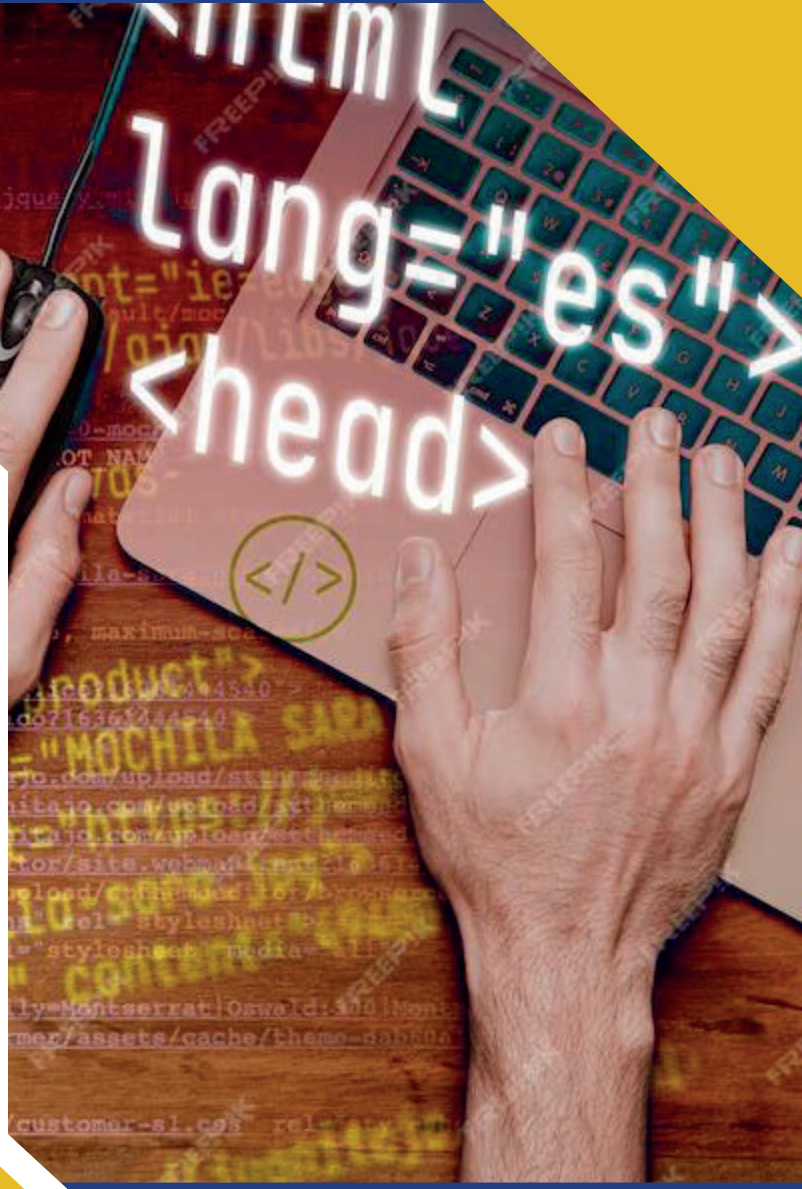


BÖLÜM 4

PHP'DE XML VE AJAX

ANAHTAR KAVRAMLAR

XML, XML Ayırıştırıcı, AJAX



ÖĞRENME HEDEFLERİ

- 1 XML ve AJAX ile ilgili temel kavramları bilme
- 2 XML ayrıştırıcılarını kullanarak XML analiz edebilme
- 3 AJAX ile veritabanına bağlanabilme
- 4 AJAX ile canlı arama yapabilme
- 5 AJAX ile anket hazırlayabilme

PHP XML

PHP’de XML konusunu anlayabilmek için önce XML nedir bilmek gerekmektedir. XML dili, web siteleri arasında paylaşım için verileri yapılandırmanın bir yoludur. RSS Akışları ve Podcast’ler gibi çeşitli web teknolojileri XML’de yazılmıştır. XML’in oluşturulması kolaydır. Ayrıca, kendi etiketlerinizi oluşturmanız dışında HTML’ye çok benzemektedir. PHP’de XML kullanmak için XML ayrıştırıcılarına ihtiyaç vardır.

XML Ayrıştırıcılar

PHP’de bir XML belgesini okumak, güncellemek, oluşturmak ve değiştirmek için bir XML ayrıştırıcı kullanılmaktadır. PHP’de iki ana XML ayrıştırıcı türü vardır:

- Ağaç Tabanlı Ayrıştırıcılar
- Olay Tabanlı Ayrıştırıcılar

Ağaç Tabanlı Ayrıştırıcılar

Ağaç tabanlı ayrıştırıcılar belgenin tamamını bellekte tutar ve XML belgesini Ağaç yapısına dönüştürür. İlgili ayrıştırıcı, belgenin tamamını analiz eder ve oluşturulan Ağaç öğelerine (DOM) erişim sağlar.

Bu tür ayrıştırıcı, daha küçük XML belgeleri için iyi bir seçenektir. Büyük XML belgeleri için tercih edildiğinde ise performans sorunlarına neden olur.

Ağaç tabanlı ayrıştırıcılara örnek:

- SimpleXML
- DOM

Olay Tabanlı Ayrıştırıcılar

Olay tabanlı ayrıştırıcılar Ağaç tabanlının aksine belgenin tamamını bellekte tutmaz; bunun yerine, her seferinde bir düğümde okur ve gerçek zamanlı olarak etkileşim kurmanıza olanak tanır. Bir sonraki düğüme geçtiğinizde ise eskisi atılmaktadır.

Bu tür ayrıştırıcı büyük XML belgeleri için uygundur. Daha hızlı ayrıştırır ve daha az bellek tüketir.

Olay tabanlı ayrıştırıcılara örnek:

- XMLReader
- XML Expat Parser

SimpleXML Ayrıştırıcısı

SimpleXML, XML verilerini kolayca işlememize ve almamıza olanak tanıyan bir PHP uzantısıdır. SimpleXML yukarıda da bahsedildiği üzere ağaç tabanlı bir ayrıştırıcıdır.

SimpleXML, XML belgesinin yapısı veya düzeni biliniyorsa, bir öğenin adını, niteliklerini ve metin içeriğini almanın kolay bir yolunu sağlamaktadır.

SimpleXML, bir XML belgesini, diziler ve nesneler koleksiyonu gibi yineleyebileceğiniz bir veri yapısına dönüştürür.

DOM veya Expat ayrıştırıcıyla karşılaştırıldığında SimpleXML, bir öğeden metin verilerini okumak için daha az kod satırı kullanır.

SimpleXML Ayrıştırıcısı Kurulumu

PHP 5 sürümünden itibaren SimpleXML işlevleri PHP çekirdeğinin bir parçası olarak karşımıza çıkmaktadır. Bu bağlamda ayrıştırıcının işlevlerini kullanmak için herhangi bir kurulum gerekmez.

PHP SimpleXML - Dizeden Okuma

PHP `simplexml_load_string()` işlevi bir dizeden XML verilerini okumak için kullanılır.

Aşağıda verilen kod bloğunda olduğu gibi XML verilerini içeren bir değişkenimiz olduğunu varsayalım:

```
$myXMLData =
"<?xml version='1.0' encoding='UTF-8'?>
<note>
<to>Öğrenci 1</to>
<from>ANKUZEF</from>
<heading>Hatırlatıcı</heading>
<body>Bu hafta kayıt olmayı unutmayınız!</body>
</note>";
```

Aşağıdaki örnek kod bloğunda da, bir dizeden XML verilerini okumak için `simplexml_load_string()` işlevinin nasıl kullanılacağı gösterilmektedir:

```
<?php
$myXMLData =
"<?xml version='1.0' encoding='UTF-8'?>
<note>
<to>Öğrenci 1</to>
<from>ANKUZEF</from>
<heading>Hatırlatıcı</heading>
<body>Bu hafta kayıt olmayı unutmayınız!</body>
</note>";

$xml=simplexml_load_string($myXMLData) or die("Error: hatası ile nesne oluşturulamadı");
print_r($xml);
?>
```

Yukarıdaki kodun çıktısı şöyle olacaktır:

SimpleXMLElement Object ([to] => Öğrenci 1 [from] => ANKUZEF [heading] => Hatırlatıcı [body] => Bu hafta kayıt olmayı unutmayınız!)

Belgeyi yüklerken tüm XML hatalarını almak ve ardından hataları yinlemek için de libxml işlevi kullanılmaktadır. Aşağıdaki örnek, bozuk bir XML dizesini yüklemeye çalışır:

```
<?php
libxml_use_internal_errors(true);
$myXMLData =
"<?xml version='1.0' encoding='UTF-8'?>
<document>
  <user>Öğrenci 1</wronguser>
  <email>ogrenci1@ankara.edu.tr</wrongemail>
</document>";

$xml = simplexml_load_string($myXMLData);
if ($xml === false) {
    echo "XML yükleme başarısız: ";
    foreach(libxml_get_errors() as $error) {
        echo "<br>", $error->message;
    }
} else {
    print_r($xml);
}
?>
```

Yukarıdaki kodun çıktısı şöyle olacaktır:

XML yükleme başarısız:

Opening and ending tag mismatch: user line 3 and wronguser

Opening and ending tag mismatch: email line 4 and wrongemail

Görüldüğü üzere 3. ve 4. Satırdaki user ve email etiketleri yanlış kapatılmış ve ilgili hata libxml_get_errors() işlevi ile ekrana yazdırılmıştır.

PHP SimpleXML - Dosyadan Okuma

PHP simplexml_load_file() işlevi bir dosyadan XML verilerini okumak için kullanılır.

Aşağıda verildiği gibi "note.xml " adında bir XML dosyamız olduğunu varsayalım:

```
$myXMLData =
"<?xml version='1.0' encoding='UTF-8'?>
<note>
<to>Öğrenci 1</to>
<from>ANKUZEF</from>
<heading>Hatırlatıcı</heading>
<body>Bu hafta kayıt olmayı unutmayınız!</body>
</note>";
```

Aşağıdaki kod bloğunda ise, bir dosyadan XML verilerini okumak için simplexml_load_file() işlevinin nasıl kullanılacağı verilmektedir:

```
<?php
$xml=simplexml_load_file("note.xml") or die("Error: hatası ile nesne oluşturulamadı");
print_r($xml);
?>
```

Yukarıdaki kodun çıktısı şöyle olacaktır:

SimpleXMLElement Object ([to] => Öğrenci 1 [from] => ANKUZEFE [heading] => Hatırlatıcı [body] => Bu hafta kayıt olmayı unutmayınız!)

PHP SimpleXML - Düğüm Değerlerini Alma

SimpleXML'in, XML verilerini kolayca işlememize ve almamıza olanak tanıyan bir PHP uzantısı olduğundan daha önce bahsetmiştik. İlgili ayrıştırıcı ile düğüm değerlerini almak için aşağıdaki gibi kod yazılmaktadır. Bu betik yukarıda daha önce bahsi geçen note.xml dosyasına erişmektedir:

```
<?php
$xml=simplexml_load_file("note.xml") or die("Error: hatası ile nesne oluşturulamadı");
echo $xml->to . "<br>";
echo $xml->from . "<br>";
echo $xml->heading . "<br>";
echo $xml->body;
?>
```

Yukarıdaki kodun çıktısı şöyle olacaktır:

Öğrenci 1
ANKUZEFE
Hatırlatıcı
Bu hafta kayıt olmayı unutmayınız!

Benzer şekilde bir de kitaplar için tutulan bir xml dosyamıza erişim sağlayalım. İlgili dosya aşağıdaki gibidir (books.xml):

```
<?xml version="1.0" encoding="utf-8"?>
<bookstore>
  <book category="Masaüstü">
    <title lang="en">Kitap 1</title>
    <author>Yazar 1</author>
    <year>2020</year>
    <price>130.00</price>
  </book>
  <book category="Mobil">
    <title lang="en">Kitap 2</title>
    <author>Yazar 2</author>
    <year>2022</year>
    <price>229.99</price>
  </book>
  <book category="WEB">
    <title lang="en-us">Kitap 3</title>
    <author>Yazar 3</author>
    <year>2024</year>
    <price>249.99</price>
  </book>
  <book category="WEB">
    <title lang="en-us">Kitap 4</title>
    <author>Yazar 4</author>
    <year>2018</year>
    <price>139.95</price>
  </book>
</bookstore>
```

PHP SimpleXML - Belirli Öğelerin Düğüm Değerlerini Alma

Aşağıdaki kod parçası ile "books.xml" dosyasının birinci ve ikinci <book> öğelerindeki <title> öğesinin düğüm değerini alınmaktadır:

```
<?php
$xml=simplexml_load_file("books.xml") or die("Error: hatası ile nesne oluşturulmadı");
echo $xml->book[0]->title . "<br>";
echo $xml->book[1]->title;
?>
```

Yukarıdaki kodun çıktısı:

Kitap 1
Kitap 2

şeklindedir.

PHP SimpleXML - Düğüm Değerlerini Alma – Döngü Kullanımı

Aşağıdaki örnek, "books.xml" dosyasındaki tüm <book> öğeleri arasında döngü yapar ve <title>, <author>, <year> ve <price> öğelerinin düğüm değerlerini alır:

```
<?php
$xml=simplexml_load_file("books.xml") or die("Error: hatası ile nesne oluşturulmadı");
foreach($xml->children() as $books) {
    echo $books->title . ", ";
    echo $books->author . ", ";
    echo $books->year . ", ";
    echo $books->price . "<br>";
}
?>
```

Yukarıdaki kodun çıktısı şöyle olacaktır:

Kitap 1, Yazar 1, 2020, 130.00
Kitap 2, Yazar 2, 2022, 229.99
Kitap 3, Yazar 3, 2024, 249.99
Kitap 4, Yazar 4, 2018, 139.95

PHP SimpleXML - Öznitelik Değerlerini Alma

SimpleXML ile ilk <book> öğesinin "category" niteliğinin öznitelik değerini ve ikinci <book> öğesindeki <title> öğesinin "lang" niteliğinin öznitelik değerini almak istersek kod bloğumuz aşağıdaki gibi olmalıdır:

```
<?php
$xml=simplexml_load_file("books.xml") or die("Error: hatası ile nesne oluşturulmadı");
echo $xml->book[0]['category'] . "<br>";
echo $xml->book[1]->title['lang'];
?>
```

Yukarıdaki kod çalıştırıldığında,

Masaüstü
en

ekrana yazar.

PHP SimpleXML - Öznitelik Değerlerini Alma – Döngü

Aşağıdaki örnek, "books.xml" dosyasındaki <title> öğelerinin nitelik değerlerini bir döngü içerisinde almaktadır:

```
<?php
$xml=simplexml_load_file("books.xml") or die("Error: hatası ile nesne oluşturulmadı");
foreach($xml->children() as $books) {
    echo $books->title['lang'];
    echo "<br>";
}
?>
```

Yukarıdaki kodun çıktısı:

en
en
en-us
en-us

şeklinde olacaktır.

PHP XML Expat Ayırıştırıcı

XML Expat Ayırıştırıcısı, SimpleXML de olduğu gibi XML belgelerinin PHP'de işlenmesini mümkün kılar. Expat ayırıştırıcısı olay tabanlı bir ayırıştırıcıdır.

Aşağıdaki XML ögesine bakın:

```
<from>ANKUZEF</from>
```

Olay tabanlı bir ayırıştırıcı, yukarıdaki XML'i üç olaydan oluşan bir dizi olarak bildirir:

Başlangıç ögesi: from
CDATA bölümünü başlat, değer: ANKUZEF
Kapanış ögesi: from

XML Expat Ayırıştırıcı işlevleri PHP çekirdeğinin bir parçasıdır. Bu işlevleri kullanmak için herhangi bir kurulumu gerek yoktur.

XML Expat Ayırıştırıcısını Başlatma

PHP'de XML Expat Ayırıştırıcısını başlatmak, farklı XML olayları için bazı işleyiciler tanımlamak ve ardından XML dosyasını ayırtırmak istiyoruz. Bu duruma uygun kod parçacağı aşağıdaki gibi olacaktır:

```

<?php
//XML ayrıştırıcıyı başlatma
$parser=xml_parser_create();
//Bir ögenin başlangıcında kullanılacak işlev
function start($parser,$element_name,$element_attrs) {
    switch($element_name) {
        case "NOTE": echo "-- Note --<br>";
        break;
        case "TO": echo "To: ";
        break;
        case "FROM": echo "From: ";
        break;
        case "HEADING": echo "Heading: ";
        break;
        case "BODY": echo "Message: ";
        break;
    }
}
//Bir ögenin sonunda kullanılacak işlev
function stop($parser,$element_name) { echo "<br>"; }
//Karakter verilerini bulurken kullanılacak işlev
function char($parser,$data) { echo $data; }
//Öge işleyicisini belirtme
xml_set_element_handler($parser,"start","stop");
//Veri işleyiciyi belirtme
xml_set_character_data_handler($parser,"char");
//XML dosyasını açma
$fp=fopen("note.xml","r");
//Veriyi okuma
while ($data=fread($fp,4096)) {
    xml_parse($parser,$data,feof($fp)) or
    die (sprintf("XML Error: %s at line %d",
    xml_error_string(xml_get_error_code($parser)),
    xml_get_current_line_number($parser)));
}
//XML ayrıştırıcıyı serbest bırakma
xml_parser_free($parser);
?>

```

Yukarıdaki kod bloğunun aşağıdaki gibi bir sıralaması olmaktadır:

1. XML ayrıştırıcısını xml_parser_create() işlev ile başlatın
2. Farklı olay işleyicileriyle kullanılacak işlevler oluşturun
3. Ayrıştırıcı açılış ve kapanış etiketleriyle karşılaştığında hangi işlevin yürütüleceğini belirtmek için xml_set_element_handler() işlevi ekleyin
4. Ayrıştırıcı karakter verileriyle karşılaştığında hangi işlevin yürütüleceğini belirtmek için xml_set_character_data_handler() işlevi ekleyin
5. "note.xml" dosyasını xml_parse() fonksiyonuyla ayrıştırın
6. Hata durumunda, XML hatasını metinsel açıklamaya dönüştürecek xml_error_string() işlevi ekleyin
7. xml_parser_create() işleviyle ayrılan belleği serbest bırakmak için xml_parser_free() işlevini çağırın

PHP XML DOM Ayırıştırıcı

DOM ayırıştırıcısı ağaç tabanlı bir ayırıştırıcıdır.

Aşağıdaki XML belge bölümüne bakın:

```
<?xml version="1.0" encoding="UTF-8"?>
<from>ANKUZEF</from>
```

DOM yukarıdaki XML'i bir ağaç yapısı olarak görür:

Seviye 1: XML Belgesi

Seviye 2: Kök öge: <from>

Seviye 3: Metin ögesi: "ANKUZEF"

Kurulum

DOM ayırıştırıcı işlevleri PHP çekirdeğinin bir parçasıdır. Bu işlevleri kullanmak için herhangi bir kurulumla gerek yoktur.

XML'i Yükleme ve Çıktı

XML ayırıştırıcısını başlatmak, xml'i yüklemek ve çıktısını almak için,

```
<?php
$xmlDoc = new DOMDocument();
$xmlDoc->load("note.xml");

print $xmlDoc->saveXML();
?>
```

Yukarıdaki kodun çıktısı şöyle olacaktır:

Öğrenci 1 ANKUZEF Hatırlatıcı Bu hafta kayıt yapmayı unutmayınız!

Ayrıca yukarıdaki kod bloğunu herhangi tarayıcı penceresinde sağ tıklayıp “kaynağı görüntüle” seçildiğinde,

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
<to>Öğrenci 1</to>
<from>ANKUZEF</from>
<heading>Hatırlatıcı</heading>
<body>Bu hafta kayıt yapmayı unutmayınız!</body>
</note>
```

şeklinde bir XML yapısı gibi gözüktüğü görülecektir. Bu kod, bir DOM Document-Object oluşturur ve XML'i "note.xml"den ona yükler.

Daha sonra saveXML() işlevi ile dâhili XML belgesini bir dizeye yerleştirir, böylece çıktısını alabiliriz.

XML'de Döngü Yapma

XML DOM ayrıştırıcısını başlatmak, XML'i yüklemek ve <note> ögesinin tüm öğeleri arasında döngü yapmak istediğimizde,

```
<?php
$xmlDoc = new DOMDocument();
$xmlDoc->load("note.xml");

$x = $xmlDoc->documentElement;
foreach ($x->childNodes AS $item) {
    print $item->nodeName . " = " . $item->nodeValue . "<br>";
}
?>
```

şeklinde kod yazmamız gerekmektedir. foreach ile DOM ayrıştırıcısı tabanlı öğelerin isimleri ve değerleri çekilmektedir. Yukarıdaki kodun çıktısı şöyle olacaktır:

```
#text =
to = Öğrenci 1
#text =
from = ANKUZEFE
#text =
heading = Hatırlatıcı
#text =
body = Bu hafta kayıt olmayı unutmayınız!
#text =
```

Yukarıdaki örnekte her öge arasında boş metin düğümleri olduğunu görüyorsunuz.

XML oluşturulurken genellikle düğümler arasında boşluklar bulunur. XML DOM ayrıştırıcısı bunları sıradan öğeler olarak ele alır. Ancak sizler bunların farkında değilseniz bazen sorunlara neden olmaktadır.

PHP - AJAX

AJAX, bir web sayfasının bazı bölümlerinin, tüm sayfayı yeniden yüklemekten güncellenmesiyle ilgilidir. Bu kısımda AJAX'ın ne olduğu ve PHP'deki kullanımını göreceğiz.

AJAX nedir?

AJAX = Eşzamansız JavaScript ve XML (Asynchronous JavaScript and XML).

AJAX hızlı ve dinamik web sayfaları oluşturmaya yönelik bir tekniktir.

AJAX, arka planda sunucuyla küçük miktarlarda veri alışverişi yaparak web sayfalarının eşzamansız olarak güncellenmesine olanak tanır. Bu, bir web sayfasının bazı kısımlarını, sayfanın tamamını yeniden yüklemekten güncellenmenin mümkün olduğu anlamına gelir.

Klasik web sayfaları (AJAX kullanmayanlar), içeriğin değişmesi durumunda sayfanın tamamını yeniden yüklemelidir.

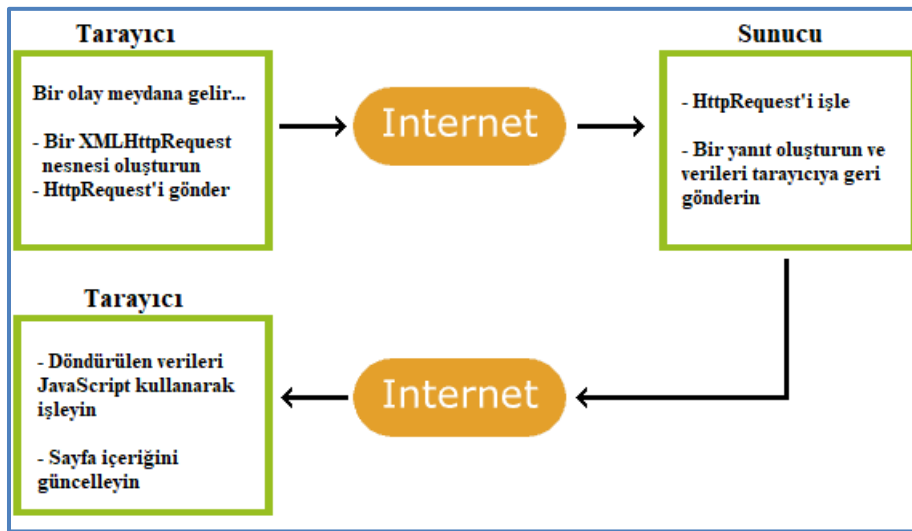
AJAX kullanan uygulamalara örnekler: Google Haritalar, Gmail, Youtube ve Facebook sekmeleri.

AJAX Nasıl Çalışır?

AJAX internet standartlarını temel alır ve aşağıdakilerin bir birleşim unu kullanır:

- XMLHttpRequest nesnesi (bir sunucuyla eşzamansız olarak veri alışverişi yapmak için)
- JavaScript/DOM (bilgileri görüntülemek/etkileşimde bulunmak için)
- CSS (verilere stil vermek için)
- XML (genellikle veri aktarma formatı olarak kullanılır)

AJAX uygulamaları tarayıcı ve platformdan bağımsızdır. Şekil 1 Sunucu ile İstemcinin tarayıcısı arasında gerçekleşen AJAX yapısını göstermektedir:



Şekil 1. AJAX Çalışma Prensipleri

AJAX daha etkileşimli uygulamalar oluşturmak için kullanılır.

AJAX PHP

Aşağıdaki örnek, kullanıcı giriş alanına karakterleri yazarken (onkeyup) bir web sayfasının sunucusuyla nasıl iletişim kurabileceğini göstermektedir. Bunun için arama işleminin yapıldığı index.php ve arama işlemi için kullanılan gethint.php dosyaları aşağıdaki gibidir:

index.php dosyası;

```

<html>
<head><meta charset="UTF-8">
<script>
function showHint(str) {
  if (str.length == 0) {
    document.getElementById("txtHint").innerHTML = "";
    return;
  } else {
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
      if (this.readyState == 4 && this.status == 200) {
        document.getElementById("txtHint").innerHTML = this.responseText;
      }
    };
    xmlhttp.open("GET", "gethint.php?q=" + str, true);
    xmlhttp.send();
  }
}
</script>
</head>
<body>
<p><b>Aşağıdaki girdi alanına bir isim yazınız</b></p>
<form action="">
  <label for="fname">İsim:</label>
  <input type="text" id="fname" name="fname" onkeyup="showHint(this.value)">
</form>
<p>Öneriler: <span id="txtHint"></span></p>
</body>
</html>

```

şeklinde. Çıktısı ise aşağıdaki gibidir:

Aşağıdaki girdi alanına bir isim yazınız

İsim:

Öneriler: samet, saitcan, sıla, sudenaz

Burada XMLHttpRequest, web sayfalarının sunucu ile iletişim kurmasını sağlayan bir API'dir.

this.readyState XMLHttpRequest nesnesinin durumunu belirtir ve 0 ile 4 arasında bir tamsayı değeri alır. Bu değerler şunları ifade eder:

- 0: İstek henüz başlatılmadı
- 1: Bağlantı kuruldu
- 2: İstek alındı
- 3: İstek işleniyor
- 4: İstek tamamlandı ve yanıt alındı

this.status ise sunucudan dönen HTTP yanıt kodunu belirtir. Genellikle, 200 bir "Başarılı" yanıt kodunu ve isteğin başarılı olduğunu gösterir.

Yani, this.readyState == 4 && this.status == 200 ifadesi, bir XMLHttpRequest nesnesinin durumunun 4 (tamamlanmış) ve sunucudan gelen yanıt kodunun 200 (başarılı) olduğu durumu kontrol eder. Bu durum genellikle bir AJAX isteğinin başarıyla tamamlandığını gösterir. Gelen istek gethint.php dosyasına q parametresi ile *gethint.php?q=" + str* şeklinde gönderilmektedir. str değişkeni giriş alanına yazılan değerleri tutmaktadır.

gethint.php dosyası;

```
<?php
// İsimler ile bir dizi
$a[]="samet";
$a[]="saitcan";
$a[]="s[ı]la";
$a[]="sudenaz";
// URL'den q parametresini alma
$q = $_REQUEST["q"];
$hint = "";
// $q "" ögesinden farklıysa dizideki tüm ipuçlarını arayın
if ($q != "") {
    $q = strtolower($q);
    $len=strlen($q);
    foreach($a as $name) {
        if (stristr($q, substr($name, 0, $len))) {
            if ($hint == "") { $hint = $name; }
            else { $hint .= ", $name"; }
        }
    }
}
// Hiçbir ipucu bulunamazsa "öneri yok" çıktısı alın
echo $hint == "" ? "öneri yok" : $hint;
?>
```

AJAX Veritabanı

AJAX, bir veritabanıyla etkileşimli iletişim için kullanılabilir. Aşağıdaki örnek, bir web sayfasının AJAX ile bir veritabanından nasıl bilgi alabileceğini gösterecektir. Bunun için ajax_demo isimli veritabanı ve user isimli aşağıdaki gibi bir tabloya ihtiyaç vardır:

user tablosu;

id	ad	soyad	yas	sehir	meslek
1	isim1	soyad1	30	Ankara	Öğretmen
2	isim2	soyad2	20	İzmir	Mühendis
3	isim3	soyad3	25	İstanbul	Tekniker

Veritabanından Ajax ile verilere erişmek için index.php ve uye.php olmak üzere iki dosya daha oluşturulmalıdır.

index.php dosyası;

```

<html>
<head>
<script>
function showUser(str) {
  if (str == "") {
    document.getElementById("txtHint").innerHTML = "";
    return;
  } else {
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
      if (this.readyState == 4 && this.status == 200) {
        document.getElementById("txtHint").innerHTML = this.responseText;
      }
    };
    xmlhttp.open("GET","uye.php?q="+str,true);
    xmlhttp.send();
  }
}
</script>
</head>
<body>
<form>
<select name="users" onchange="showUser(this.value)">
  <option value="">Kişi seç:</option>
  <option value="1">isim1 soyisim1</option>
  <option value="2">isim2 soyisim2</option>
  <option value="3">isim3 soyisim3</option>
</select>
</form>
<br>
<div id="txtHint"><b>Kişinin bilgisi burada listelenmektedir:</b></div>
</body>
</html>

```

Yukarıdaki kod bloğunda, kullanıcı açılır listeden bir kişiyi seçtiğinde, çağrılan showUser() isimli bir işlev yürütülmektedir. İlgili işlev onchange olayı tarafından tetiklenmektedir.

onchange olayı, bir HTML form elemanının değeri değiştiğinde tetiklenen bir JavaScript olayıdır. Bu olay genellikle <input>, <select>, ve <textarea> gibi form elemanlarında kullanılır.

Örneğin, bir <select> elemanında seçilen öğe değiştirildiğinde veya bir <input type="checkbox"> veya <input type="radio"> elemanındaki seçim değiştirildiğinde onchange olayı devreye girmektedir.

JavaScript'te bu olaya bir işlev atanarak, kullanıcı bir değer seçtiğinde veya girdiğinde yapılması gereken işlemleri tanımlayabilirsiniz. Örneğin, bir seçim yapıldığında bir fonksiyon çağrılabilir veya belirli bir kontrolün durumu değiştiğinde bir işlem gerçekleştirilebilir.

index.php dosyasının çıktısı aşağıdaki gibidir:

isim1	soyisim1	ad	Yaş	Şehir	Meslek
isim1	soyad1	30	Ankara	Öğretmen	

Burada öncelikle bir kişinin seçilip seçilmediği kontrol edilmektedir. Hiç kimse seçilmediyse (str == "") txtHint id'li div ögesi temizlenmektedir (document.getElementById("txtHint").innerHTML = "");. Eğer bir kişi seçilirse XMLHttpRequest nesnesi oluşturulmakta ve seçim URL'ye q parametresi ile gönderilmektedir.

uye.php dosyası ise aşağıdaki gibi olup seçilen kişi bir tablo içerisinde listelenmektedir.

```
<html>
<head>
    <style>table,tr,th,td{
        border:1px solid black;
        border-collapse:collapse;
        text-align:left;
        padding:15px;}</style>
</head>
<body>
<?php
$q = intval($_GET['q']);
$con = mysqli_connect('localhost','root','');
if (!$con) {
    die('Bağlantı kurulmadı: ' . mysqli_error($con));
}
mysqli_select_db($con,"ajax_demo");
$sql="SELECT * FROM user WHERE id = '". $q ."'";
$result = mysqli_query($con,$sql);
echo "<table>
<tr>
<th>Ad</th><th>Soyad</th><th>Yaş</th><th>Şehir</th><th>Meslek</th>
</tr>";
while($row = mysqli_fetch_array($result)) {
    echo "<tr>";
    echo "<td>" . $row['ad'] . "</td>";
    echo "<td>" . $row['soyad'] . "</td>";
    echo "<td>" . $row['yas'] . "</td>";
    echo "<td>" . $row['sehir'] . "</td>";
    echo "<td>" . $row['meslek'] . "</td>";
    echo "</tr>";
}
echo "</table>";
mysqli_close($con);
?>
</body></html>
```

AJAX XML

AJAX, bir XML dosyasıyla etkileşimli iletişim için kullanılabilir. Aşağıdaki örnek, bir web sayfasının AJAX ile bir XML dosyasından nasıl bilgi getirebileceğini gösterecektir. Bu bağlamda uygulama için cd_catalog.xml isimli sanatçı albümlerinin tutulduğu bir dosya ve XML analizini yapabilmek adına index.php, getcd.php isimli dosyalar oluşturulmaktadır.

cd_catalog.xml dosyası,

```

<?xml version="1.0" encoding="utf-8"?>
<KATALOG>
  <CD>
    <ALBUM>Başlık 1</ALBUM>
    <ARTIST>Sanatçı 1</ARTIST>
    <ULKE>Ülke 1</ULKE>
    <SIRKET>Şirket 1</SIRKET>
    <FIYAT>Fiyat 1</FIYAT>
    <YIL>Yıl 1</YIL>
  </CD>
  <CD>
    <ALBUM>Başlık 2</ALBUM>
    <ARTIST>Sanatçı 2</ARTIST>
    <ULKE>Ülke 2</ULKE>
    <SIRKET>Şirket 2</SIRKET>
    <FIYAT>Fiyat 2</FIYAT>
    <YIL>Yıl 2</YIL>
  </CD>
  <CD>
    <ALBUM>Başlık 3</ALBUM>
    <ARTIST>Sanatçı 3</ARTIST>
    <ULKE>Ülke 3</ULKE>
    <SIRKET>Şirket 3</SIRKET>
    <FIYAT>Fiyat 3</FIYAT>
    <YIL>Yıl 3</YIL>
  </CD>
</KATALOG>

```

index.php dosyası,

```

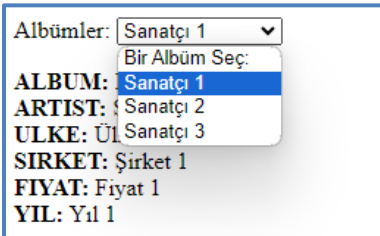
<html>
<head>
<script>
function showCD(str) {
  if (str=="") {
    document.getElementById("txtHint").innerHTML="";
    return;
  }
  var xmlhttp=new XMLHttpRequest();
  xmlhttp.onreadystatechange=function() {
    if (this.readyState==4 && this.status==200) {
      document.getElementById("txtHint").innerHTML=this.responseText;
    }
  }
  xmlhttp.open("GET","getcd.php?q="+str,true);
  xmlhttp.send();
}
</script>
</head>
<body>
<form>
Albümler:
<select name="cds" onchange="showCD(this.value)">
  <option value="">Bir Albüm Seç:</option>
  <option value="Sanatçı 1">Sanatçı 1</option>
  <option value="Sanatçı 2">Sanatçı 2</option>
  <option value="Sanatçı 3">Sanatçı 3</option>
</select>
</form>
<div id="txtHint"><b>Albüm bilgisi burada listelenecektir...</b></div>
</body>
</html>

```

getcd.php dosyası,

```
<?php
$q=$_GET["q"];
$xmlDoc = new DOMDocument();
$xmlDoc->load("cd_catalog.xml");
$x=$xmlDoc->getElementsByTagName('ARTIST');
for ($i=0; $i<=$x->length-1; $i++) {
    //Yalnızca öge düğümlerini işle
    if ($x->item($i)->nodeType==1) {
        if ($x->item($i)->childNodes->item(0)->nodeValue == $q) {
            $y=($x->item($i)->parentNode);
        }
    }
}
$cd=($y->childNodes);
for ($i=0;$i<$cd->length;$i++) {
    //Yalnızca öge düğümlerini işle
    if ($cd->item($i)->nodeType==1) {
        echo("<b>" . $cd->item($i)->nodeName . ":</b> ");
        echo($cd->item($i)->childNodes->item(0)->nodeValue);
        echo("<br>");
    }
}
?>
```

şeklindedir. Uygulamanın çıktısı ise,



Kullanıcı yukarıdaki açılır listeden bir CD seçtiğinde "showCD()" adı verilen bir işlev yürütülür. İşlev "onchange" olayıyla tetiklenmektedir. showCD() işlevi aşağıdakileri yapar:

- Bir CD'nin seçilip seçilmediğini kontrol eder,
- XMLHttpRequest nesnesi oluşturur,
- Sunucu yanıtı hazır olduğunda yürütülecek işlevi belirler,
- İsteği sunucudaki bir dosyaya gönderir,
- URL'ye (getcd.php) açılır listeden seçilen değeri q isimli bir parametre ile gönderir.

getcd.php içindeki betik ile "cd_catalog.xml" adlı bir XML belgesi yüklenmekte ve XML dosyasında bir sorgu çalıştırılmaktadır. Sorgunun sonucu HTML olarak dönmektedir.

Albüm sorgusu JavaScript'ten PHP sayfasına gönderildiğinde aşağıdakiler gerçekleşir:

- PHP bir XML DOM nesnesi oluşturur,
- JavaScript'ten gönderilen <artist> etiketi ile eşleşen tüm öğeleri bulur,

- Albüm bilgilerinin çıktısı alınır. Yani "txtHint" id'li div ögesine gönderilir.

AJAX Canlı Arama

AJAX, daha kullanıcı dostu ve etkileşimli aramalar oluşturmak için kullanılabilir. Aşağıdaki örnek, yazarken arama sonuçlarını alacağınız canlı bir aramayı göstermektedir.

Canlı aramanın geleneksel aramaya kıyasla birçok avantajı vardır:

- Siz yazdıkça sonuçlar gösterilir
- Siz yazmaya devam ettikçe sonuçlar daralır
- Sonuçlar çok dar olursa daha geniş bir sonuç görmek için karakterleri silmek gerekir.

İlgili uygulama için, links.xml isimli dosyada ANKUZE'ye bağlı Açıköğretim Lisans Programlarının isimleri ve url'leri tanımlanmaktadır. İlâveten index.php ve livesearch.php isimli dosyalara ihtiyaç duyulmaktadır.

links.xml dosyası,

```
<?xml version="1.0" encoding="utf-8"?>
<sayfalar>
  <baglanti>
    <baslik>Elektronik Ticaret ve Yönetimi Açıköğretim Lisans Programı</baslik>
    <url>https://ankuzef.ankara.edu.tr/programlar/acikogretim-programlari/
    lisans-programlari/47-ankuzef/programlar/acikogretim/lisans-programlari/
    310-elektronik-ticaret-ve-yonetimi-ac-koegretim-lisans-program</url>
  </baglanti>
  <baglanti>
    <baslik>Yönetim Bilişim Sistemleri Açıköğretim Lisans Programı</baslik>
    <url>https://ankuzef.ankara.edu.tr/programlar/acikogretim-programlari/
    lisans-programlari/47-ankuzef/programlar/acikogretim/lisans-programlari/
    311-yonetim-bilisim-sistemleri-ac-koegretim-lisans-program</url>
  </baglanti>
</sayfalar>
```

index.php dosyası,

```
<html>
<head>
<script>
function showResult(str) {
    if (str.length==0) {
        document.getElementById("livesearch").innerHTML="";
        document.getElementById("livesearch").style.border="0px";
        return;
    }
    var xmlhttp=new XMLHttpRequest();
    xmlhttp.onreadystatechange=function() {
        if (this.readyState==4 && this.status==200) {
            document.getElementById("livesearch").innerHTML=this.responseText;
            document.getElementById("livesearch").style.border="1px solid #A5ACB2";
        }
    }
    xmlhttp.open("GET","livesearch.php?q="+str,true);
    xmlhttp.send();
}
</script>
</head>
<body>
<form>
<input type="text" size="30" onkeyup="showResult(this.value)">
<div id="livesearch"></div>
</form>
</body>
</html>
```

livesearch.php dosyası,

```

<?php
$xmlDoc=new DOMDocument();
$xmlDoc->load("links.xml");
$x=$xmlDoc->getElementsByTagName('baglanti');
// URL'den q parametresini al
$q=$_GET["q"];
// uzunluđu q>0 ise xml dosyasındaki tüm bağlantıları arayın
if (strlen($q)>0) {
    $hint="";
    for($i=0; $i<($x->length); $i++) {
        $y=$x->item($i)->getElementsByTagName('baslik');
        $z=$x->item($i)->getElementsByTagName('url');
        if ($y->item(0)->nodeType==1) {
            // arama metniyle eşleşen bir bağlantı bulun
            if (strstr($y->item(0)->childNodes->item(0)->nodeValue,$q)) {
                if ($hint=="") {
                    $hint="<a href=' " .
                        $z->item(0)->childNodes->item(0)->nodeValue .
                        "' target='_blank'>" .
                        $y->item(0)->childNodes->item(0)->nodeValue . "</a>";
                } else {
                    $hint=$hint . "<br /><a href=' " .
                        $z->item(0)->childNodes->item(0)->nodeValue .
                        "' target='_blank'>" .
                        $y->item(0)->childNodes->item(0)->nodeValue . "</a>";
                }
            }
        }
    }
}
// İpucu bulunamazsa çıktıyı "öneri yok" olarak ayarlayın
// veya doğru değerlere
if ($hint=="") { $response="öneri yok"; }
else { $response=$hint; }
// yanıtı göster
echo $response;
?>

```

Uygulamanın ara yüzü ise aşağıdaki gibidir:

yönetim bil
Yönetim Bilişim Sistemleri Açıköğretim Lisans Programı

Yukarıdaki giriş alanında bir ANKUZEF Açıköğretim lisans programı sayfası aranmaktadır. Gerekli sonuçlar bir XML dosyasında (links.xml) bulunmaktadır. Bu örneği küçük ve basit hale getirmek için yalnızca iki sonuç mevcuttur. Daha fazla sonuç için xml dosyasına daha fazla kayıt girilmesi gerekir.

index.php dosyasında tanımlı <input> ögesine bir karakter yazdığında "showResult()" isimli işlev yürütülür. İşlev "onkeyup" olayıyla tetiklenir. Giriş alanı boşsa (str.length==0), işlev canlı arama yer tutucusunun içeriğini temizler ve işlevden çıkar.

Giriş alanı boş değilse showResult() işlevi aşağıdakileri yürütür:

- XMLHttpRequest nesnesi oluşturur,
- Sunucu yanıtı hazır olduğunda yürütülecek işlevi oluşturur,
- İsteği sunucudaki bir dosyaya gönderir,
- URL'ye (giriş alanının içeriğiyle birlikte) bir parametre (q) ekler.

"livesearch.php" dosyasındaki kaynak kod ise arama dizesiyle eşleşen başlıkları bir XML dosyasında arar ve sonucu döndürür. JavaScript'ten gönderilen herhangi bir metin varsa (strlen(\$q) > 0), aşağıdakiler gerçekleşir:

- XML dosyasını yeni bir XML DOM nesnesine yükler,
- JavaScript'ten gönderilen metindeki eşleşmeleri bulmak için tüm <baslik> öğeleri arasında dolaşır,
- "\$response" değişkeninde doğru URL'yi ve başlığı ayarlar. Birden fazla eşleşme bulunursa tüm eşleşmeler değişkene eklenir,
- Hiçbir eşleşme bulunamazsa \$response değişkeni "öneri yok" olarak ayarlanır.

AJAX Anket

Buradaki örnek, sonucun yeniden yüklenmeden gösterildiği bir anket uygulamasıdır. Kullanıcı seçeneklerden birini seçtiğinde "getVote()" adı verilen bir işlev yürütülür. İşlev "onclick" olayı tarafından tetiklenmektedir. Uygulama için index.php ve poll_vote.php isimli iki temel dosya kullanılmaktadır. Anket sonuçları ise php kodları çalıştırıldığında poll_result.txt dosyasına kaydedilmektedir. İlavenen anket sonuçlarına görsellik katmak adına poll.gif isimli bir görüntü dosyasına ihtiyaç vardır.

index.php dosyası,

```
<html>
<head>
<script>
function getVote(int) {
    var xmlhttp=new XMLHttpRequest();
    xmlhttp.onreadystatechange=function() {
        if (this.readyState==4 && this.status==200) {
            document.getElementById("poll").innerHTML=this.responseText;
        }
    }
    xmlhttp.open("GET","poll_vote.php?vote="+int,true);
    xmlhttp.send();
}
</script>
</head>
<body>
<div id="poll">
<h3>Şu ana kadar PHP ve AJAX'ı beğendiniz mi?</h3>
<form>
Evet: <input type="radio" name="vote" value="0" onclick="getVote(this.value)"><br>
Hayır: <input type="radio" name="vote" value="1" onclick="getVote(this.value)">
</form>
</div>
</body>
</html>
```

poll_vote.php dosyası,

```

<?php
$vote = $_REQUEST['vote'];
// metin dosyasının içeriğini al
$filename = "poll_result.txt";
$content = file($filename);
// içeriği diziye koy
$array = explode("||", $content[0]);
$yes = $array[0];
$no = $array[1];
if ($vote == 0) { $yes = (int)$yes + 1; }
if ($vote == 1) { $no = (int)$no + 1; }
// txt dosyasına oyları ekleme
$insertvote = $yes."||".$no;
$fp = fopen($filename,"w");
fputs($fp,$insertvote);
fclose($fp);
?>
<h2>Anket Sonuçları</h2>
<table>
<tr>
<td>Evet:</td>
<td>'
height='20'>
<?php echo(100*round((int)$yes/((int)$no+(int)$yes),2)); ?>%
</td>
</tr>
<tr>
<td>Hayır:</td>
<td>'
height='20'>
<?php echo(100*round((int)$no/((int)$no+(int)$yes),2)); ?>%
</td>
</tr>
</table>

```

Uygulama çalıştırıldığında,

Şu ana kadar PHP ve AJAX'ı beğendiniz mi?

Evet: ☐
Hayır: ☐

ekrana oylama anketi gelir. Seçeneklerden hangisinin işaretlendiğine göre, evet'lerin sayısı toplam seçim sayısına ve hayır'ların sayısı toplam seçim sayısına bölünerek yüzde değerler hesaplanmaktadır.

Anket Sonuçları:

Evet:  75%
Hayır:  25%

getVote() işlevi aşağıdakileri yapar:

- XMLHttpRequest nesnesi oluşturur,
- Sunucu yanıtı hazır olduğunda yürütülecek işlevi oluşturur,
- İsteği sunucudaki bir dosyaya gönderir,
- URL'ye (poll_vote.php) evet veya hayır seçeneğinin değeriyle birlikte bir parametre (vote) gönderir.

BÖLÜM ÖZETİ

Bu bölümde X emel ve ajax'ın PHP ile kullanımı üzerine bilgiler verilmektedir. X emel, verilerin yapısal olarak organize edildiği bir formattır ve PHP'de bu verileri işlemek için bir dizi X emel ayrıştırıcı bulunur. Bu ayrıştırıcılar sayesinde X emel belgelerindeki verileri kolayca okuyabilir, değiştirebilir ve işleyebiliriz. X emel belgesini yükleyerek, X Path veya dom gibi yöntemlerle belirli öğeleri seçebilir, veri çekebilir ve manipüle edebiliriz. Simple X emel, dom Dokument ve X emel Reader gibi PHP'nin yerleşik sınıfları, X emel belgeleriyle etkileşim sağlamak için sıkça kullanılan araçlardır. ajax ise Asenkron Java Script ve X emel'in kısaltmasıdır ve web uygulamalarında kullanıcı etkileşimini geliştirmek için kullanılan bir tekniktir. Sayfayı yenilemeden sunucu ile veri alışverişi yapmayı sağlar, böylece kullanıcılar daha hızlı ve daha akıcı bir deneyim yaşarlar. ajax, tarayıcı tarafında Java Script kullanarak sunucuya http istekleri yapar ve gelen yanıtları işler. Ji SON veya X emel formatında veri alışverişi yapılabilir ve bu verileri dinamik olarak sayfaya ekleyebiliriz. ajax, kullanıcı dostu ve etkili web uygulamaları oluşturmanın temel taşlarından biridir ve modern web geliştirme süreçlerinde yaygın olarak kullanılmaktadır.

GÖZDEN GEÇİRELİM

1. PHP'de XML dosyasından veri çekmek için hangi yöntemler kullanılabilir?
2. AJAX nedir ve nasıl kullanılır?
3. SimpleXML nedir ve ne işe yarar?
4. DOMDocument ile XML ayrıştırma nasıl yapılır?
5. AJAX ile XML verisi nasıl alınır ve işlenir?

Cevaplar:

1. PHP'de XML dosyasından veri çekmek için SimpleXML, DOMDocument, ve XMLReader gibi yerleşik XML ayrıştırıcıları kullanılabilir. Bu sınıflar, XML belgelerini yüklemek, belirli öğeleri seçmek, veri çekmek ve manipüle etmek için kullanılır.
2. AJAX, Asenkron JavaScript ve XML'in kısaltmasıdır ve web uygulamalarında kullanıcı etkileşimini geliştirmek için kullanılan bir tekniktir. AJAX kullanılarak, tarayıcı tarafında JavaScript kullanılarak sunucuya HTTP istekleri yapılır ve gelen yanıtlar işlenir. Bu şekilde, sayfa yenilenmeden sunucu ile veri alışverişi yapılabilir, böylece daha hızlı ve daha akıcı bir kullanıcı deneyimi sağlanır.
3. SimpleXML, PHP'de XML belgelerini işlemek için kullanılan basit ve kullanıcı dostu bir ayrıştırıcıdır. XML belgelerini yüklemek, belirli öğeleri seçmek ve veri çekmek için kullanılır.
4. DOMDocument, PHP'de XML belgelerini yüklemek ve işlemek için kullanılan bir ayrıştırıcıdır. XML belgesini yükledikten sonra, belirli öğeleri seçmek için XPath veya DOM yöntemlerini kullanabilirsiniz. Bu sayede, belgedeki verileri çekebilir ve manipüle edebilirsiniz.
5. AJAX kullanılarak, tarayıcı tarafında JavaScript ile sunucuya HTTP istekleri yapılır ve gelen yanıtlar işlenir. XML verisi almak için genellikle XMLHttpRequest objesi kullanılır. Gelen XML verisi JavaScript'te işlenerek, istenen verilere erişilir ve kullanıcı arayüzünde gösterilir veya manipüle edilir. Bu sayede, sayfa yenilenmeden dinamik içerikler görüntülenebilir.

KAYNAKÇA

Anbiah, R. R. J., Bhattarai, R., & Sedliak, M. (2011). *PHP ajax cookbook*. Packt Pub.

Learn to code. W3Schools Online Web Tutorials. (n.d.). <https://www.w3schools.com/>

Vaswani, V. (2002). *XML and PHP*. Sams Pearson Education distributor.

FAYDALI KAYNAKLAR VE INTERNET KAYNAKLARI

PHP - ajax and mysql. PHP AJAX and MySQL. (n.d.).
https://www.w3schools.com/php/php_ajax_database.asp

PHP - Ajax and php. (n.d.). https://www.w3schools.com/php/php_ajax_php.asp

PHP - Ajax introduction. (n.d.). https://www.w3schools.com/php/php_ajax_intro.asp

PHP example - ajax and XML. PHP AJAX and XML. (n.d.).
https://www.w3schools.com/php/php_ajax_xml.asp

PHP example - ajax live search. PHP AJAX Live Search. (n.d.).
https://www.w3schools.com/php/php_ajax_livesearch.asp

PHP example - ajax poll. PHP AJAX Poll. (n.d.).
https://www.w3schools.com/php/php_ajax_poll.asp

PHP SimpleXML - get node/attribute values. (n.d.).
https://www.w3schools.com/php/php_xml_simplexml_get.asp

PHP simplexml parser. (n.d.). https://www.w3schools.com/php/php_xml_simplexml_read.asp

PHP XML Dom Parser. (n.d.). https://www.w3schools.com/php/php_xml_dom.asp

PHP XML expat parser. (n.d.). https://www.w3schools.com/php/php_xml_parser_expat.asp

PHP XML parsers. (n.d.). https://www.w3schools.com/php/php_xml_parsers.asp