	Sayfa Tipi	Gösterim Tipi	Şablon-Bileşen Dosyası
Yanbar	1	1	Yanbar_1_1.php
Baslik	1	1	Baslik_1_1.php
Govde	1	1	Govde_1_1.php
Baslik	2	1	Baslik_2_1.php
Altbar	3	2	Altbar_3_2.php
Govde	99	1	Govde_1.php

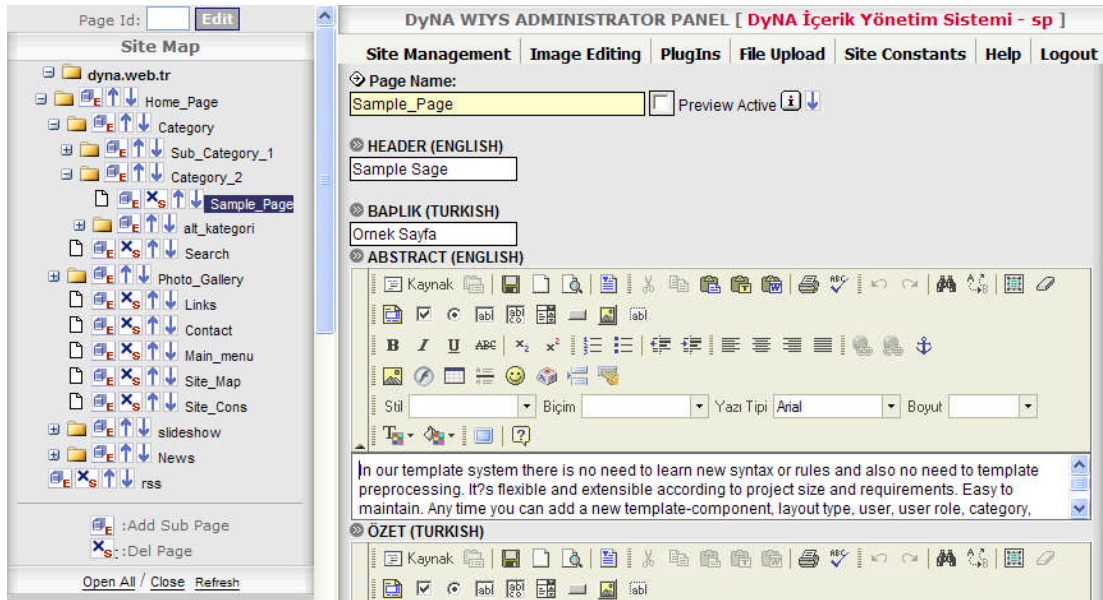
#### 4.7. Düzenleme Şablonları Kullanarak Sayfa Düzenleme

DyNA’da site yönetim paneli (Şekil 4.8), teknik bilgisi olmayan kullanıcıların dahi kolaylıkla güncelleyip içerik ekleyebileceği kullanıcı dostu bir ortam sunmaktadır. Bunun nedeni mevcut site tasarımını bozmadan web sitesine bütünleşik bir şekilde içerik girişinin ve yönetiminin hedeflenmiş olmasıdır. Bu arayüze erişmek için kullanıcı tarafında belli bir işletim sistemi kullanma zorunluluğu yoktur. Yönetim paneline erişim dinamik ip kontrol mekanizması, kullanıcı adı-şifre eşleştirmesi ve sürekli değişen anahtar-değer sistemleriyle kontrol altına alınmıştır.

Yönetim paneline erişildikten sonra kullanıcıya ekranın sol tarafında sitede yer alan tüm sayfalar bir ağaç hiyerarşisi şeklinde sergilenir. Kullanıcı kendi rolü ve yetkileri doğrultusunda web sayfalarını, diğer kullanıcı hesaplarını yönetebilir. Eğer yeni bir sayfa eklenmek istenirse yeni sayfanın ekleneceği sayfa adının solundaki ekle düğmesine tıklaması yeterlidir. Sonrasında, yeni eklenecek sayfanın tipi liste kutusundan seçilir ve yeni sayfaya bir isim verilir. Bir sayfa silinmek istenirse yine sayfa adının solundaki “sil” düğmesi tıklanır. Eğer altında başka bir sayfa bulunmuyorsa silme işlemi gerçekleşir, aksi takdirde silme onaylanmaz. Eğer sayfanın solunda izin sembolü varsa bu ona bağlı sayfalar olduğunu gösterir ve “sil” düğmesi görüntülenmez. Kullanıcı aynı zamanda aşağı ve yukarı yön ok simgelerini kullanarak sayfaların yerlerini ve sıralamalarını değiştirebilir.

Kullanıcı, site haritası üzerinde herhangi bir sayfa adının üzerine tıkladığında sayfa ile ilgili tüm içerik düzenlemek üzere sağ tarafa yüklenir. Eğer kullanıcının yetkisi düzenlemek için yeterliyse yüklenen tüm içeriği değiştirebilir. DyNA, HTML dili hakkında herhangi bir bilgisi olmayan kullanıcıların bile rahatlıkla içerik girip düzenleyebileceği Ne Görürsen Onu Alırsın (What You See Is What You Get - WYSIWYG) tarzındaki çevirim içi çalışan HTML editörlerini desteklemektedir.

Bunun yanı sıra tecrübesiz kullanıcılar için bir başka yazılım kullanmaksızın dosya gönderimi, resim dosyaları üzerinde yeniden boyutlandırma, kesme ve döndürme gibi temel işlemleri de gerçekleştirmek mümkündür.



Şekil 4.8: DyNA yönetim paneli ekran görüntüsü

DyNA sayfa düzenleme işlemleri için de bir şablon sistemi kullanmaktadır. Her bir sayfa tipi, farklı editör şablonlar kullanılarak, ayrı ayrı düzenlenebilir. Sayfa görünümleri ve tipleri tasarlanırken geliştiriciler dinamik ve statik bölgeler belirleyebilirler. Bir sayfadaki dinamik bölgeleri yönetmek için DyNA, geliştiriciler tarafından kolaylıkla değiştirilebilen, şablon dosyalarını kullanmaktadır. Bu sebeple sayfa düzenlemek için tek bir sayfa (duzenle.php) varken sayfa tipi kadar düzenleme şablonu dosyası bulunmaktadır. Böylece tek çatı altında tüm sayfa tipleri birbirinden bağımsız olarak düzenlenebilmektedir.

Düzenleme şablonları metin alanı, metin kutusu, seçenek düğmesi, liste kutusu, onay kutusu gibi html form elemanlarını içerirler. Böylece kullanıcı seçili bölgedeki içeriği yönetebilmektedir.

Makale sayfası (Tip 1) için örnek düzenleme şablonu kodu aşağıdadır.

```
Yazar Adı <input type="text" name="yazar_adi" value="
<? Echo tag_oku($xml," yazar_adi ");?>">
Başlık (in English)<input type="text" name="eng_baslik" value="
<? Echo tag_oku ($xml," eng_baslik ");?>">
Makale (in English)<textarea name="eng_makale" rows="10" cols="40">
<? Echo tag_oku ($xml," eng_makale");?></textarea>
Makale (in Turkish)<textarea name="tr_makale" rows="10" cols="40">
<? Echo tag_oku ($xml,"tr_makale");?></textarea>
```

Kullanıcı içeriği güncelledikten sonra editör şablonun içindeki tüm form elemanları toplanır ve aşağıdaki gibi bir XML yapısına dönüştürülür.

```
<DYNA_YAZAR_ADI>George Smith</DYNA_YAZAR_ADI>
<DYNA_ENG_BASLIK>About CMS</DYNA_ENG_BASLIK>
<DYNA_ENG_MAKALE>
CMS is the most important management tool for large size web sites.
</DYNA_ENG_MAKALE>
<DYNA_TR_MAKALE>
IYSler büyük çaplı web siteleri için vazgeçilmez bir yönetim
aracıdır.
</DYNA_TR_MAKALE>
```

İçeriğe ilişkin tüm bilgi içerik tablosundaki bilgi alanında saklanır. XML yapısı editör şablonu değiştirilerek kolaylıkla değiştirilebilir. Her sayfa tipinin tekil bir XML yapısı vardır.

DyNA rol tabanlı içerik gösterim özelliğini de desteklemektedir. Örneğin sadece yetkili yöneticinin görebildiği gizli alanlar oluşturulabilir.

PHP dili dizilerle ilgili gelişmiş fonksiyonlara sahiptir. Bu güçlü yapıyı ve fonksiyonları etkili bir şekilde kullanabilmek için DyNA'nın kendi fonksiyonları bu dizi tipinde dönüş değerleri üretirler. Böylece geliştiriciler PHP'nin kendi yazım kurallarını ve döngü yapılarını kullanabilirler. Bu da yeniden bir dil ve yazım kuralı öğrenme zorunluluğunu ortadan kaldırmaktadır.

Aşağıda *id* adlı değişken ile belirtilmiş sayfanın kendisine bağlı alt sayfaların (çocuklar) “başlık” etiketlerini liste halinde ekrana yazan bir kod gösterilmiştir. DyNA içerisinde yer alan tüm fonksiyonlar ve hata kodları Türkçe olarak yazılmıştır.

```
$cocuklar_dizisi=cocuklar($id);  
For ($i=0;count($cocuklar_dizisi);$i++)  
Echo tag_oku(xml_al(cocuklar_dizisi [$i]),"baslik");
```

#### 4.8. İçerik Girişi ve Düzenleme Şablonları

İçerik girişi ve düzenleme şablonları, tüm gösterim tiplerine ilişkin verileri tek bir veritabanı tablosunun kayıt satırı olarak XML biçiminde tutulmasını ve tüm işlemlerin bu kayıt üzerinde yapılmasını sağlar.

Web sitesinin tasarlanması aşamasında her tip için içeriği değiştirilecek bölgeler belirlenir. Bu aşamadan sonra değişmeyen alanlar ortaya çıkacağından en genel tip de (varsayılan tip) tespit edilmiş olur. Belirlenen her bölge bir etiket ile ifade edilir. Örneğin sayfanın gövde bölümünde yer alacak metnin Türkçe başlığı için “tr\_baslik” isimlendirmesi yapılabilir. Gövde metninin kendisi için “tr\_govde” adı verilebilir.

Bir sayfa tipini meydana getiren her bölge aynı zamanda bütün gösterim tiplerini de kapsayacak şekilde belirlenmiş ve isimlendirilmiş olmalıdır. Böylece o tipe ait düzenleme sayfasında tüm gösterim tipleri için kullanılacak etiketlere ayrı ayrı içerik girişi yapılabilir.

Düzenleme sayfası aracılığı ile yapılan içerik girişleri bir XML yapısına uyacak şekilde biçimlendirilir ve içerik tablosuna bir kayıt olarak eklenir.

Örneğimizde altbar, ustbar, ve navigasyon bölümleri tüm tiplerde aynı görünüme sahiptir. Sadece dil değişimi olduğunda (gösterim tipi değiştiğinde) farklılaşırlar. Bu değişimi gerçekleştirebilmek için bu bölgeleri düzenleyen bileşen-şablon dosyalarından gösterim tipi sayısı kadar hazırlamak yeterlidir. Gösterim tipi bilgisi siteye girildiği andan itibaren gosterim adlı bir oturum (session) değişkeninde tutulmaktadır

Web sayfasının kullanıcı tarafında görüntülenmesi için, görünüm katmanında yer alan bileşen-şablonlar, içeriği elde edebilmek için id numarası kullanarak veritabanı kaydına konumlanırlar ve o sayfaya ait tüm içeriği `xml_al()` fonksiyonuyla bir değişkene aktarılırlar. Sonrasında hangi etiketin içeriği görüntülenmek isteniyorsa `tag_oku()` fonksiyonuyla etikette belirtilen içeriğe erişilmiş olur.

```
$xml=xml_al($id);  
$baslik=tag_oku($xml,"tr_baslik");  
Echo $baslik;
```

Veya aynı sonucu elde etmek için kod tek satır olarak da çalıştırılabilir.

```
echo tag_oku(xml_al($id),"tr_baslik");
```

## 5. DyNA KURULUM VE ÖZELLİKLERİ

DyNA, PHP dilinde, endüstri standardı olarak kabul edilen XML yapısı kullanılarak yazılmıştır. DyNA, UNIX, LINUX ve Windows işletim sistemlerinin farklı sürümlerinde çalışabilecek bir yapıya sahiptir.

DyNA projesinde geliştirilebilir ve esnek bir web sitesi içerik yönetim sistemini oluşturabilmek için PHP, MySQL ve APACHE kullanımı tercih edilmiştir. PHP esnek kodlama imkanı sağlarken MySQL veritabanı güvenilirlik, kullanım kolaylığı ve hızlı çalışması sebebiyle seçilmiştir.

HTML dilinin yetersiz kaldığı pek çok konuda yapısal eksiklikleri tamamlamak, esnek veri yapıları kullanarak kayıt ve karşılıklı veri alışverişi işlemlerini yürütmek amacıyla XML dilinin kullanımı tercih edilmiştir. HTML verinin biçimi ile ilgilenirken, XML yapı ile ilişkili bilgiyi tutmaktadır. XML'in gücü web sayfalarının işlevselliğini artırarak, tam ve net olarak verinin içeriğini ve anlamını tanımlayabilmesinden gelmektedir [50-53].

DyNA kullanıcılar ve geliştiriciler için karmaşık web sayfalarını yönetmek için dinamik şablon-bileşenler kullanımıyla kolay metotlar sunmaktadır [54]. Bileşik görünüm, içeriğin sayfa yapısından ayrılması, sayfa görünümünün parçalar halinde ele alınması gibi pek çok özelliğe sahiptir. Aynı anda birden fazla kullanıcının veri girişine ve yetkileri doğrultusunda işlemler yapmasına olanak tanır. Dahası rol tabanlı kullanıcı sistemi sayesinde farklı yetki seviyelerdeki farklı kullanıcıları destekler. XML, DyNA'nın çekirdeğini oluşturmaktadır. Bu yaklaşım sayesinde web sitesi içerisinde veya farklı web siteleri arasında veri transferi ve sorgulaması yapılabilmektedir. Aynı zamanda çeşitli platformlar için çeşitli biçimlerde yayın yapılması sağlanmıştır [55].

Web tabanlı bir uygulama olarak Apache ve IIS vb. bir web sunucusu uygulaması üzerinde çalışır. Verileri kaydetmek için MySQL veritabanını kullanır. İstenirse sistem gereksinimleri doğrultusunda Microsoft SQL Server veya Oracle veritabanları da kullanılabilir.

İçerik bilgisi XML biçiminde MySQL veritabanının “icerik” tablosunun “bilgi” adlı alanında tutulmaktadır. Görünüm ile ilgili bilgiler ve işlem kodları PHP dosyaları şeklinde tutulmaktadır. Web sunucudan bir sayfa istendiğinde içerik bilgisi veritabanından alınır ilgili şablon kullanılarak çalıştırılır. Bu işlemin sonucunda elde edilen HTML biçimindeki sonuç verisi de talepte bulunana (bu genellikle ziyaretçinin kullandığı İnternet tarayıcısıdır.) gönderilir.

Şablon dosyaları içerlerinde XML bilgisiyle işlemsel uygulamalar yapılabilmesi için PHP dilinde hazırlanmışlardır. Web sitesinin genel görünümünü kontrol etmek amacıyla XSL ve CSS dosyaları da kullanılabilir. DyNA ile hazırlanmış herhangi iki sitenin görsel olarak herhangi bir ortak yönünün veya benzerliğinin olması gerekmemektedir. Site kurulumu için önceden belirlenmiş bir şablon kullanılabileceği gibi ihtiyaç ve istekler doğrultusunda eklemeler yapmak mümkündür.

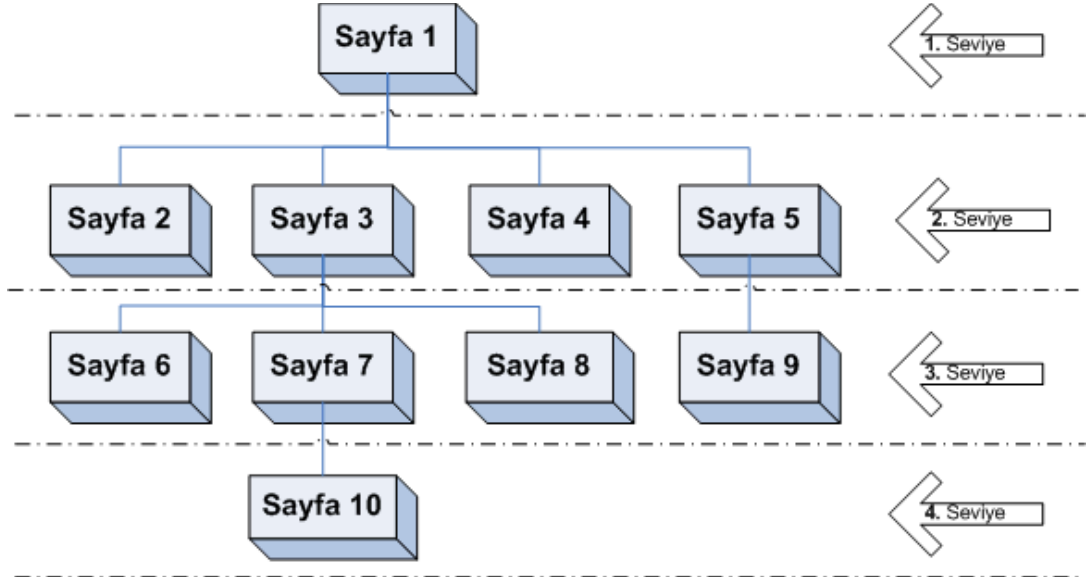
### **5.1. DyNA’da Sayfa Hiyerarşisi**

DyNA ile yönetilen web sayfaları arasında bir hiyerarşi mevcuttur. Ana sayfa diğer tüm sayfaların üstünde yer alır. Diğer tüm sayfalar ana sayfaya doğrudan yada dolaylı olarak bağlantılıdır. Hiçbirisi bağımsız olamaz. Tüm sayfalar ağaç hiyerarşisi şeklinde gösterilebilir. Böylece farklı sayfalar arasındaki bağlantılar kolayca ifade edilebilir.

DyNA ile yönetilen web sitelerinde; aynı seviyede ve aynı sayfaya bağlı olan sayfalara “kardeş sayfalar”, bir üst seviyede bağlı olunan sayfaya “ata” veya “ebeveyn”, bir alt seviyede bağlı olunan sayfa yada sayfalara da “çocuk” adı verilir.



Şekil 5.1’de gösterilen örnekte, sayfa 7 ele alındığında, sayfa 6 ve sayfa 8 kardeş, sayfa 10 çocuk, sayfa 3 birinci dereceden ata, sayfa 1 de ikinci dereceden ata olarak adlandırılır.



Şekil 5.1: DyNA sayfa hiyerarşisi

Sayfalar arasında bilgi alışverişi ve erişim için kardeşler(), upid() ve çocuklar() fonksiyonları kullanılır.

Kardesler(id) fonksiyonu, id parametresi girilen sayfayla aynı seviyedeki ve aynı ebeveyne bağlı olan sayfaların id değerlerini bir dizi şeklinde geri döndürür.

Upid(id,seviye) fonksiyonu, id parametresi girilen sayfanın bağlı olduğu ebeveyn sayfasının id değerini geri döndürür. Seviye değeri verildiğinde belirtilen seviyedeki ebeveynin id değeri döndürülür. Seviye parametresinin varsayılan değeri 1’dir. Örneğin upid fonksiyonuna id değeri olarak sayfa 10’un id değeri, seviye olarak “1” değeri verildiğinde sayfa 7’nin, “2” verildiğinde sayfa 3’ün, “3” değeri verildiğinde sayfa 1’in id değeri döndürülür.

Cocuklar(id) fonksiyonu, id parametresi girilen sayfaya bağlı olan bir alt seviyedeki sayfaların id değerlerini sıralı bir şekilde dizi türünde geri döndürür. Örneğin,

cocuklar fonksiyonuna id değeri olarak sayfa 3 ün id değeri girilirse, dönüş değeri olarak sırasıyla sayfa 6, sayfa 7 ve sayfa 8'in id değerleri döndürülür.

DyNA'da yapısal bakımdan herhangi bir seviye ve sayfa sınırlaması yoktur.

## **5.2. DyNA'da Veritabanı Yapılandırması**

DyNA kurulmadan önce hazırlanacak sitede yayınlanması planlanan tüm sayfa tipleri, dil sayısı ve yetkilendirmeye bağlı görünüm sayısı tespit edilir. Web sitesinde yayınlanan bir sayfa, içerik bileşenlerinin daha önceden hazırlanmış şablonlara göre bir araya getirilmiş halidir.

DyNA bütün web sitesi içeriğini site geliştiricilerinin belirleyebileceği XML etiketleri halinde "icerik" adlı bir MySQL tablosunda saklamaktadır. icerik tablosu sekiz alana sahiptir. Bunlar id, upid, ad, bilgi, seviye, tip, durum, sıra, zaman ve param şeklindedir.

Bir web sayfasının barındırdığı tüm içerik XML biçiminde bilgi alanında tutulur ve tekil, otomatik artan şeklinde belirtilmiş bir "id" parametresiyle temsil edilir. DyNA'da her bir sayfanın tekil bir "id" değeri vardır

Upid parametresi bağlı olunan sayfanın id değerini tutar. Web sitesinin planlama aşamasında geliştiriciler, sayfa tipleri ve sayfa tipleri arasındaki ilişkileri belirlerler. Hangi sayfa tiplerinin hangi sayfa tipleri altına yerleştirilebileceğini belirleyen kuralları tespit ederler. Örneğin "kategori" sayfaları dal olarak tanımlanırlarsa "makale" sayfaları bu dalların yaprakları olurlar. "Kategori" sayfasının altına "makale" eklenebilir fakat "makale" sayfasının altına "kategori" eklenemez. Kullanıcı yeni bir sayfa eklerken oluşturulacak sayfanın tipini sayfa tipleri listesinden doğru olarak seçmelidir. Sayfa tipleri listesi "tip izinleri" adlı MySQL tablosunda tutulmaktadır ve geliştiricilerin tip ekleme izinleri ile ilgili kurallarını saklar.

Web sitesinde istemci ziyaretçi tarafından çağrılan “**index.php**” adlı tek bir sayfa vardır. Web sitesine ilişkin tüm içeriğin görüntülenmesi index.php dosyasının id ve gosterim\_tipi parametresiyle çalıştırılması suretiyle gerçekleştirilir.

Sitenin kaç dille yayınlanacağı belirlendikten sonra gösterim tipinin alacağı değerler belirlenir.

Tablo 5.1: Gösterim tipi adları ve değerleri

<b>Gösterim Tipi</b>	<b>Değer</b>
Türkçe	1
İngilizce	2
Almanca	3
Fransızca	4
...	...

Birbirinden yapısal olarak farklı olacak sayfalar tespit edilir ve bunlara isim verilir. Tip adları ve karşılık gelen tip numaraları belirlenir ve tipler tablosunda tutulurlar (Tablo 5.2).

Tablo 5.2: Örnek tipler tablosu içeriği

<b>Tip Adı</b>	<b>Tip No</b>
Temel	1
Kök	2
Makale	3
Kategori	4
Site Sabitleri	5
Haberler	6
Foto Galeri	7
Site Haritası	8
Yorumlar	9
...	...

Hangi sayfa tipinin hangi sayfa tipi altına eklenebileceğini göstermek ve bu konuda kısıtlamalar yapmak üzere sayfa tipleri içerisindeki ebeveyn-çocuk ilişkisini düzenleyen tip\_ izinleri tablosu oluşturulur (Tablo 5.3). Bu tabloda tip numaraları tutulmaktadır.

Tablo 5.3: Örnek tip\_ izinleri tablosu içeriği

Ata	Çocuk
1	2
1	3
4	3
1	8
...	...

Örneğin Tablo 5.3'ün 3. satırında “Kategori” sayfa tipinin altına “Makale Sayfa” tipinde sayfalar oluşturulabileceğine dair bir izin tanımlaması yapılmıştır.

Gösterim tipine göre, oluşturulacak içerik bileşeninin hangi şablon dosyasına göre oluşturulacağına dair bilgiler gösterim adlı tabloda tutulmaktadır (Tablo 5.4).

Sayfa tipleri web sayfasının görünümünü, yerleşim düzenini ve iş mantığını belirler. Şekil 4.2' de görüldüğü üzere sayfa tipi, hangi şablon-bileşeninin sayfanın neresinde yer alacağını gösterir.

Tip numarası olarak “99” değeri verildiğinde varsayılan tip gösterilmiş olur. Örneğin Tablo 5.4'ün 1. satırdaki kayıt “govde” içerik bileşeni Türkçe dilinde oluşturulduğunda aksi belirtilmediği sürece tüm sayfa tipleri için “D\_govde.php” şablon dosyasının kullanılacağını göstermektedir.

Tablo 5.4'ün 7. satırında ise yine “gövde” içerik bileşeni İngilizce dilinde, “Makale” sayfa tipi için oluşturulduğunda şablon dosyası olarak “D\_govde\_3\_en.php” dosyasının kullanılacağını göstermektedir.

Tablo 5.4: Örnek gösterim tablosu içeriği

ID	Gösterim Adı	Tip Numarası	Gösterim Tipi	Şablon Dosyası
1	Govde	99	1	D_govde.php
2	Govde	99	2	D_govde_en.php
3	Govde	3	2	D_govde_3_en.php
4	Govde	3	1	D_govde_3.php
5	Govde	1	2	D_govde_en.php
6	Govde	3	1	D_govde_3.php
7	Govde	3	2	D_govde_3_en.php
8	UstBar	1	1	D_altbar.php
9	Altbar	99	1	D_altbar.php
10	Altbar	3	1	D_altbar_makale.php
11	Altbar	3	2	D_altbar_makale_en.php
12	Ana_menu	1	1	D_altbar.php
13	SolBar	3	1	D_solbar_makale.php
14	SolBar	99	1	D_solbar.php
15	Baslik	99	1	D_baslik.php
16	Yorum	99	1	D_yorum.php
17	Navigasyon	99	1	D_navigasyon.php
18	Duzenle	99	1	D_tip_99.php
19	Duzenle	3	1	D_tip_3.php
20	Duzenle	4	1	D_tip_4.php
21	Duzenle	5	1	D_tip_5.php
22	...	...	...	...

gosterim tablosunu oluşturan MySQL sorgusu aşağıdaki gibidir:

```
CREATE TABLE `gosterim` ( `id` int(3) unsigned NOT NULL
auto_increment, `gosterim_adi` char(40) NOT NULL default '',
`tip_no` tinyint(3) unsigned NOT NULL default '0', `dosya_adi`
char(50) NOT NULL default '', `gosterim_tipi` tinyint(3) unsigned
NOT NULL default '0', PRIMARY KEY
(`id`,`gosterim_tipi`,`tip_no`,`gosterim_adi`)) ENGINE=MyISAM
DEFAULT CHARSET=utf8 COLLATE=utf8_turkish_ci;
```

DyNA, farklı yetki seviyeleri tanımlanmasına olanak veren geniş bir rol ve kullanıcı tanımlama yeteneğine sahiptir. Sisteme tanımlı kullanıcıların her biri, rollerine göre farklı yetki seviyelerinde operasyonlar gerçekleştirebilirler. Rollerin yetki tanımlamaları roller tablosunda tutulmaktadır.

Tablo 5.5: Örnek roller tablosu içeriği

Rol Adı	YETKİLER						
	Okuma	Yazma	Değiştirme	Silme	Ekleme	Görme	Super
Admin	5	5	3	3	4	5	0
Editör	1	1	1	0	2	2	0
Yetkili	9	9	9	9	9	9	1
...	...	...	...	...	...	...	...

roller tablosunu oluşturan MySQL sorgusu aşağıdaki gibidir:

```
CREATE TABLE `roller` (
  `rol_id` tinyint(3) unsigned NOT NULL
  auto_increment,
  `rol_adi` char(30) NOT NULL default '',
  `okuma`
  tinyint(1) unsigned zerofill NOT NULL default '0',
  `yazma`
  tinyint(1) unsigned zerofill NOT NULL default '0',
  `degistirme`
  tinyint(1) unsigned zerofill NOT NULL default '0',
  `silme`
  tinyint(1) unsigned zerofill NOT NULL default '0',
  `ekleme`
  tinyint(1) unsigned zerofill NOT NULL default '0',
  `gorme`
  tinyint(1) unsigned zerofill NOT NULL default '0',
  `super`
  tinyint(1) unsigned zerofill NOT NULL default '0',
  PRIMARY KEY
  (`rol_id`)) ENGINE=MyISAM DEFAULT CHARSET=utf8
  COLLATE=utf8_turkish_ci;
```

Kullanıcı adları, rolleri, kullanıcı şifreleri ve e-posta adresleri gibi bilgiler “kullanıcılar” adlı tabloda tutulmaktadır. Kullanıcı şifreleri veritabanında geri dönüşümsüz şifreleme algoritmasıyla (MD5) şifrelenerek tutulmakta, böylece şifre geri dönüşümü engellenmektedir. Kullanıcı ile ilgili düzenlemeler yapıldığında, yapılan işlemle ilgili bilgilendirmeler, e-posta alanında belirtilmiş olan adrese mesaj şeklinde gerçekleştirilmektedir.

Tablo 5.6: Örnek kullanıcılar tablosu içeriği

Kullanıcı Adı	Rol	Şifre	E-posta
Ahmet	Yetkili	ee7e946fea14b7fddbe3260a592c30e2	aa@aa.com
Cem	Editör	1952a01898073d1e561b9b4f2e42cbd7	cem@hotmail.com
Pınar	Admin	f458ba2b79c362e5d053de43a8143df7	pkocak@yaho.com
...	...	...	...

kullanıcılar tablosunu oluşturan MySQL sorgusu aşağıdaki gibidir:

```
CREATE TABLE `kullanici` (
  `id` tinyint(3) unsigned NOT NULL
  auto_increment,
  `rol_adi` char(30) NOT NULL default '',
  `kullanici_adi` char(30) NOT NULL default '',
  `sifre` char(32) NOT NULL default '',
  `eposta` char(60) NOT NULL default
  'deneme@deneme.com',
  PRIMARY KEY (`id`)) ENGINE=MyISAM DEFAULT
  CHARSET=utf8 COLLATE=utf8_turkish_ci;
```

Web sitesinde yayınlanacak metinler içerik tablosunda yer alan bilgi adlı alanda XML biçiminde tutulmaktadır. id değeri tüm kayıtlar için tekildir. upid değeri o kaydın ebeveyninin id değerini göstermektedir. Böylece sayfalar arasındaki ebeveyn-çocuk ilişkisi upid ve id alanlarında tutulmuş olmaktadır. Örneğin, Tablo 5.7’ye göre 2. ve 3. satırdaki kayıtlar 1. satırdaki kaydın çocukları olarak ilişkilendirilmiştir. durum alanında sayfanın yayınlanma izni tutulur, değer “1” ise sayfa yayındadır, “0” olduğunda yayından kaldırılmış olur. tip\_no alanı sayfa tipi değerini saklarken, zaman alanında kayıta yapılan güncelleme tarihi bilgisi tutulur.

Tablo 5.7: Örnek içerik tablosu içeriği

<i>Up id</i>	<i>Id</i>	<i>Ad</i>	<i>Bilgi</i>	<i>Durum</i>	<i>Tip No</i>	<i>Zaman</i>
0	1	Ana Sayfa	<DYNA_TR_BASLIK>Ana Sayfa</DYNA_TR_BASLIK> <DYNA_EN_BASLIK>Home Page</DYNA_EN_BASLIK> <DYNA_TR_GOVDE>Ana sayfa gövde metni bu yazı olacak<DYNA_TR_GOVDE> <DYNA_EN_GOVDE>This is a home page<DYNA_EN_GOVDE> <DYNA_DURUM>1</DYNA_DURUM>	1	1	2006-09-27 03:00:25
1	2	İletişim	<DYNA_TR_BASLIK>İletişim</DYNA_TR_BASLIK><DYNA_TR_GOVDE> <p>Site Yöneticisi ile iletişim kurmak için lütfen aşağıdaki formu kullanınız.<b>E-posta:</b> ismail@kirbas.com</DYNA_TR_GOVDE><DYNA_EPOSTA>ismail@kirbas.com</DYNA_EPOSTA><DYNA_IMAJ> <hr size="1"></DYNA_IMAJ>	1	14	2005-01-25 19:05:24
1	3	Yazı	<DYNA_TR_BASLIK>İnternette</DYNA_TR_BASLIK> <DYNA_SON_GUNCELLEME> 27 Mayıs 2005, Cuma </DYNA_SON_GUNCELLEME>	0	3	2005-05-27 15:00:37
...	...	...	...	...	...	...

icerik tablosunu oluşturan MySQL sorgusu aşağıdaki gibidir:

```
CREATE TABLE `icerik` (
  `upid` int(5) unsigned NOT NULL default '0',
  `id` int(5) unsigned NOT NULL auto_increment,
  `seviye` tinyint(3) unsigned NOT NULL default '2',
  `ad` varchar(60) character set latin1 NOT NULL default 'sayfa_adi_verilmemis',
  `bilgi` text collate utf8_turkish_ci,
  `durum` int(10) unsigned NOT NULL default '0',
  `tip_no` tinyint(4) NOT NULL default '0',
  `sira` int(3) unsigned NOT NULL default '0',
  `zaman` datetime NOT NULL default '0000-00-00 00:00:00',
  `param` tinyint(3) unsigned default '1',
  PRIMARY KEY (`id`),
  UNIQUE KEY `ad_indeksi` (`ad`,`id`),
  KEY `hizli` (`upid`,`id`,`tip_no`,`durum`),
  FULLTEXT KEY `bilgi` (`bilgi`))
ENGINE=MyISAM
DEFAULT CHARSET=utf8
COLLATE=utf8_turkish_ci;
```



Sayfaların arşivlenmesi amacıyla içerik tablosunda yapılan değişiklikler düzenlemeler sırasında değişiklikten önceki halleriyle ve tarih bilgisiyle birlikte geçmiş tablosuna aktarılırlar. Böylece sayfaların daha önceki hallerine erişmek daha önceden belirlenmiş geri dönüş sayısında mümkün hale gelir. geçmiş tablosunun alanları aşağıdaki gibidir (Tablo 5.8).

Tablo 5.8: Geçmiş tablosunun alanları

<b>Gid</b>	Her kayıt için tekil bir değerdir.
<b>Id</b>	Bilgi tutulan kaydın içerik tablosundaki <i>id</i> değeridir.
<b>Bilgi</b>	Id değeri ile gösterilen kaydın geçmiş tarihli içeriğini tutar.
<b>Tarih</b>	Kaydı tutulan sayfanın güncelleme zamanını tutar.
<b>Kullanici</b>	Düzenleme ve güncelleme işlemini yapan kullanıcının adını tutar.

geçmiş tablosunu oluşturan MySQL sorgusu aşağıdaki gibidir.

```
CREATE TABLE `gecmis` (
  `gid` bigint(20) unsigned NOT NULL
  auto_increment,
  `id` bigint(20) unsigned NOT NULL default '0',
  `bilgi` text collate utf8_turkish_ci,
  `tarih` datetime default
  NULL,
  `kullanici` varchar(100) character set latin1 default NULL,
  PRIMARY KEY (`gid`),
  KEY `id` (`id`,`tarih`)
  ENGINE=MyISAM
  DEFAULT CHARSET=utf8 COLLATE=utf8_turkish_ci;
```

### 5.3. DyNA’da Kullanılan Şablon-Bileşen Dosya Örnekleri

Aşağıda DyNA ile hazırlanmış ve yönetilen bir web sitesine ait örnek bir ana sayfa ve bu ana sayfayı oluşturan temel bileşenlere ait kodlar yer almaktadır. Örnek ana sayfa (index.php) dosyası toplam on satır içermekte ve içerisine üç adet dinamik şablon bileşeni çağırılmaktadır. DyNA için hazırlanan şablon bileşenlerinde az sayıda ve Türkçe fonksiyonların kullanımı ile site geliştiriciler tarafından kolay okunan ve ihtiyaçlar doğrultusunda kolayca şekillendirilebilen bir yapı ortaya konmaktadır.

### Örnek index.php dosyası içeriği

```
<html>
<body>
<?
If !($id) {$id=1;}
Goster($id,"ustbar");
Goster($id,"govde");
Goster($id,"altbar");
?>
</body>
</html>
```

### Örnek ustbar.php dosyası içeriği

```
<?
$xml=xml_al($id);
echo tag_oku($xml,"tr_baslik");
?>
<br>
```

### Örnek govde.php dosyası içeriği

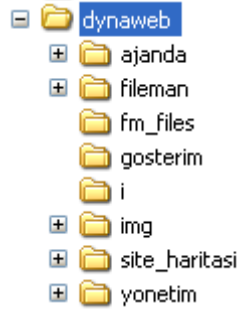
```
<?
$xml=xml_al($id);
echo tag_oku($xml,"tr_govde");
?>
```

### Örnek altbar.php dosyası içeriği

```
<?
$cck[]=cocuklar($id);
for ($i=0;$i<=count($cck);$i++)
{ echo tag_oku(xml_al($cck[$i]),"tr_baslik"); }
?>
```

## 5.4. DyNA’da Dizin Yapılandırması

DyNA içerisinde iskeletin kurulumu sırasında verilen sekiz temel dizin bulunmaktadır (Şekil 5.2).



Şekil 5.2: DyNA temel dizin yapısı

“Ajanda” dizini içerisinde rehber gibi sonradan eklenen modüller tanımlanır ve böylece DyNA’nın temel sistem dosyaları ile birlikte olması önlenmiş olur.

“fileman” ve “fm\_files” dizinleri DyNA içerisinde yer alan dosya yöneticisi tarafından kullanılan dosyaları barındırmaktadır.

DyNA’nın omurgasını oluşturan dinamik şablon-bileşen dosyaları ise “gosterim” dizini içerisinde bulunmaktadır. DyNA’nın yapısında bulunan görseller için “i”, web sitesinde kullanılan diğer tüm görseller ve çoklu ortam dosyaları için de “img” dizini kullanılmaktadır.

“site haritasi” adlı dizin içerisinde yönetim panelinde yer alan arayüzle ilgili çerçeve yapısını oluşturan dosyalar bulunur. Bu dizin içerisinde AJAX (Asynchronous JavaScript and XML) teknolojisini kullanan site haritası ile ilgili “.php” uzantılı dosyalar da yer almaktadır.

Yönetim paneli ile ilgili tüm yönetim işlemlerinin gerçekleştirilmesini sağlayan dosyalar “yonetim” adlı dizin altında yer alır. Yönetim paneline ilişkin özelleştirme ve düzenlemeler buradaki dosyalar üzerinde değişiklik ve ilaveler yapılmak suretiyle gerçekleştirilebilir. Yedekleme işlemleri sonunda meydana getirilen SQL içerikli dosyalar ise “yonetim” dizini altındaki “yedek” dizini içerisinde oluşturulur.

## 5.5. DyNA’da Site Sabitlerinin Tanımlanması

DyNA ile oluşturulmuş bir web sitesine ilişkin tüm sabit tanımlamaları (site adı, adresi, veritabanı adı, veritabanı şifresi, web sitesi dil sayısı vb.) “yonetim” dizini altında yer alan “dyna\_sabitler.php” adlı dosyada tutulmakta ve yönetilmektedir.

Site sabitleri dosyasında tanımlanmış sabitlerin ve değişkenlerin tamamı genel değişken tipinde tanımlanmış olur ve her dosyadan erişilebilir haldedirler. Örnek bir site sabitleri dosyası içeriği aşağıda verilmiştir.

```
<?
$SUNUCU="localhost";
$KULLANICI="root";
$SIFRE="deneme";
$VERITABANI="dynaweb";

$root_url="http://www.dyna.web.tr/";

define ("SITE_ADI","DyNA İçerik Yönetim Sistemi");
define ("SITE_ADRESİ","www.dyna.web.tr");
define ("DİL_ADEDİ","2");
?>
```

## 5.6. DyNA’da Sayfa Oluşturma ve İlişkilendirme Yaklaşımı

DyNA içerisinde web sayfalarının üretimi ve yönetimi belli kurallar ve varsayımlar çerçevesinde gerçekleştirilmektedir. Bunlar:

- Her sayfa birbirinden farklı ve tekil bir kimlik numarasına (id numarası) sahiptir.
- Her sayfa, sayfa üretilmeden önce belirlenmiş, bir sayfa tipine sahiptir.
- Sayfa kimlik numarası “1” değeri dışındaki her sayfa başka bir sayfaya bağlı olmak zorundadır.
- Bir sayfanın dinamik olarak oluşturulabilmesi için en az bir dinamik şablon model dosyasının kullanılması gerekir.
- Sayfalar arasında ayırım yapılabilmesi için sadece sayfa id numarası kullanımı yeterlidir.
- Her gösterim tipinde, her farklı dinamik şablon dosyası bileşeni için, “temel” sayfa tipine ait bir bileşen dosyası eşleştirmesi yapılmış olmalıdır.

- Dinamik olarak oluşturulmaya başlanan sayfanın sayfa tipine ait bileşenlerinden biri tanımlanmamış veya bulunamıyorsa “temel” sayfa tipine ait aynı adlı bileşen dosyası kullanılır.
- Sayfayı oluşturan içerik bileşenlerinin yerleşim düzeni dinamik şablon bileşenlerince belirlenir.
- İçeriği veritabanı tablolarında tutmak için kullanılan XML yapısı ve etiketler, aynı zamanda sayfada yer alan verilerin yapısına ilişkin meta-datayı barındırırlar.
- Bir web sayfasının herhangi bir istemci tarafından çağırılabilmesi için “post” veya “get” metodu kullanılarak o sayfaya ilişkin id değerinin gönderilmesi gerekir. Aksi takdirde id numarası olarak “1” varsayılan değeri kabul edilir.
- Web sitesi ana sayfasının varsayılan id değeri “1” olarak tespit edilmiştir.
- Aynı sayfa için istemci tarafından değiştirilecek parametreler (gösterim tipi vb.) doğrultusunda farklı içerik bileşenleri üretilebilir veya farklı bir sayfa yerleşim düzeni gerçekleştirilebilir.

### **5.7. DyNA ile Kullanıcı Tarafından Talep Edilen Bir Sayfanın Dinamik Olarak Oluşturulması**

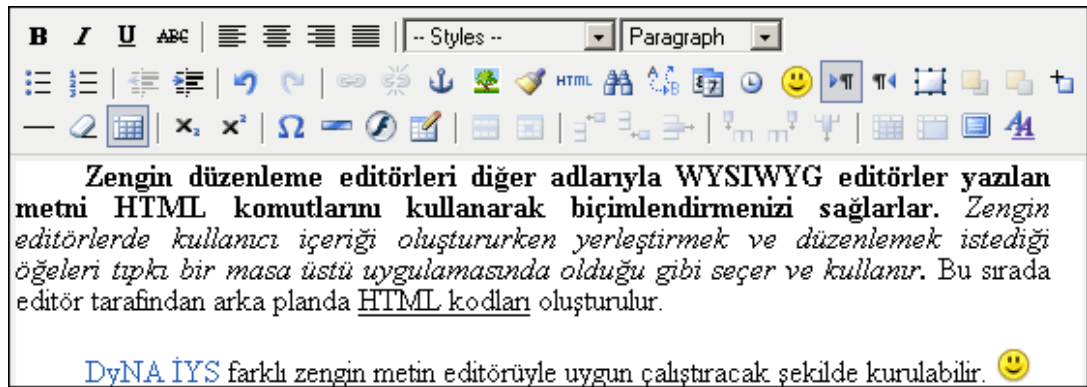
Bir kullanıcı tarafından talep edilen bir sayfanın dinamik olarak oluşturulup kullanıcının tarayıcısına gönderimi için DyNA tarafından aşağıdaki işlem basamakları gerçekleştirilir:

- 1) Ziyaretçi İnternet tarayıcısı tarafından id ve gösterim\_tipi parametresiyle istekte bulunulur. (id ve gösterim\_tipi tanımlanmamışsa varsayılan değer “1” kabul edilir). Örneğin:  
[http://www.denemesitesi.com/index.php?id=23&gosterim\\_tipi=2](http://www.denemesitesi.com/index.php?id=23&gosterim_tipi=2)
- 2) Web sunucusu yazılımı, isteği değerlendirir ve DyNA’ya yönlendirir.
- 3) DyNA id parametresiyle belirtilen sayfanın sayfa tipini içerik tablosunun sayfa\_tipi alanı yardımıyla bulur.
- 4) Bulunan sayfa ve gösterim tipine ilişkin içerik bileşenlerine ait şablon dosyaları gosterim adlı tablodan çağırılırlar ve kullanıcı yetkileri doğrultusunda işletilirler.
- 5) Belirtilen id ile tanımlanmış sayfa tipine ait tüm içerik bileşenleri oluşturulduktan sonra elde edilen HTML kodu istekte bulunan İnternet tarayıcısına gönderilir.

Ziyaretçi tarafındaki İnternet tarayıcısı HTML, CSS ve varsa içerisindeki JavaScript kodlarını yorumlar ve sonucu ekranda görüntüler.

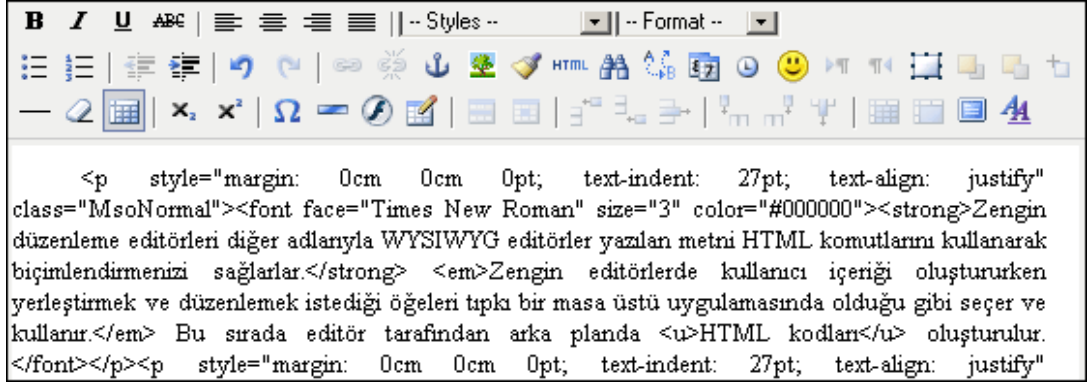
## 5.8. Web Tabanlı Çevrimiçi HTML Editörler

Çevrimiçi HTML düzenleme editörleri, yazılan metni HTML komutlarını kullanarak biçimlendirmenizi sağlarlar. Web tabanlı içerik yönetim sistemlerinde, içerik oluşturma ve düzenleme işlemleri için sıkça kullanılırlar. Zengin editörlerde kullanıcı içeriği oluştururken, yerleştirmek ve düzenlemek istediği öğeleri tıpkı bir masa üstü uygulamasında olduğu gibi seçer ve kullanır (Şekil 5.3). Bu sırada editör tarafından arka planda HTML kodları oluşturulur (Şekil 5.4).



Şekil 5.3: HTML zengin editör normal görünüm

Zengin metin editörleri, genelde teknik olarak TEXTAREA nesnesinde dönüşüm yaparak çalışırlar ve pek çoğu ActiveX desteği veren İnternet tarayıcıları tarafından desteklenirler.



Şekil 5.4: HTML zengin editör HTML kod görünümü

DyNA farklı zengin metin editörüyle uygun çalıştıracak şekilde kurulabilir. Bu editörler çoğu zaman diğer uygulamalardan kopyalama-yapıştırma hizmetini desteklerler. Metinleri biçimlendirmenin yanı sıra resim, tablo, animasyon, özel karakterler, madde imleri ekleme, yazı içerisinde arama ve değişiklikler yapma imkanı veren türleri de vardır.

İçerik oluşturan kişi, HTML bilgisine sahipse, editör tarafından oluşturulan koda müdahale edebilir.

## 6. DyNA YÖNETİM PANELİ

DyNA Yönetim Paneline girebilmek için web tarayıcının adres kısmına [www.orneksite.com/yonetim](http://www.orneksite.com/yonetim) şeklinde giriş yapılır (örneğimizde site adresi olarak [www.orneksite.com](http://www.orneksite.com) alınmıştır). Kullanıcının karşısına gelen ekranda (Şekil 6.1), daha önceden sisteme tanıtılmış ve yetkileri belirlenmiş kullanıcı için, kullanıcı adı ve şifre ikilisinin yanı sıra, her girişte değişen anahtar değerinin karşılığı girilir. Anahtar kullanılmasının sebebi giriş güvenliğinin artırılması ve şifre ele geçirilmiş olduğunda dahi giriş kuralının bilinmemesi halinde sisteme girişin engellenmesidir. Girilen kullanıcı adı, şifre ve anahtar karşılığı doğru ise kullanıcı adı değeri oturum değişkenine saklanarak yönetim paneli ekranı görüntülenir.

Şifrenin unutulması halinde kullanıcı giriş ekranının alt bölümünde yer alan “Şifremi unuttum” bağlantısına tıklanarak sistem kayıtlı kullanıcı adı ve e-posta adresi girilerek belirtilen kullanıcı için yeni bir şifre oluşturulur ve kullanıcı tarafından sisteme daha önceden belirtilmiş e-posta adresine gönderilir.

Yönetim paneline giriş işleminde güvenliği artırmak amacıyla kullanıcının internete bağlanırken kullanmış olduğu IP adresi bir veritabanı tablosunda saklanarak önceden belirlenmiş bir zaman dilimi süresince yetkilendirme de yapılabilir. Bunun için yönetim paneli giriş kısmından önce kullanıcı IP adresinin kayıt edildiği bir giriş paneli kullanılır. Giriş panelinde de sürekli değişen ve algoritması projeye göre belirlenebilen bir dinamik anahtar uygulaması geçerlidir. Yönetim panelinde yapılacak her işlemde önce işlemin yapılmasını isteyen kullanıcının IP adresinin güvenilir ve önceden kayıt edilmiş olup olmadığı karşılaştırması yapılarak adres doğrulandıktan sonra işlemlere devam edilir.



**DyNA İçerik Yönetim Sistemi**

[Kullanıcı Giriş Ekranı]

**Kullanıcı Adı:**

**Kullanıcı Şifre:**

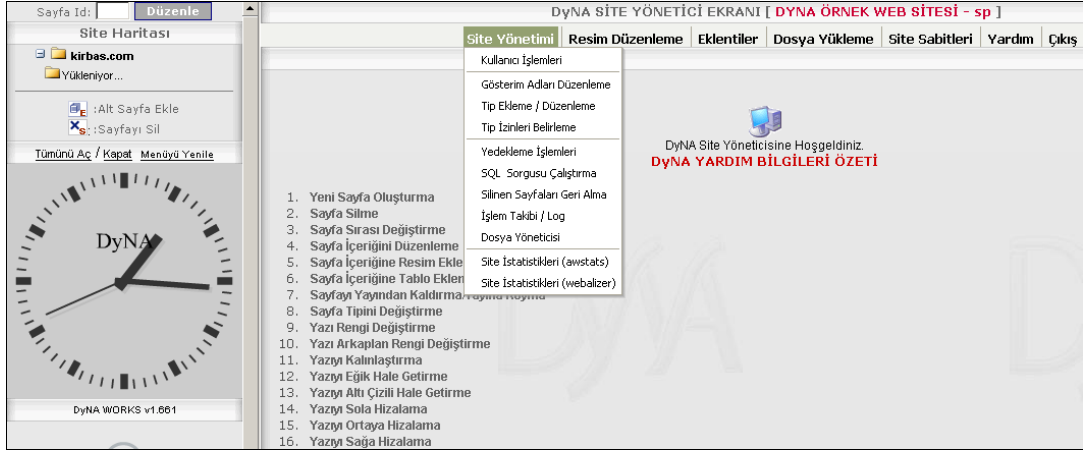
**Anahtar [FuxkO617]**

[127.0.0.1] [Şifremini Unuttum]

Şekil 6.1: DyNA yönetim paneli yönetici giriş ekranı

Kullanıcı giriş ekranının sol alt köşesinde köşeli parantez içerisinde sisteme bağlanılan bilgisayarın IP numarası görüntülenmektedir (Şekil 6.1).

Yönetim panelinde yapılabilecek düzenlemeler, kullanıcı rolünün yetkileri tarafından belirlenir. En yüksek seviyeli rol tipi “superadmin” olarak tanımlanmıştır ve site üzerinde tüm yetkilere sahiptir. Projenin ölçeği doğrultusunda sınırsız sayıda farklı rol ve bu roller için yetki tanımlamaları veya daha önceden verilmiş değerler üzerinde değişiklikler yapılabilir. Yönetim paneli için belirlenecek kullanıcı sayısında da herhangi bir kısıtlama yoktur. Yetkili kullanıcı olarak giriş yapıldıktan sonra DyNA’nın yönetim paneli Şekil 6.2’de gösterildiği şekilde açılmış olur.



Şekil 6.2: DyNA yönetim paneli genel görünümü

Yönetim paneli genel hatlarıyla iki bölümden oluşmaktadır. Yönetim panelini oluşturan web sayfası dikey olarak iki parçalı “frameset” yapısı kullanılarak oluşturulmuştur. Sol taraftaki bölümün içerisinde ağaç yapısı şeklindeki “site haritası” bölümü bulunmaktadır. Site haritasının üst kısmında id numarası bilinen sayfaya doğrudan erişimi kolaylaştırmak için id değeri giriş bölümü yer alır. Site haritasının hemen alt kısmında flash animasyonu olarak hazırlanmış bir saat ve onunda altında DyNA’nın sürüm numarası ve ek modüllere ait bağlantılar gösterilmektedir.

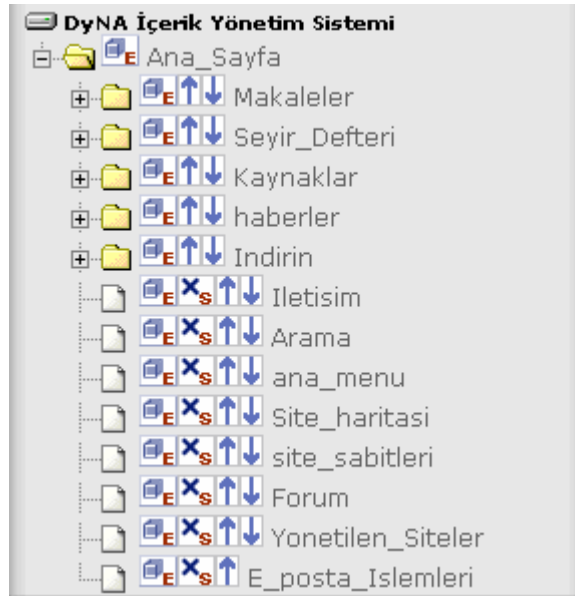
Sağ taraftaki bölüm kendi içinde yatayda iki parçaya bölünmüştür ve üst kısmında; yönetilen web sitesinin adı, o anda yönetim paneline bağlı bulunan kullanıcının adı ve yönetim panelinin ana menüsü yer alır. Alt kısımda ise yönetim paneli ilk yüklendiğinde sistemin genel kullanımıyla ilgili yardım bilgileri görüntülenir. Daha sonra bu alan yönetim paneli ana menüsü ve site haritası üzerindeki komutları yerine getirmek için yüklenen sayfaları görüntüler.

### 6.1. DyNA Yönetim Panelinde Site Haritası Bölümü

Proje ihtiyaçları ve ölçeğine göre yönetim panelinin farklı sürümlerinde site haritasının yapısı değişkenlik gösterse de görünümü ve kullanımı aynı kalmaktadır. Ağaç hiyerarşisindeki yapının kök kısmında web sitesinin adı veya ana sayfa yer alır. Ana sayfanın altında birinci seviyeden ana sayfaya bağlı sayfalar görüntülenir. Ağaç

yapısındaki site haritası elemanlarının sol tarafında izin veya dosya düğmeleri bulunmaktadır (Şekil 6.3). Dizin düğmesi bu sayfaya bağlı alt sayfaların bulunduğunu gösterir ve kendisine bağlı olan alt sayfaların tümü silinmeden silinemezler.

Site haritası ile ilgili bir diğer önemli husus web sitesinde yayınlanmakta olan veya hazırlanmış olup yayınlanmayan tüm sayfaların görüntüleniyor olmasıdır. Aynı isme sahip birden fazla sayfanın oluşturulması her güncelleme işleminde otomatik isim kontrolü ve sayfa numaralandırma (indisleme) metoduyla engellenmiştir. Eğer sayfa oluşturma kısmında yanlışlıkla daha önceden verilmiş bir ad kullanılırsa sayfa adına otomatik numaralandırma ile numara verilir. Örneğin, daha önceden oluşturulmuş “iletisim” adlı bir sayfa varken tekrar aynı isimde bir sayfa oluşturulacak olursa yeni sayfanın adı olarak “iletisim1” şeklinde değiştirilir eğer “iletisim1” adında bir sayfa da mevcut ise “iletisim2” adı verilir ve otomatik numaralandırma bu şekilde devam eder.



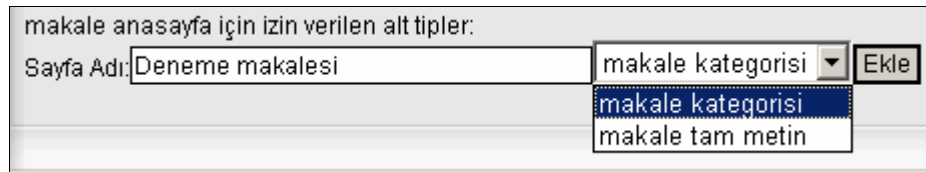
Şekil 6.3: DyNA yönetim paneli sol menü görüntüsü

Sayfa adlandırması yapılırken kullanıcı, Türkçe’de olup İngilizce’de olmayan harfler, boşluklar veya özel karakterler kullanırsa bunlar otomatik olarak İngilizce’deki biçimlerine çevrilir. Örneğin, sayfa adı olarak “İletişim Sayfası” girildiğinde bu ad “Iletisim\_Sayfasi” şeklinde değiştirilecektir.

Web sitesini oluşturacak sayfa adedi bin veya üzerinde olacağı öngörülüyor ise site haritasında HTML ve JavaScript kısıtlamalarından dolayı AJAX teknolojisinin kullanımı gerekir. AJAX destekli site haritasında her bir dal kullanıcının isteği üzerine o anda talep edilerek sunucudan yüklenir. Dolayısıyla site haritası parçalı olarak yüklenmiş olur.

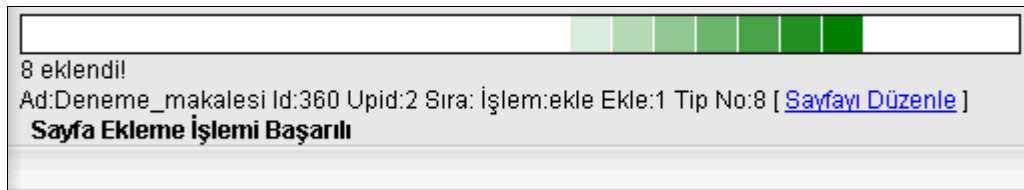
### 6.1.1. Sayfa ekleme

Sayfa ekleme işlemi için öncelikle yeni oluşturulacak sayfaya ebeveyn olacak olan sayfanın önünde bulunan “sayfa ekle” düğmesine basılır. Yönetim panelinin sağ bölümünde yeni oluşturulacak sayfanın adı girilir. Ebeveyn sayfa tipi altına eklenebilecek sayfa tipleri sayfa tipleri liste kutusunda belirtilir. Bunlardan uygun olanı seçilerek “Ekle” düğmesine basılır (Şekil 6.4).



Şekil 6.4: DyNA yönetim paneli sayfa ekleme işlemi

İşlem başarılı ise Şekil 6.5’e benzer bir sonuç görüntüsü elde edilir.



Şekil 6.5: DyNA yönetim paneli sayfa ekleme işlemi başarılı mesajı

“Sayfayı Düzenle” bağlantısına tıklanırsa yeni oluşturulan sayfa için düzenleme ve veri girişi yapılabilir.

### **6.1.2. Sayfa silme**

Site haritasında silinmesi istenen sayfa adının solunda yer alan “sayfa sil” düğmesine tıklandığında eğer kendisine bağlı bir alt sayfa yoksa sayfa silindi olarak işaretlenir ve yayından kaldırılmış olur. Sayfa silme işlemi aslında veri tabanından silme işlemi değil sadece silindi olarak işaretlenmesi şeklindedir. Bu sebeple silinmiş sayfalar daha sonradan istendiği takdirde DyNA yönetim paneli ana menüsü veya sayfa düzenleme kısmında fare ile sağ tıklandığında açılan hızlı menü yardımıyla geri alınabilmektedir.

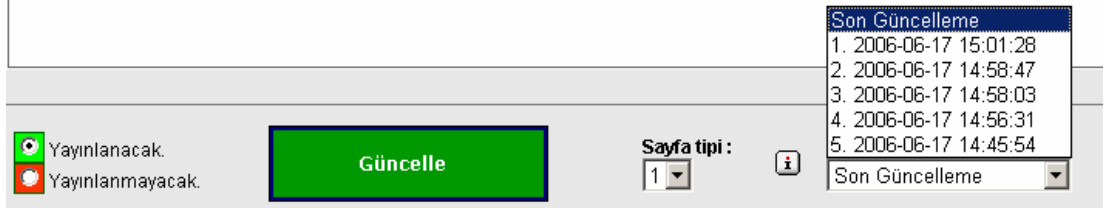
### **6.1.3. Sayfaların sıralamasını değiştirme**

Site haritası içerisinde kardeş sayfalar arasındaki sıralamayı değiştirmek için istenen sayfa adının önünde yer alan aşağı ve yukarı yön simgelerine tıklayarak sıra artırılıp azaltılabilir.

### **6.1.4. Sayfaları düzenleme**

Sayfalar üzerinde bilgi girişi ve düzenleme yapmak için site haritasındaki sayfa adına tıklamak yeterlidir. Sayfanın tipine göre ilgili düzenleme şablonu yönetim panelinin sağ tarafına yüklenerek düzenlemeye hazır hale gelir. Şablon, düzenleme yapılan sayfaya ait içerik bileşenlerini barındırır. Her içerik bileşeni bir XML etiketi içerisinde tutulur ve sayfaya ait XML biçimli bilgi, içerik tablosunun bilgi adlı alanına kayıt edilir.

Sayfanın ziyaretçiler tarafından görüntülenmesi isteniyorsa düzenleme sayfasının sol alt köşesinde yer alan “Yayınlanacak” seçeneği, görüntülenmemesi isteniyorsa da “Yayınlanmayacak” seçeneği işaretlenir. Varsayılan değer “Yayınlanacak” şeklindedir. Düzenlenen sayfanın sayfa tipi değiştirilmek istenirse güncelleme düğmesinin sağında yer alan liste kutusundan istenen sayfa tipi seçilebilir (Şekil 6.6).



Şekil 6.6: DyNA yönetim paneli sayfa düzenleme işlemleri

Sayfanın daha önceki hallerine geri dönmek veya göz atmak için, önceki güncellemelerin listelendiği liste kutusundan seçim yapılabilir.

Yapılan bilgi girişi ve değişiklik işlemlerinin geçerli olabilmesi için son olarak “Güncelle” düğmesine basılması gerekir. Bu işlem için yeterli yetki varsa işlem gerçekleşecek, kaydın değişiklik tarihi sayfanın baş kısmında görüntülenecektir.

#### 6.1.5. Sayfa düzenlemede hızlı menü kullanımı




Sayfa düzenleme bölümünde kullanılan, sağ tıklamak suretiyle aktif olan hızlı menü, üzerinde çalışılan sayfa ile ilgili işlemlere hızlıca erişebilmek amacıyla gerçekleştirilmiştir (Şekil 6.7). Sayfanın kopyasını oluşturma, bağlı olduğu ebeveyni değiştirme, kendisine bağlı alt sayfa oluşturma, kardeş sayfa oluşturma, sayfa silme, sayfanın sıralamasını değiştirme, ebeveyn sayfasını düzenleme, daha önceden silinmiş sayfalara ulaşma, genel yardım bilgilerini görüntüleme, sayfaya ait id, tip numarası, sıra ve ebeveyn id bilgilerine direk ulaşım gibi pek çok fonksiyona doğrudan erişme imkanı verir.

Sayfayı Kopyala Sayfayı Taşı Alt Sayfa Ekle Kardeş Sayfa Ekle Sayfayı <b>Sil</b>
Sayfayı Yukarı Taşı Sayfayı Aşağı Taşı
Düzenle ' <b>Makaleler</b> ' Silinmiş Sayfalar
Yardım
Id= <b>1</b> Tip No: <b>1</b> Sıra: <b>0</b> UpId: <b>0</b>

Şekil 6.7: DyNA yönetim paneli hızlı erişim menüsü

#### 6.1.6. Sayfa içeriğini taşıma

DyNA içerisinde bir sayfanın taşınması, ebeveyninin ve/veya sırasının değiştirilmesi işlemidir. Bunun için, öncelikle sayfa düzenlenecek şekilde site haritasında tıklanır, sonra da düzenleme sayfasının üstünde yer alan taşıma bağlantısına tıklanır veya sağ menü kullanılarak taşıma komutu verilir (Şekil 6.8). Sayfa taşıma bölümü ekranında yeni ebeveynin *id* numarası elle girilir, ardından belirtilen ebeveyn sayfaya bağlı kardeş sayfalar liste kutusunda listelenir ve yeni sıra değeri liste kutusundan seçilerek “Değişikliği Kaydet” düğmesine tıklanıldığında taşıma işlemi gerçekleşmiş olur (Şekil 6.9).

Son güncelleme: 2007-09-14 11:21:16 11:21:16 Sayfa Sürümü: 2 <a href="#">[Taşıma]</a> <a href="#">[Kopyalama]</a>	
Durum: Güncelleme yapıldı.	
<div>  <b>Sayfa Adı:</b> <input type="text" value="yeni_sayfa"/> <input type="checkbox"/> Önizleme Yayına Alınsın   </div>	
Bu sayfa Ürün tipi için	

Şekil 6.8: DyNA yönetim paneli sayfa taşıma işlemi

Sayfa Taşıma Bölümü					
Başlık	Id	UpId	Sıra	Yeni UpId	Yeni Sıra
Grafik_Tasarimcinin_Egitimi	78	94	2	65	6

Değişikliği Kaydet

Şekil 6.9: DyNA yönetim paneli sayfa taşıma işlemi

#### 6.1.7. Sayfa içeriğini kopyalama

Sayfa kopyalama işlemi bir sayfa içeriğinin tümüyle bir başka sayfa içeriğine kopyalanması işlemidir. Bunun için sayfa düzenlenecek şekilde site haritasında tıklanır, ardından düzenleme sayfasının üstünde yer alan kopyalama bağlantısına tıklanır veya sağ menü kullanılarak kopyalama komutu verilir. Gelen ekranda web sitesi içerisinde seçilen sayfa ile aynı tipe sahip bütün sayfalar adlarıyla birlikte listelenmektedir (Şekil 6.10). Listedten içeriği kopyalanacak kaynak sayfa seçilir ve “Bilgileri Kopyala” düğmesine tıklanır.

İçeriği Kopyalanacak Sayfa	<div>Seçiniz</div> <div> Seçiniz  aa  Avantajlar  Avantajlari  Bireyler_Icin  Dinamik İçerik Yönetimi  Dyna Works  Dyna_Surum_Bilgileri  E-ticaret  Ekran_Goruntuleri  Hazirlama_Yayinlama_Asamalari  Isletmeler_Icin  Niçin Web Sitesi  Tasarım Öncesi Gerekli Belgeler  Tasarıma başlarken  yeni_sayfa </div>	Bilgileri Kopyala
----------------------------	---	-------------------

Şekil 6.10: DyNA yönetim paneli sayfa kopyalama işlemi



## 6.2. Kullanıcı İşlemleri

Yönetim Paneli ana menüsünün “Site Yönetimi” bölümünün altındaki ilk madde “Kullanıcı İşlemleri”dir. Bu seçeneğe tıklanıldığında Şekil 6.11’de gösterilen DyNA kullanıcıları bölümü sayfası görüntülenir. Bu bölümde kullanıcılar ve rollerle ilgili tüm işlemler tanımlanmıştır. Öncelikle, yönetim paneline tanıtılmış kullanıcıların adları, e-posta adresleri ve rolleri bir tabloda listelenir. Bu tablonun altında kullanıcılar için tanımlanmış işlemler, bağlantılar şeklinde görüntülenmektedir.

DyNA kurulum aşamasında yapılacak ilk işlemlerden biri sistemde yer alacak tüm rollerin ve rollere ait yetki seviyelerinin belirlenmesidir. Bu işlemin ardından, sırasıyla kullanıcılar tanımlanıp rolleri belirlenebilir.

DyNA SİTE YÖNETİCİ EKRANI [ DYNA ÖRNEK WEB SİTESİ - sp ]

Site Yönetimi | Resim Düzenleme | Eklentiler | Dosya Yükleme | Site Sabitleri | Yardım | Çıkış

DyNA Kullanıcıları Yönetim Bölümü

Hoşgeldin sp,

Sıra	Kullanıcı Adı	E-posta	Rol
1	ed	sanalman@hotmail.com	editor
2	superadmin	sanalman@hotmail.com	superadmin
3	sp	sanalman@hotmail.com	superadmin
4	yasemin	yasemin@a.com	yetkili
5	ismail	hjl@azxs.ert	yetkili
6	ziyaretci	sanalman@hotmail.com	ziyaretci

Yeni Kullanıcı Oluşturma | Kullanıcı Şifre Değiştirme | Kullanıcı Silme | Yeni Rol Tanımlama | Rol Düzenleme | Rol Silme

Yeni Kullanıcı Ekranı

Kullanıcı Adı: Günay

Şifre: .....

E-posta: gunaysunay@yahoo.com

Rol: editor

Yeni Kullanıcı Oluştur

[ Çıkış ]

Şekil 6.11: DyNA kullanıcıları yönetim bölümü ekran görüntüsü

### 6.2.1. Yeni rol tanımlama işlemi

Şekil 6.12’de “editör” adıyla görme, okuma, yazma, değiştirme, silme, ekleme ve süper yetki seviyeleri tanımlanmış bir rol oluşturulmaktadır. Roller ve yetkiler proje ihtiyaçları doğrultusunda belirlenip artırılıp azaltılabilir. Yetkiler ve yetki seviyeleri planlama safhasında belirlenmiş olmalıdır.

Etkili bir site yönetimi açısından örneğin yazma yetkisi değeri “3” şeklindeki bir atama için, 3 yazma seviyesinin sitenin hangi bölümleri ve hangi sayfa tipleri için geçerli olacağı önceden belirlenmiş olmalıdır.

Süper yetki seviyesi, site yönetiminden sorumlu kullanıcıların yetki hiyerarşisini belirlemek için kullanılabilir. Web sitesinde hiçbir yetkisi olmayan ziyaretçiler içinde bir rol tanımlaması yapılmalı ve en azından görme yetki seviyesi “0” değerinden farklı olmalıdır. Temel rol tanımlamaları (ziyaretçi, editör, yetkili, süperadmin vb.) DyNA’nın kurulumu esnasında yapılmaktadır. İstendiği takdirde bunlar üzerinde de değişiklikler yapılabilir.

Görme	Okuma	Yazma	Değiştirme	Silme	Ekleme	Süper
1	5	3	3	2	2	0

Yeni Rol Oluştur

Şekil 6.12: DyNA yönetim paneli yeni rol tanımlama bölümü ekran görüntüsü

### 6.2.2. Kullanıcı ekleme

Yönetim sistemine yeni bir kullanıcı eklemek için önceden kullanılmayan bir kullanıcı adı, şifre ve e-posta adresi belirlenir. Seçmeli liste kutusundan yetki seviyesini belirten rol seçilir (Şekil 6.13). “Yeni Kullanıcı Oluştur” düğmesine tıklandığında belirtilen e-posta adresine kullanıcı oluşturma ile ilgili bir bilgilendirme postası yollanır. Bu işlem aynı zamanda yönetim panelinin log dosyasına kayıt edilmektedir.

**Yeni Kullanıcı Ekranı**

Kullanıcı Adı: Günay

Şifre: ••••••

E-posta: gunaysunay@yahoo.com

Rol: editor ▼

Yeni Kullanıcı Oluştur

Şekil 6.13: DyNA yönetim paneli kullanıcı ekleme bölümü ekran görüntüsü

### 6.2.3. Kullanıcı silme

Kullanıcı silme işlemi için silinecek kullanıcı adı seçmeli liste kutusundan seçilir, “Seçili kullanıcıyı SİLMEK istiyorum” seçeneği işaretlenir ve “Kullanıcıyı Sil” düğmesine basılır (Şekil 6.14). Bu işlemi yapacak kullanıcının, kullanıcı silme yetkisine sahip olması gerekir. Yapılan işlem log dosyasına kayıt edilir. Yukarıdaki örnekte kullanıcı adı “ed”, kullanıcı rolü “editör”dür.

**Kullanıcı Silme Ekranı**

Kullanıcı Adı: ed-editor ▼

☒ Seçili kullanıcıyı SİLMEK istiyorum.

Kullanıcıyı Sil

Şekil 6.14: DyNA yönetim paneli kullanıcı silme bölümü ekran görüntüsü

### 6.2.4. Rol düzenleme

Sisteme tanımlı roller seçmeli liste kutusundan seçilir, ardından seçili olan rolün yetki seviyeleri görüntülenir ve istenilen yetki seviyesi belirlendikten sonra “Rol Değişikliklerini Onayla” düğmesine basılır (Şekil 6.15). Bu işlemi gerçekleştirecek kullanıcının rol düzenleme işlemi için gerekli yetki seviyesine sahip olması gereklidir.

**Rol Düzenleme Ekranı**

Rol Adı:

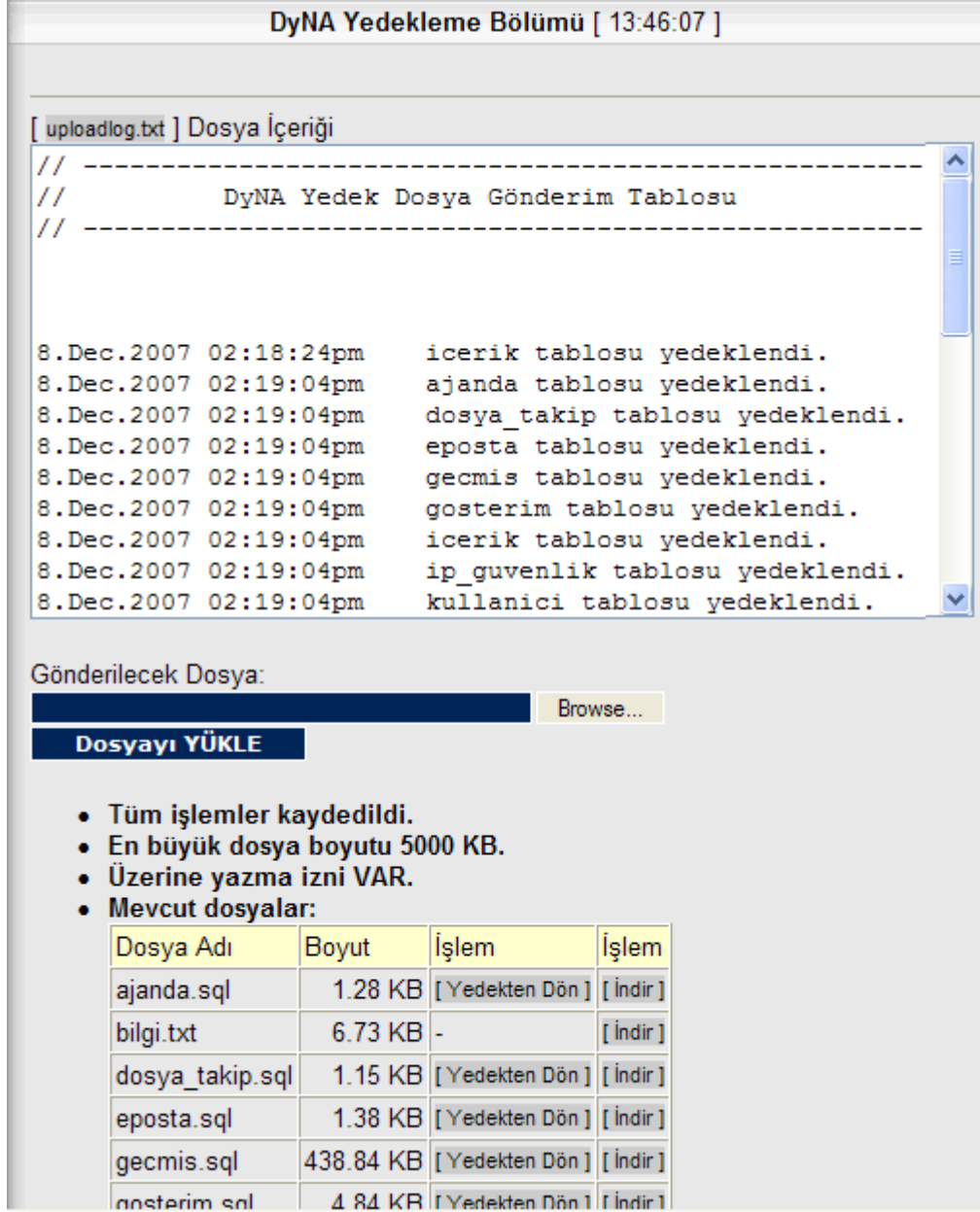
Görme	Okuma	Yazma	Değiştirme	Silme	Ekleme	Süper
<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

Şekil 6.15: DyNA yönetim paneli rol düzenleme bölümü ekran görüntüsü

### 6.3. DyNA’da Yedek Alma Sistemi

DyNA’da içerik ve yönetime ilişkin tüm bilgiler MySQL tablolarında saklanmaktadır. Yedek alma işlemi; tüm tabloların (içerik, gösterim yapısı, kullanıcılar, roller, tipler, roller, içerik arşivi ve diğer modüllere ait tablolar) sunucu üzerinde dosyalara dönüştürülmesi, sıkıştırılması, istemci tarafına indirilmesi, yedek alınmış dosyaların sisteme geri yüklenmesi, yapılan işlem tarihlerinin kayıt edilmesi ve görüntülenmesi, istemci tarafından dosya yüklenmesi, MySQL veritabanı üzerinde SQL sorgulamaları yapılması gibi fonksiyonların yerine getirildiği bölümdür.

Yedekleme ve yedekten geri dönme işlemi bir WIYS için hayati önem taşır. Veritabanı kullanan bir WIYS’de yetkiler ölçüsünde serbest SQL sorgularının çalıştırılması sistemin esnekliğini artırır. DyNA tek düğme ile tüm tabloların yedeklenmesine tüm site içeriğinin sıkıştırılmış tek bir dosya haline getirilmesine, sunucu ve istemci tarafında yedeklenmesine imkan tanır (Şekil 6.16). Mevcut sitede yer alan tüm veritabanı tabloları otomatik olarak tanınır ve tek tek veya toplu olarak yedeklenebilir.



Şekil 6.16: DyNA yedekleme bölümü ekran görüntüsü

#### 6.4. DyNA Dosya Yöneticisi

Bir WİYS'i meydana getiren en önemli bileşenlerden birisi de sunucu üzerinde yer alan dosyalardır. Bu sebeple, WİYS dosya yönetimi için kendi içerisinde yer alan veya üçüncü parti bir yazılım kullanılabilir.

DyNA içerisinde bütünleşik olarak dosyalar üzerinde işletim sistemi operasyonlarını yapabilmeyi sağlayan bir dosya yöneticisi yer alır (Şekil 6.17). Dosya yöneticisi ile

sunucu tarafında; yüklü dosyalar üzerinde kopyalama, taşıma, ad değiştirme, dosya içeriğini görüntüleme, düzenleme, silme, oluşturma, izin oluşturma, istemci tarafından dosya yükleme, istemci tarafına dosya indirme gibi işlemler yapılabilir.

DyNA Dosya Yöneticisi							
Ana Sayfa   Çıkış							
Bölge: Yol: http://www.dyna.web.tr/							
<input type="button" value="*Yeni"/> <input type="text"/> <input type="button" value="Dizin Oluştur"/> <input type="button" value="Dosya Yükle"/>							
Dosya Adı	Ebat	Tip	İşlem				
menu.js	9.75 KB	GNC	---	Kopyala	Ad Değiştir	Sil	
menu_tpl.js	910 B	GNC	---	Kopyala	Ad Değiştir	Sil	
menu.css	1.66 KB	CSS	Düzenle	Kopyala	Ad Değiştir	Sil	
css_kalip.php	4.49 KB	PHP	Düzenle	Kopyala	Ad Değiştir	Sil	
xml.png	193 B	PNG	---	Kopyala	Ad Değiştir	Sil	
cgi-bin	4 KB	Folder	---	---	Ad Değiştir	Sil	
postinfo.html	2.39 KB	HTML	Düzenle	Kopyala	Ad Değiştir	Sil	
_vti_inf.html	1.71 KB	HTML	Düzenle	Kopyala	Ad Değiştir	Sil	
images	4 KB	Folder	---	---	Ad Değiştir	Sil	
favicon.ico	459 B	GNC	---	Kopyala	Ad Değiştir	Sil	
loremipsum.txt	2.95 KB	TXT	Düzenle	Kopyala	Ad Değiştir	Sil	
kalip.php	252 B	PHP	Düzenle	Kopyala	Ad Değiştir	Sil	
dyna.css	4.99 KB	CSS	Düzenle	Kopyala	Ad Değiştir	Sil	
dyna_kur.php	12.94 KB	PHP	Düzenle	Kopyala	Ad Değiştir	Sil	
robots.txt	24 B	TXT	Düzenle	Kopyala	Ad Değiştir	Sil	

Şekil 6.17: DyNA dosya yöneticisi ekran görüntüsü

## 6.5. DyNA İşlem Takibi – LOG

Bir WİYS’de yapılan tüm işlemlerin iş akış şeması içerisinde gerçekleşip gerçekleşmediğinin takip edilmesi gerekmektedir. Hangi kullanıcının, ne zaman, hangi işlemi yaptığının kaydı tutulabilir ise buna göre iş akışının daha verimli ve daha güvenli bir hale getirilmesi için istatistik verileri elde edilmiş olur (Şekil 6.18).

DyNA’da kendi içerisinde bir işlem takip sistemine sahiptir. Sayfa oluşturma, düzenleme, silme, kullanıcı oluşturma, silme, değiştirme, dosya gönderimi, yedekleme işlemleri gibi yapılan işlemler zaman, kullanıcı ve işlem açıklaması şeklinde kaydedilmekte ve tarih bazında sondan başa doğru listelenerek gösterilebilmektedir.

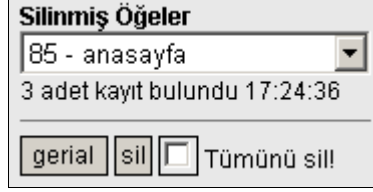
Yönetici Hareketleri Takip Ekranı			
Kayıt adedi = 90 Son Güncelleme: 19.12.2007 11:55:15			
Zaman	Id	Kullanıcı Adı	İşlem
2007-12-19 19:55:15	103	sp	tarafından güncellendi.
2007-12-19 19:55:14	103	sp	tarafından güncellendi.
2007-12-19 19:53:17	103	sp	tarafından oluşturuldu.
2007-12-08 23:25:35	102	sp	tarafından güncellendi.
2007-12-08 23:21:45	102	sp	tarafından güncellendi.
2007-12-08 23:21:09	102	sp	tarafından güncellendi.
2007-12-08 23:20:16	102	sp	tarafından oluşturuldu.
2007-12-08 21:07:46	4	sp	tarafından güncellendi.
2007-12-08 21:06:44	73	sp	tarafından güncellendi.
2007-12-08 21:05:34	73	sp	tarafından güncellendi.
2007-12-08 20:52:57	73	sp	tarafından güncellendi.
2007-12-08 20:18:12	1	sp	tarafından güncellendi.

Şekil 6.18: DyNA işlem takibi ekran görüntüsü

## 6.6. Silinen Sayfanın Geri Alınması

DyNA sisteminde silinen sayfalar/kayıtlar gerçekte veritabanından fiziksel olarak silinmiş olmazlar, sadece silindi şeklinde işaretlenirler. Kaydın silinmesi demek içerik tablosunda durum alanındaki değerin 0 haline getirilmesi demektir. Geri alma işlemi de esasında ilgili kaydın durum alanındaki bilginin 1 haline çevrilmesidir. Geri alma bölümünde istenirse kayıtlar kalıcı olarak bir daha geri dönmeyecek şekilde de silinebilir.

Yönetim panelinde silinmiş olan sayfalar site yönetimi menüsü altında yer alan “Silinen Sayfaları Geri Alma” bölümünden veya sayfa düzenleme kısmında iken, fare ile sağ tıklanarak açılacak menüden “Silinmiş Sayfalar” maddesinin tıklanmasıyla açılan sayfadan, liste kutusu içerisinde sayfa id değeri ve sayfa tipi ile listelenirler. İstenilen sayfa seçilerek “geri al” düğmesine tıklandığında silme işlemi geri alınmış olur. Sayfaları kalıcı olarak silmek için ise “sil” düğmesi kullanılır. Silinmiş tüm sayfaların veritabanından kalıcı olarak silinmesi için “Tümünü sil!” seçeneği işaretlenir.



Şekil 6.19: DyNA yönetim panelinde silinen sayfanın geri alınması

## 6.7. Gösterim Adlarının Düzenlenmesi

DyNA içerisinde yer alan tip adları ve gösterim tipi şablon eşleştirmeleri “gösterim” tablosunda tutulmaktadır. Bu tablodaki kayıtlar “Gösterim Adları Düzenleme” bölümünde düzenlenir.

gosterim tablosundaki kayıtlar id, gösterim adı, tip adı, gösterim tipi ve dosya adına göre sıralanarak listelenir (Şekil 6.20). Tablo üzerinde gösterim adı, tip adı ve gösterim tipi alanlarında sorgulama yapılabilir. Kayıt ekleme, değiştirme, silme işlemleri gerçekleştirilebilir.

“düzenle” gösterim adı DyNA için özeldir. Sayfa tiplerinin içerik bileşenlerinin yapısını barındırırlar ve gösterim tipi olarak “1” değerine sahiptirler. “temel” tip adı da DyNA için ayrı bir öneme sahiptir. Her gösterim adı için bir temel tip şablonu hazırlanmalıdır. Bu DyNA için varsayılan tip şablonlarını göstermektedir.



## Gösterim Adları Yönetim Bölümü

Gösterim Adı	Tip Adı	GT	Dosya Adı
<input type="text" value="duzenle"/>	<input type="text" value="fotograf"/>	<input type="text" value="1"/>	<input type="text" value="d_tip_7.php"/>
İşlem: <input type="button" value="Değiştir"/> <input type="button" value="Onayla"/>			

## Sorgulama Bölümü

<input type="text" value="altbar"/>	<input type="text" value="kök"/>	<input type="text" value="1"/>
<input type="button" value="Sorgula"/>		

ID	Gösterim Adı	Tip Adı	GT	Dosya Adı	İşlem
17	altbar	kök	1	d_altbar.php	Düzenle
7	altbar	temel	1	d_altbar.php	Düzenle
13	altmenu	temel	1	d_altmenu.php	Düzenle
9	anamenu	temel	1	d_anamenu.php	Düzenle
23	anamenubar	temel	1	anamenubar.php	Düzenle
84	anket	Anket	1	anket.php	Düzenle
33	cocuk_yaz	temel	1	cocuk_yaz.php	Düzenle
20	duzenle	ileti?im	1	d_tip_3.php	Düzenle
22	duzenle	foto galeri	1	d_tip_5.php	Düzenle
53	duzenle	alt galeri	1	d_tip_5.php	Düzenle
29	duzenle	fotograf	1	d_tip_7.php	Düzenle
71	duzenle	Kategori	1	d_tip_9.php	Düzenle
72	duzenle	Makale	1	d_tip_9.php	Düzenle
19	duzenle	site sabitleri	1	d_tip_11.php	Düzenle

Şekil 6.20: DyNA yönetim panelinde gösterim adlarının düzenlenmesi

İçeriği değiştirilmek istenen şablon dosyası hizasına üzerine gelinerek “Düzenle” bağlantısına tıklandığında şablon bileşeni dosyası görüntülenir ve kullanıcı yetkileri doğrultusunda istenen değişiklikler gerçekleştirilebilir.

## 6.8. Sayfa Tiplerinin Düzenlenmesi

DyNA’da içerik kayıtları, içerik dizininde tip numaraları ile tutulmaktadır. Yönetimi kolaylaştırmak ve anlaşılır kılmak üzere her tip için bir isim verilmiştir. Tipler üzerinde ekleme, değiştirme ve silme işlemleri yönetim paneli üzerinde

gerçekleştirilebilir (Şekil 6.21). DyNA içerisinde tip sayısı konusunda herhangi bir sınırlama bulunmamaktadır.

Id	Tip Adı	Tip No
7	fotograf	7

İşlem: Değiştir Onayla

ID	Tip Adı	Tip No
6	alt galeri	6
14	alt madde	13
4	arama	4
5	foto galeri	5
7	fotograf	7
19	haber	19
18	haberler	18
11	hakkımızda	10
3	iletişim	3
1	kök	1
13	madde	12
16	menu	15
20	rss	20
17	site haritası	17
12	site sabitleri	11
15	slide show	14
8	temel	99
10	ürün açıklama	9
2	ürün anasayfa	2
9	ürün kategorisi	8

Kayıt Adedi:20

Şekil 6.21: DyNA yönetim panelinde sayfa tiplerinin düzenlenmesi

## 6.9. Tip İzinlerinin Düzenlenmesi

Tip izinlerinin düzenlenmesi, sayfa ekleme sırasında hangi sayfa tipinin altına hangi sayfa tiplerinin eklenebileceğinin belirlenmesi için gereklidir. Ata sütununda belirtilen tipin karşısına denk gelen çocuk sütunundaki tip eklenebilir. Aynı tipin altına kendi tipinden eklenme izni verilebilir yada tipler arasında bire çok eşleme yapılabilir.

Tip izinlerine ilişkin kayıtlar “tip\_izinleri” tablosunda tutulmaktadır. Ekleme, değiştirme ve silme işlemleri yönetim paneli üzerinden yapılabilir.

Id	Ata	Çocuk
15	slide show	fotograf

İşlem: Ekle Onayla

Ekle  
Değiştir  
Sil

ID	Ata	Çocuk
21	kök	rss
20	haberler	haber
19	kök	haberler
18	kök	site haritası
17	kök	menu
16	ürün kategorisi	ürün kategorisi
15	slide show	fotograf
14	kök	slide show
13	ürün anasayfa	ürün açıklama
12	ürün kategorisi	ürün açıklama
11	ürün anasayfa	ürün kategorisi
10	madde	alt madde
9	alt galeri	fotograf
8	foto galeri	alt galeri
7	kök	site sabitleri
6	kök	hakkımızda
5	kök	foto galeri
4	kök	arama
3	kök	madde
2	kök	iletişim
1	kök	ürün anasayfa

Kayıt Adedi:21

Şekil 6.22: DyNA yönetim panelinde sayfa tip izinlerinin düzenlenmesi

#### 6.10. Sunucu Tarafına Toplu Dosya ve Resim Yükleme

Sunucu tarafında bulunan tüm resimler bu bölümde dosya adları, dosya ebatları ve son güncelleme tarihleri listelenerek görüntülenir. Bir sayfada kaç dosyanın listeleneceği daha önceden ayarlanabilmektedir.

İstemci tarafındaki birden fazla dosyasının tek seferde sunucu üzerine gönderilmesi için “çoklu gönderim” seçeneği kullanılabilir (Şekil 6.23). Burada gönderilecek dosya sayısı belirtilerek belirtilen sayıda toplu gönderim gerçekleştirilmiş olur.

**YÜKLENECEK DOSYA: (EN ÇOK 3000000 BYTE)**

☐ Eskisinin üzerine yazılsın mı? **Çoklu gönderim**

[ Dosya 1 / 3 **1 2 3 > »** ] [ [Hepsini Göster](#) ]

	Dosyalar +	Ebat(KB)	Son Güncelleme	<input type="checkbox"/>
1	ALTBASLIK2.JPG	9.53	20/11/2004 4:38 pm	<input type="checkbox"/>
2	ALTBASLIK3.JPG	10.11	20/11/2004 7:26 pm	<input type="checkbox"/>
3	BASLIK.JPG	27.13	20/11/2004 4:38 pm	<input type="checkbox"/>
4	DYNA_FON1.JPG	4.41	16/05/2006 8:24 pm	<input type="checkbox"/>
5	ImageEditor.php	10.32	16/05/2004 9:15 pm	<input type="checkbox"/>
6	JOIN.GIF	0.07	16/05/2006 8:22 pm	<input type="checkbox"/>
7	KULLANICI.GIF	0.39	16/05/2006 8:02 pm	<input type="checkbox"/>
8	S.GIF	0.04	16/05/2006 8:22 pm	<input type="checkbox"/>
9	arrow_r.gif	0.83	16/05/2006 8:22 pm	<input type="checkbox"/>
10	arrow_r1.gif	0.83	16/05/2006 8:22 pm	<input type="checkbox"/>

[ Dosya 1 / 3 **1 2 3 > »** ] [ [Hepsini Göster](#) ]

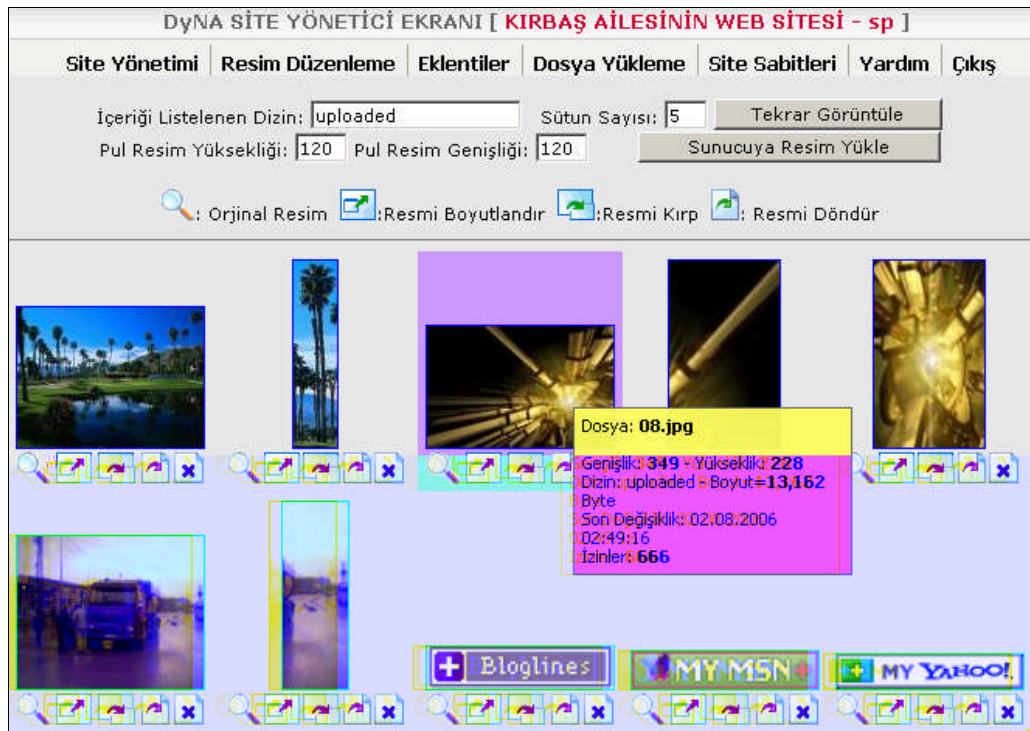
Şekil 6.23: Sunucu tarafına toplu dosya ve resim yükleme

### 6.11. DyNA İçerisindeki Sunucu Tarafı Çevrimiçi Resim Araçları

Gelişmiş bir WIYS bünyesinde içerik üretme ve düzenlemeye ilişkin çeşitli araçlar barındırılır. İçerik üretimi sırasında, üzerinde işlem yapılan temel materyaller metinler ve resimlerdir. Resimler üzerinde çok fazla tecrübe gerektirmeden kolay kullanım imkanı veren araçların varlığı içerik üreticilerin işlerini kolaylaştırır. Bu amaçla, DyNA içerisinde imajlar üzerinde çeşitli temel resim işleme fonksiyonlarını yerine getirecek bir araçlar dizisi yer almaktadır.

Yönetim paneli ana menüsünden “Resim Düzenleme”, buradan da “Resim Boyutlandırma” komutu seçilir ise, sunucuya yüklenmiş olan resimlerin küçültülmüş kopyaları oluşturularak tablo halinde görüntülenir.

Resimlerin üzerine fare ile gelindiğinde, resim hakkındaki dosya adı, piksel cinsinden yükseklik ve genişlik bilgisi, dosya boyutu, son değişiklik zamanı ve dosya erişim yetkisi bilgilerine ulaşılabilir (Şekil 6.24). Resim araçları kullanılarak boyutlandırma, kırpma ve saat yönünde döndürme işlemleri gerçekleştirilebilir.



Şekil 6.24: DyNA yönetim paneli resim işleme bölümü ekran görüntüsü

#### 6.11.1. Resim boyutlandırma

Resmi boyutlandırma, basitçe resmin üzerine bir kez tıkladıktan sonra kesikli mavi şeritlerden tutup sürükleyerek veya resim bilgisi bölümünde yer alan piksel cinsinden genişlik ve yükseklik değeri girilerek yapılır (Şekil 6.25). “En boy oranı koru” seçeneği işaretlenerek resmin orantısı bozulmadan büyütme ve küçültme yapılabilir. Resme son boyutlandırma yapıldıktan sonra “Son Halini Kaydet” düğmesine basılarak kayıt işlemi tamamlanır. Boyutlandırma sırasında herhangi bir hata oluşmaz ise işlemin başarılı olduğuna dair bir mesaj görüntülenir.



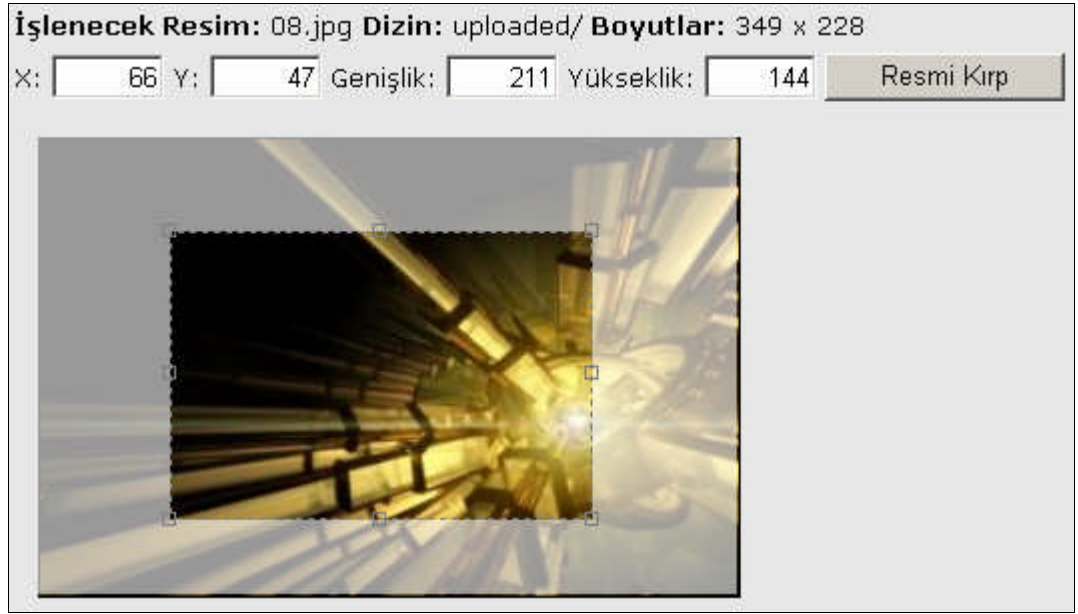
Şekil 6.25: DyNA resim boyutlandırma bölümü ekran görüntüsü

### 6.11.2. Resim kırpma

Kırılmak istenen resim üzerinde net görünen bölge kırpma işlemi sonunda elde edilecek sonucu göstermektedir. Dörtgeninin kenar ortalarında ve köşelerinde bulunan küçük kareler, fare yardımıyla hareket ettirilerek seçili bölge taşınabilir ve boyutlandırılabilir (Şekil 6.26). Seçili bölgenin sol üst koordinatlarının, piksel cinsinden X ve Y kutucuklarına girilerek, dikdörtgenin genişlik ve yükseklik değerleri de belirtilebilir.

Kırpma işleminden sonra üzerinde işlem yapılan dosyada herhangi bir değişiklik olmazken seçili bölge, çalışılan dosya ile aynı biçime sahip, seçme dörtgeninin ebatlarına göre isimlendirilmiş bir resim dosyası haline getirilir.

“İşlenecek Resim” bölümünün karşısında, üzerinde çalışılan resmin dosya adı, bulunduğu dizin ve ebatları görüntülenir. Resim kırpma işlemi hatasız olarak tamamlanırsa kırılan bölgenin yeni bir dosya adıyla (“08\_211\_144.jpg”) kaydedildiğini söyleyen bir bilgilendirme mesajı görüntülenir.



Şekil 6.26: DyNA resim kırpma bölümü ekran görüntüsü

## 7. SONUÇLAR

Bu tez çalışmasında, tamamen Türkçe olarak hazırlanmış ve kendi geliştirdiğimiz bir dinamik web sitesi içerik yönetim sistemi olan DyNA tanıtılmıştır. Birbirlerinin içerisinden çağırılabilen şablon-bileşenlerin kullanılmasıyla karmaşık parçaların oluşturulması prensibine dayanan DyNA, web sitesi geliştiricileri ve kullanıcıları için kolaylıklar ve performans sağlamaktadır. Sitenin bütünlüğü bozulmadan çok kısa sürelerde içerik ve sayfa eklenebilmekte tüm sitede bir anda tasarımla ilgili değişikliklere gidilebilmektedir.

DyNA kullanıcılar ve geliştiriciler için karmaşık web sayfalarını yönetmek üzere dinamik şablon-bileşenler kullanarak kolay metotlar sunmaktadır. Bileşik görünüm, içeriğin sayfa yapısından ayrılması, sayfa görünümünün parçalar halinde ele alınması gibi pek çok özelliğe sahiptir. Aynı anda birden fazla kullanıcının veri girişine ve yetkileri doğrultusunda işlemler yapmasına olanak tanır. Dahası rol tabanlı kullanıcı sistemi sayesinde farklı yetki seviyelerdeki farklı kullanıcıları destekler. XML, DyNA'nın çekirdeğini oluşturmaktadır. Bu yaklaşım sayesinde web sitesi içerisinde veya farklı web siteleri arasında veri transferi ve sorgulaması yapılabilmektedir. Aynı zamanda çeşitli platformlar için çeşitli biçimlerde yayın yapılması sağlanmıştır.

DyNA yapısı itibarı ile çok çeşitli görünüm stillerini desteklemektedir. Geliştiriciler kendi görünüm stillerini de sınırsız olarak gerçekleştirebilirler. Bir web sitesi içerisinde kullanılacak her bir farklı yerleşim stili “sayfa tipi” olarak tanımlanmaktadır. Site yayına konduktan sonra da yeni sayfa tipleri türetilip eklenebilmektedir.

DyNA ile hazırlanmış herhangi iki sitenin görsel olarak herhangi bir ortak yönünün veya benzerliğinin olması gerekmemektedir. Site kurulumu için önceden belirlenmiş bir şablon kullanılabileceği gibi ihtiyaç ve istekler doğrultusunda eklemeler yapmak mümkündür.



DyNA aynı zamanda gelişmiş veri gönderme ve veri alma özelliklerine de sahiptir. Eğer bir web sitesi kendi içeriğinin XML, RSS veya RDF şeklinde ihraç edebiliyorsa bu içerik XML yapısı kullanılarak kolaylıkla DyNA kullanılan başka bir web sitesi tarafından ithal edilebilir. DyNA, farklı platformlar için (RSS, WML, PDF gibi) farklı çıkış şekillerinde çıktılar sunabilecek kapasiteye sahiptir. Ayrıca, DyNA'nın şablon sistemini kullanmak için yeni yazım kurallarının öğrenilmesi gerekmez. Projenin boyutları ve ihtiyaçları doğrultusunda genişletilebilir ve esnek bir yapıya sahiptir. Geliştiriciler her zaman için yeni bir şablon-bileşeni, sayfa tipi, kullanıcı, kullanıcı rolü, kategori, içerik, dil ve hatta yeni bir platform desteği ekleyebilirler.

DyNA ile; hızlı çalışan, kolay kullanılan, kolay programlanabilen, ekleme, silme ve değiştirme işlemlerinin kolaylıkla gerçekleştirilebileceği, sayfaların teknik bilgi gerektirmeden oluşturulabileceği, yetki ve rol tabanlı bir yönetim sistemi olan, çoklu dil desteğine sahip, kişiselleştirilebilen, proje ölçeğine göre genişletilebilen, güvenilir, XML, RSS, RDF, XHTML, WAP gibi teknolojileri destekleyen, sadece bilgisayarlar tarafından değil aynı zamanda mobil cihazlarla da gezilebilir siteler üreten, birbirleriyle veri alışverişi ve karşılıklı sorgulama yapabilen, kullanıcı tarafında herhangi bir program kurulumu gerektirmeyen, tamamen Türkçe olarak yazılmış, özgün, esnek yapı, geliştirilebilir, dinamik, çoklu yazar ve editör desteğine sahip, bir web sitesi yönetim ve geliştirme aracı hazırlanmıştır.

Sonuç olarak DyNA, web sitesi içerik yönetimi ile ilgili gerçek problemlere etkili ve kalıcı çözümler üretilmektedir. Değişen ve gelişen web teknolojilerini destekleyecek esnek ve genişleyebilir bir altyapı sağlamaktadır. İçerik üretimini, yönetimini ve yayınlanmasını belli bir iş akışı içerisinde gerçekleştirerek, birbirinden farklı sorunlara sahip, farklı ve duruma özel çözümler gerektiren web siteleri için ortak bir platform oluşturmaktadır.

## KAYNAKLAR

- [1] Morville, P., Rosenfeld, L., "Information Architecture for the World Wide Web", 2nd Edition, **O'Reilly**, USA., (2002).
- [2] Browning, P., Lowndes, M., "JISC Techwatch Report: Content Management Systems", **Technical Report TSW01-02, Joint Information Systems Committee**, (2001).
- [3] Michelinakis, D., "Open Source Content Management Systems : An Argumentative Approach", **The University of Warwick, Warwick Manufacturing Group**, August (2004).
- [4] Mauthe, A., Thomas, P., "Professional Content Management Systems: Handling Digital Media Assets", **J.Wiley**, (2004).
- [5] Rockley, A., Kostur, P., Manning, S., "Managing Enterprise Content: A Unified Content Strategy", 1st edition, **New Riders Press**, ISBN: 0735713065, (2002).
- [6] Friedlein, A., "Web Project Management: Delivering successful commercial web site", **Morgan Kaufmann Publishers**, (2001).
- [7] Vidgen, R., Goodwin, S., Barnes, S., "Web Content Management", **14th Bled Electronic Commerce Conference Bled**, Slovenia, June 25 - 26, 465-480, (2001).
- [8] Balasubramanian, V., and Bashian, A., "Document Management and Web Technologies: Alice Marries the Mad Hatter," **Communications of the ACM**, vol. 41, pp. 107-115, (1998).
- [9] Maurer, H., "Web-based knowledge management," **IEEE Computer**, pp.122-123, (1998).
- [10] Parr, T. J., "Enforcing Strict Model-View Separation in Template Engines", **Proceedings of the 13th international conference on World Wide Web**, ACM Press, 224-233, New York, USA, (2004).
- [11] *Template view* (Çevrimiçi), [http://www.phpwact.org/pattern/template\\_view](http://www.phpwact.org/pattern/template_view) (**Ziyaret Tarihi: 20 Ekim 2006**)
- [12] Hartmann, F., "An Architecture for an XML-Template Engine Enabling Safe Authoring". **DEXA Workshops**, 502-507, (2006).
- [13] *PHP- Hypertext Preprocessor* (Çevrimiçi), <http://www.php.net> (**Ziyaret Tarihi: 10 Eylül 2007**)

- [14] *World Wide Web Consortium page on XML* (Çevrimiçi), <http://www.w3c.org/TR/REC-XML> (**Ziyaret Tarihi: 18 Eylül 2006**)
- [15] R. Vidgen and S. Goodwin, “XML: what is it good for?,” *IEE Computing & Control Engineering Journal*, vol. June, pp. 119-124, (2000).
- [16] *Mysql Open Source Database* (Çevrimiçi), <http://www.mysql.com> (**Ziyaret Tarihi: 15 Ocak 2007**)
- [17] *Apache Http Server* (Çevrimiçi), <http://httpd.apache.org> (**Ziyaret Tarihi: 11 Eylül 2006**)
- [18] Gousios, G., and Spinellis, D., “A comparison of portable dynamic web content technologies for the apache web server”, In *Proceedings of the 3rd International System Administration and Networking Conference SANE 2002*, pages 103–119, May (2002).
- [19] Gurugé, A., “Corporate Portals Empowered with XML and Web Services”, *Digital Press*, ISBN 978-1555582807, (2002).
- [20] Boiko, B., “Content Management Bible”, Second Ed., *Wiley Publishing Inc.*, Indianapolis, Indiana, USA., ISBN: 0-7645-7371-3, (2005).
- [21] *History of Content Management Systems*, 2005, Cmswiki (Çevrimiçi), <http://www.cmswiki.com/tiki-index.php?page=HistoryOfCMS>, (**Ziyaret Tarihi: 15 Temmuz 2006**)
- [22] Nakano, R., “Web Content Management A Collaborative Approach”, *Addison-Wesley*. Boston, MA, USA. (2001).
- [23] Erdal, M., “Elektronik Ticarette Web Sitesi Üretimi ve Proje Yönetim Süreci”, *Active*, Yıl:5, Sayı:25, Temmuz-Ağustos, 46-52, (2002).
- [24] Stephen R., Fraser G., “Building a Content Management System”, *Apress*, New York, NY, USA., ISBN: 1590590244, (2002).
- [25] Pastore, S., “Web Content Management Systems: using Plone open source software to build a website for research institute needs”, *Digital Telecommunications*, ICDT apos;06. International Conference on Volume , Issue , 2006, Page(s):24 – 24, (2006).
- [26] Stuckenschmidt, H., van Harmelen, F., “Generating and Managing Metadata for Web-Based Information Systems”, *Knowledge-Based Systems*, 17(5-6), (2004).
- [27] David Bodoff, D., Ben-Menachem, M., Hung, K., C., P., “Web Metadata Standards: Observations and Prescriptions”, *IEEE Software*, Volume 22 , Issue 1, 78-85, (2005).

- [28] Cao, J., Chan, C., Chan, K., “Workflow analysis for web publishing using a sage-activity process model”, *Journal of Systems and Software*, Volume 76 , Issue 3 (June 2005), 221–235, (2005).
- [29] *KnowledgeHills* (Çevrimiçi), <http://erptoday.knowledgehills.com/CMS/Content-Management-Tutorial.aspx> (**Ziyaret Tarihi: 14 Şubat 2007**)
- [30] Sheth, A., “Managing Semantic Content for the Web”, *IEEE Internet Computing*, Vol. 6, Issue 4, pp. 80-87, July/Aug (2002).
- [31] Jose D., 2005, *Content Engineering – A Lifecycle Management Approach*, ITtoolbox Web Design, (Çevrimiçi), <http://research.ittoolbox.com/white-papers/datamgt/km/content-engineering-a-life-cycle-management-approach-3035>, <http://hosteddocs.ittoolbox.com/DJ042205.pdf>, (**Ziyaret Tarihi: 12 Mart 2006**)
- [32] *Seven Stages of the Content Lifecycle* , Cmsreview (Çevrimiçi), <http://www.cmsreview.com/Stages/> (**Ziyaret Tarihi: 12 Eylül 2007**)
- [33] *Solving the ECM Puzzle*, AIIM - The Enterprise Content Management Association (Çevrimiçi), <http://www.aiim.org/poster/puzzleposter.html> (**Ziyaret Tarihi: 9 Aralık 2006**)
- [34] Challenger J., Iyengar A., Witting K., Ferstat, C., Reed, P., “A Publishing System for Efficiently Creating Dynamic Web Content (2000)”, *proceedings of IEEE, INFOCOM*, (2000).
- [35] Krasnerand, G.E., Pope, S.T., “A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System”, *Parc Place Systems Inc*, MountainView, (1988).
- [36] Kerer, C., Kirda, E., “Layout, Content and Logic Separation in Web Engineering”, *3rd Workshop on Web Engineering, World Wide Web Conference (WWW9)*, Amsterdam, The Netherlands.
- [37] Parr, T. J., *About The String Template Template Engine*, StringTemplate (Çevrimiçi), <http://www.stringtemplate.org/about.html> (**Ziyaret Tarihi: 12 Eylül 2007**)
- [38] Fernandez, M., Florescu, D., Kang, J., Levy, A., Suciu D., “Catching the boat with Strudel: experience with a Web-site management system”, *SIGMOD*, 414-425, (1998).
- [39] *XML.com:Push, Pull, Next!*, <http://www.xml.com/pub/a/2005/07/06/tr.html> (**Ziyaret Tarihi: 6 Ekim 2006**)
- [40] Kirda, E., Kerer, C., “MyXML: An XML based template engine for the generation of flexible Web content”, *In Proceedings of WEBNET 2000*, San Antonio, Texas, USA, (2000).

- [41] *Patterns-CompositeView* (Çevrimiçi),  
<http://java.sun.com/blueprints/patterns/CompositeView.html>,  
**(Ziyaret Tarihi: 20 Ekim 2006)**
- [42] *Core J2EE Patterns - Composite View* (Çevrimiçi),  
<http://java.sun.com/blueprints/corej2eepatterns/Patterns/CompositeView.html>  
**(Ziyaret Tarihi: 20 Ekim 2006)**
- [43] *J2EE Design Patterns - Composite view* (Çevrimiçi),  
[http://www.allapplabs.com/j2ee\\_design\\_patterns/j2ee\\_design\\_patterns\\_composite\\_view.htm](http://www.allapplabs.com/j2ee_design_patterns/j2ee_design_patterns_composite_view.htm) **(Ziyaret Tarihi: 20 Ekim 2006)**
- [44] Bouras, C., Kounenis, G., Misedakis, I., “A Web Content Manipulation Technique Based On Page Fragmentation”, *Journal of Network and Computer Applications*, Volume 30, Issue 2, ISSN:1084-8045, 563-585, (2007).
- [45] Zdun, U., “Dynamically Generating Web Application Fragments From Page Templates”, *In Proceedings of Symposium of Applied Computing (SAC2002)*, Madrid, Spain, (2002).
- [46] Geary, D., *Web Application components made easy with Composite View* (Çevrimiçi), <http://www.javaworld.com/javaworld/jw-12-2001/jw-1228-jsptemplate.html> **(Ziyaret Tarihi: 20 Ekim 2006) [k49 ]**
- [47] Bouras, C., Konidaris, A., “Web Components: A Concept for Improving Personalization and Reducing User Perceived Latency on the World Wide Web”, *Proceedings of the 2nd International Conference on Internet Computing*, (238-244), Las Vegas, NV, June, (2001).
- [48] *Smarty* (Çevrimiçi), <http://smarty.php.net/> **(Ziyaret Tarihi: 20 Ekim 2006)**
- [49] Parr, T. J., “Web Application Internationalization And Localization In Action”, *Proceedings of the 6th international conference on Web engineering*, California, USA, (2006)
- [50] Berners-Lee, T., Hendler, J., Lassila, O., “The Semantic Web - A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities”, *Scientific American*, May, (2001)
- [51] Surjanto, B., Ritter, N., Loeser, H., “XML content management based on object relational database technology”. *In Proc. Int. Conf. on Web Information Systems Engineering (WISE)*, 64-73, (2000).
- [52] Nguyen, B., Abiteboul, S., Cobena, G., Preda, M., “Monitoring XML Data on the Web”, *In Proceedings of the ACM SIGMOD Conference 2001*, Santa Barbara, CA, USA, (2001).
- [53] Norrie, M. C., Palinginis, A., Signer, B., Content Publishing Framework for Interactive Paper Documents, *DocEng'05*, November 2-4, Bristol, UK, (2005).

[54] Kırbaş, İ., Yıldırım, M., “Dinamik Web Sitesi İçerik Yönetim Sistemi:DyNA”, *Ulusal Teknik Eğitim Mühendislik ve Eğitim Bilimleri Genç Araştırmacılar Sempozyumu UMES’07*, Cilt 1, s.62-66, Kocaeli, 20-22 Haziran (2007).

[55] Kırbaş, İ., *DyNA Türkçe Web Sitesi İçerik Yönetim Sistemi*, DyNA Web Sitesi İçerik Yönetimi (Çevrimiçi), <http://www.dyna.web.tr/index.php?id=104> (**Ziyaret Tarihi: 10 Aralık 2007**)

## **ÖZGEÇMİŞ**

1977 yılında Bulgaristan'ın Şumnu şehrinde doğdu. İlk, orta ve lise öğrenimini İstanbul'da tamamladı. 1995 yılında girdiği Kocaeli Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Öğretmenliği Bölümünden 2000 yılında dereceyle mezun oldu. 2000-2003 yılları arasında internet erişimi, finans ve iletişim sektörlerinde faaliyet gösteren farklı firmaların ar-ge bölümlerinde görev aldıktan sonra 2003 yılından beri MEB bünyesinde kadrolu bilgisayar öğretmeni olarak görev yapmakta olup evli ve bir çocuk babasıdır.