

**T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI**

**AJAX TEKNİĞİ KULLANILARAK ÇOKTAN SEÇMELİ
SINAV SİSTEMİ UYGULAMASI**

YÜKSEK LİSANS TEZİ

Oğuz ATA

İSTANBUL 2008

**T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI**

**AJAX TEKNİĞİ KULLANILARAK ÇOKTAN SEÇMELİ
SINAV SİSTEMİ UYGULAMASI**

YÜKSEK LİSANS TEZİ

Oğuz ATA

**Doç.Dr. Hasan Hüseyin BALIK
(Tez Danışmanı)**

İSTANBUL 2008

TEŞEKKÜR

Bu günlere gelmemi sağlayan aileme, çalışmanın her safhasında gayretini esirgemeyen ve özveri ile yaklaşan danışmanım Doç. Dr. Hasan Hüseyin BALIK ve ailesine teşekkürler.

YEMİN METNİ

Sunduğum yüksek lisans tezini akademik etik ilkelerine bağlı kalarak, hiç kimseden akademik ilkelere aykırı bir yardım almaksızın bizzat kendimin hazırladığına and içerim.

13/11/2008

Oğuz Ata



T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜ
TEZLİ YÜKSEK LİSANS TEZ SINAV TUTANAĞI

13.../11.../2008

Enstitümüz Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Bilim Dalı yüksek lisans öğrencilerinden 060820007 numaralı **Oğuz ATA'a** "*Beykent Üniversitesi Lisansüstü Eğitim - Öğretim Yönetmeliği*"nin ilgili maddesine göre hazırlayarak, Enstitümüze teslim ettiği "**AJAX TEKNİĞİ KULLANILARAK ÇOKTAN SEÇMELİ SINAV SİSTEMİ UYGULAMASI**" tezini, Yönetim Kurulumuzun 13.10.2008 tarih ve 2008/17 sayılı toplantısında seçilen ve Fakülte binasında toplanan jüri üyeleri huzurunda, ilgili yönetmeliğin (c) bendi gereğince aday tarafından savunulmuş ve sonuçta adayın tezi hakkında **oybirliği** ile **Kabul** kararı verilmiştir.

İşbu tutanak, 4 nüsha olarak hazırlanmış ve Enstitü Müdürlüğü'ne sunulmak üzere tarafımızdan düzenlenmiştir.

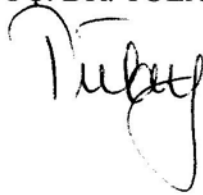
DANIŞMAN
DOÇ. DR. HASAN HÜSEYİN BALIK



ÜYE
Prof. DR. İLHAMİ YAVUZ



ÜYE
YRD. DOÇ. DR. TÜLAY TİNLİ



ÖZET

AJAX TEKNIĐİ KULLANILARAK ÇOKTAN SEÇMELİ SINAV SİSTEMİ UYGULAMASI

Oğuz ATA

Günümüzde bilgisayar teknolojisinin kullanımı yaygınlaştıkça internet hayatımızın vazgeçilmez bir parçası olmuştur. Son yıllarda internet bankacılıktan ticarete, eğitimden eğlenceye kadar birçok alanda yaygın bir kullanım alanı bulmuştur.

Bu tez çalışmasında internetin eğitim alanında kullanımının verimli ve hızlı bir şekilde yapılabilmesi için çözümler önerilmiş ve bu önerileri kullanan çoktan seçmeli sınav sistemi geliştirilmiştir.

Bu projenin bilime iki önemli katkısı bulunmaktadır. Bunlardan birincisi bant genişliğini verimli şekilde kullanan bir yazılımın üretilmesi, diğeri ise verimli bir soru bankası üretme mekanizmasının geliştirilmesidir.

Anahtar Kelimeler: Çoktan seçmeli sınav sistemi, AJAX, DOM, XMLHttpRequest

ABSTRACT
IMPLEMENTATION OF MULTIPLE CHOICE EXAMINATION
SYSTEM BASED ON AJAX TECNIQUE

Oguz ATA

Nowadays, internet becomes one of the unavoidable parts of our lives as a result of wide spread usage of computer technologies. Recently internet is used form banking to trade, from education to entertainment.

In this thesis, ideas for efficient and fast usage of internet on education have been suggested and applied to implementation of multiple choice examination system.

There are two contributions to the science in this thesis. First one is the implementation of new software which uses the bandwidth efficiently. The last one is to introduce new mechanism for creating question bank.

Keywords: Multiple choice examination system, AJAX, DOM, XMLHttpRequest

İçindekiler

1. GİRİŞ.....	1
2. AJAX.....	2
2.1. AJAX Nedir?	2
2.2. Ajax'ın Avantajları	5
2.3. Ajax'ın Dezavantajları	6
3. HTML DOKÜMAN OBJE MODELİ (HTML DOM):	7
3.1. Core DOM	7
3.2. XML DOM	7
3.3. HTML DOM.....	7
3.4. DOM Metot ve Özellikleri.....	8
3.4.1. Düğümler	10
4. XMLHttpRequest	19
4.1. GİRİŞ	19
4.2. METOTLAR.....	19
4.3. ÖZELLİKLER.....	21
5. ÇOKTAN SEÇMELİ SINAV SİSTEMİ.....	22
5.1. Giriş:	22
5.2. Soru Üretim Mekanizması	22
5.3. Uygulamanın Özellikleri:	26
5.4. Dosya Yapısı ve Özellikleri	26
5.5. Uygulamanın Ekran Görüntüleri ve Kullanımı:	32
5.5.1. Sisteme giriş:	32
5.5.2. Ders Ekleme	33
5.5.3. Ders Silme	33

5.5.4.	Modül ekleme/silme	34
5.5.5.	Konu Ekleme silme	35
5.5.6.	Soru işlemleri.....	36
5.5.7.	Sınav hazırlama	37
5.5.8.	Öğrencileri sınavlara atama	38
5.5.9.	Sınav Modülü	38
6.	SONUÇ.....	40

1. GİRİŞ

Dünya üzerindeki tüm kişisel ve kurumsal bilgisayarları birbirine bağlayan bir ağ olarak internetin, yaygın olarak kullanılmaya başlanması toplumu “Endüstri” toplumundan “bilgi” toplumuna dönüştürmüştür. Günümüzde internet bankacılıktan ticarete eğitimden eğlenceye kadar birçok konuda hayatın vazgeçilmez bir parçası olmuştur. İnternetin yaygın kullanımı ile eğitimin mekândan bağımsız olarak uzaktan yapılması yaygınlaşmaktadır. Ayrıca örgün eğitimde laboratuvar için ciddi manada makine ve teçhizat ihtiyacı duyulur iken hatta bazı konularda laboratuvar kurulması imkânsız iken internet ortamında görsel programlama teknikleri kullanılarak eğitimin görselleştirilmesi mümkün hale gelmiştir.

İnternetin bu derece yaygın kullanımı bazı limitleride beraberinde getirmiştir. Bunların en başında bant genişliğindeki limitler gelmektedir. Bilgi toplumunun en önemli anahtar konusu bilginin hızlı bir şekilde yayılması olduğuna göre artan ihtiyaç bant genişliği artırma ve modülasyon teknolojilerindeki teknolojilerin her zaman bir adım önünde olmuştur. Bu nedenle mevcut bant genişliğinin verimli bir şekilde kullanılması çalışmaları güncel akademik ve ticari araştırmalarda önemli bir yer tutmaktadır.

Bilgisayar destekli eğitimin bir parçası olan ölçme ve değerlendirme, başka bir deyişle sınav ve sınavlarla bağlantılı olarak soru bankası oluşturulması ve bunun verimli bir şekilde kullanılması sorunu çözüm bekleyen başka bir konudur. Soru bankası hem güncel olmalı hem de bankadaki soru çeşitliliği bakımından büyük olmalıdır.

Bu tez çalışmasının amacı hem bant genişliğini verimli bir şekilde kullanan, hem de soru bankası oluşturmak için gelişmiş bir soru üretme mekanizmasına sahip bir çoktan seçmeli sınav sistemi geliştirmektir..Bu tez çalışmasında gerçekleştirilen projede internet tarayıcıların asenkron olarak sunucudan istekte bulunabilmesini sağlayan ajax tekniği kullanılacaktır. Bu sayede bant genişliğinin verimli bir şekilde kullanılması sağlanacaktır. Bu tez çalışmasında geliştirilen projede ayrıntıları verilen SÜM(Soru Üretme Mekanizması) ile de güncel ve kapsamlı sınav oluşturulması sağlanacaktır.

Tez altı bölümden oluşmaktadır. İkinci bölüm ajax hakkında okuyuculara genel bilgiler verip araştırmacıların bu çalışmayı kolaylıkla anlayabilmesi için bir temel oluştururken üçüncü bölüm DOM hakkında olacaktır. Dördüncü bölüm ise Ajaxın kullanılabilmesi için gerekli olan XMLHttpRequest objesi için temel bilgileri içermektedir. Beşinci bölüm bütünü ile geliştirilen projeye ayrılmıştır. Tez altıncı bölüm olan sonuç bölümü ile bitirilmiştir.

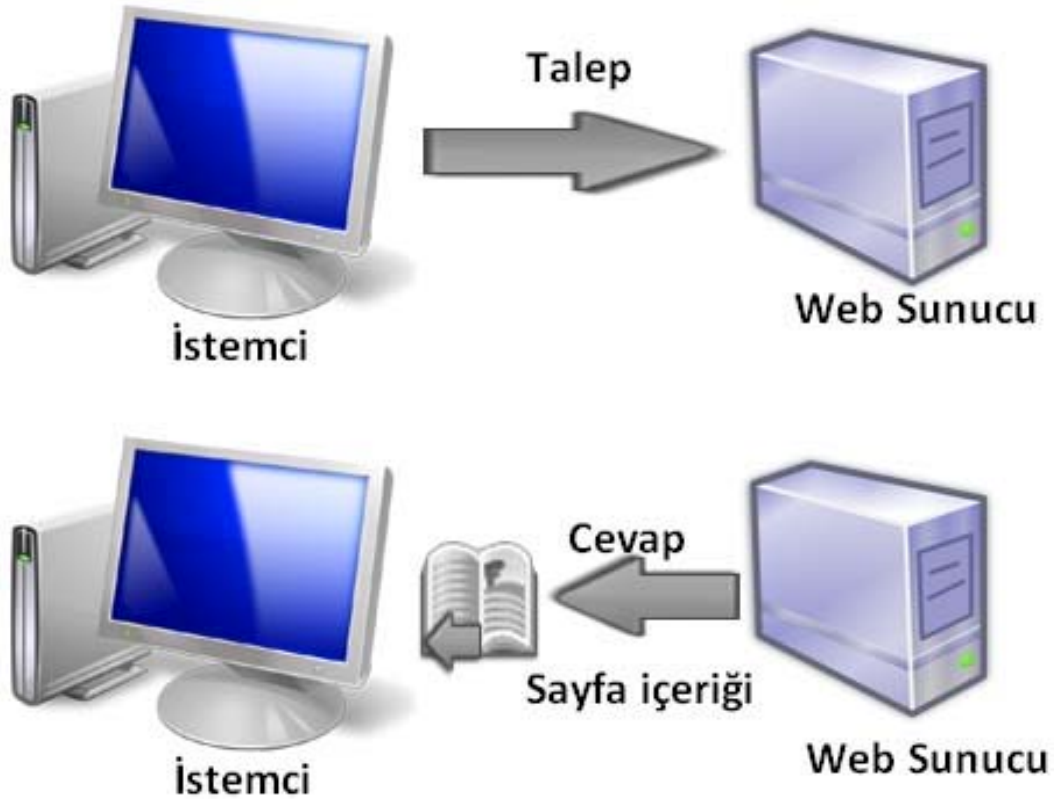
2. AJAX

Bu bölümün amacı bant genişliğini verimli kullanan çoktan seçmeli sınav sistemi geliştirme projesinde kullanılan ajax tekniğinin temel kavramları hakkında bilgi vermektir.

2.1. AJAX Nedir?

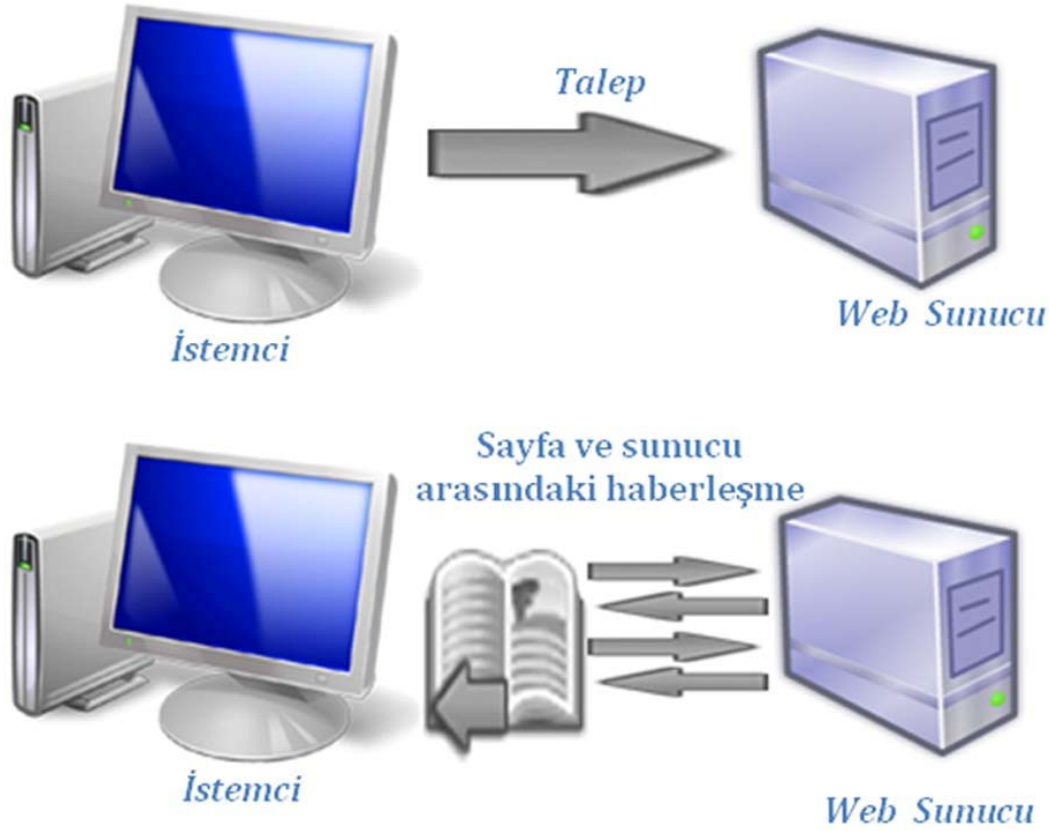
AJAX (Asynchronous Javascript and XML) birileri tarafından geliştirilmiş bir teknoloji olmayıp, daha çok bir yöntemdir ve içerisinde bir çok teknolojiyi barındırır. Bir yöntem olarak ajax, internet tarayıcılarının asenkron olarak sunucudan istekte bulunmasını sağlayarak web uygulamalarının kullanıcı arabirimini masaüstü uygulamalara bir adım daha yaklaştırır.

Geleneksel web uygulamalarında, internet tarayıcıları sunucudan, sayfanın tüm içeriğini getirmesi için istekte bulunur. Kullanıcı gelen sayfa üzerinde bulunan bir bağlantıyı tıklayarak ya da doldurduğu bir formu göndererek yeni bir istekte bulunabilir. Sunucu gelen bu yeni istek için yine kullanıcıya sayfanın tamamını gönderir. Bu durum Şekil 1 de gösterilmiştir[1].



Şekil.1 Geleneksel yapıda web sayfasın isteği[1]

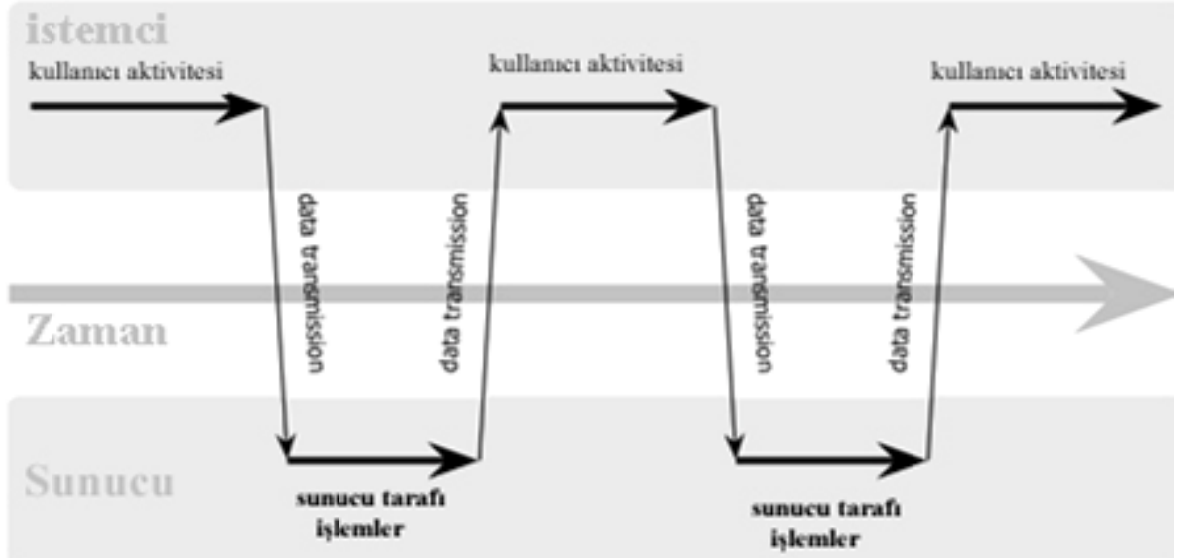
Ajax tekniğinde ise, Şekil 1’de gösterilen durum farklıdır. Kullanıcının tıkladığı bağlantı veya gönderdiği forma karşılık, sayfanın tamamı yerine mevcut sayfa içerisinde değişmesi gereken içeriğin bulunduğu bölüm, gelen cevap ile güncellenir. Bu durum Şekil 2’de gösterilmiştir. Şekil 2’de gösterilen yapıyı gerçekleştirebilme, başka bir deyişle sayfanın ilgili yerlerinin güncellenebilmesini sağlayabilmek için Microsoft’un geliştirdiği, XMLHttpRequest nesnesini kullanan bir arabirime ihtiyaç vardır.[1]



Şekil.2 AJAX Tekniği İle Web Sayfasının İsteği[1]

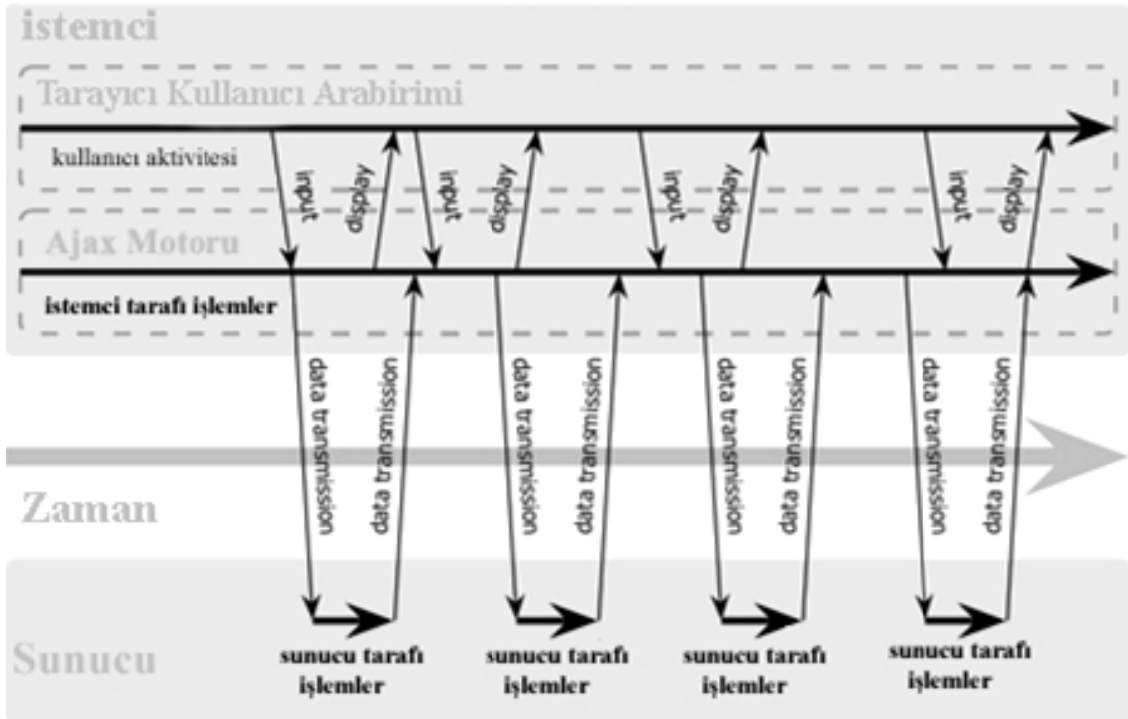
Şekil 3,4’de geleneksel yolun ajax tekniği ile karşılaştırılması zaman ekseninde gösterilmiştir. Şekil 3,4 den açıkça anlaşılacağı gibi geleneksel web modelinde her bir kullanıcı aktivitesinde senkron olarak sunucudan istemciye veri akışı mevcut iken ajax web uygulama modelinde veri akışı asenkronudur. Bu tür yaklaşım ile gereksiz veri trafiğinin önüne geçilmiştir. Yukarıda belirtilen “senkron” ve “asenkron” ifadeleri zaman ekseninde senkronizasyonu belirtmemektedir. Bu çalışmada “senkron” ile istemcide gösterilen sayfanın bütünü ile sunucudan alındığı ve bu sürede tarayıcının başka bir iş yapmadan bu isteğin cevabını beklediği, “asenkron” ile istemcide gösterilen sayfanın, herhangi bir zaman diliminde bir kısmının sunucudan alınması ve tarayıcının bu işlemler sırasında diğer işlemleri yapabilmesine olanak sağlaması kast edilmiştir.

Klasik web uygulama modeli (senkron)



Şekil.3 Geleneksel yol ile haberleşme karşılaştırması

AJAX web uygulama modeli (asenkron)



Şekil.4 Ajax tekniği ile haberleşme karşılaştırması

Şekil 3 ve Şekil 4 de görüldüğü gibi Ajax tekniğinde istekler asenkron olarak ele alınmaktadır.

Kaynaklarda [3,4] AJAX tekniğinin kullandığı teknolojiler, aşağıda sıralanmıştır.

- **HTML/XHTML:** Sayfanın içeriğinin görüntülenmesini sağlar.
- **DOM:** Sayfa içerisindeki kontrollere ulaşabilmesini ve içeriklerinin değiştirilebilmesini sağlar.
- **XMLHttpRequest:** Sunucu ile tarayıcının iletişimi sağlayan nesnedir. Bu nesne sayfanın tümünün istemciye sunucu tarafından gönderilmeden haberleşilebilmesini sağlar.
- **JavaScript:** AJAX Teknolojisini uygulamak için kullanılan script dilidir.
- **XML,JSON:** Sunucu ve istemci arasındaki veri iletişimini sağlayan modalardır.
- **XSLT:** XML tabanlı stil dosyasıdır.
- **CSS:** Sayfadaki öğelere biçim ve stiller uygulamamızı sağlar.

2.2. Ajax'ın Avantajları

- Sunucu yükünü düşürür. Ajaxın temel mantığı, sadece ihtiyaç olunan verilerin sunucudan istenmesi ve sadece ihtiyaç olunan verilerin sunucuya gönderilmesi olduğu için gereksiz istek ve cevapları azaltır.
- Kullanıcı arabirimini daha kullanışlı yapar.
- Bant genişliği maliyetini düşürür. Bazı görevler klasik yolla sunucu tarafında yapılırken Ajax tekniği ile bunun bir kısmı istemci tarafına kaymıştır. Bu sayede sunucu ihtiyaçları ve bant genişliği ihtiyaçları düştüğünden aynı bant genişliği ile daha fazla fonksiyonel verinin iletimi sağlanmış olur.
- Bu teknik ile geniş bir kullanım alanına sahip teknolojilerin kullanılması ve bunların standartlar çerçevesinde yapılması sayesinde, kullanımı için herhangi bir özel programa ya da plug-in e ihtiyaç duymamaktadır.[5]
- Arabirimleri uygulama kısmından ayırır. Bilgi ve sunum katmanı böylece birbirinden ayrılır.[6]

2.3. Ajax'ın Dezavantajları

- Ajax ile proje geliřtirmek geliřtirme süresini ve geliřtirme maliyetini arttırmaktadır. Ajax ile geliřtirme, genellikle klasik web uygulamalarını geliřtirmekten daha karmařık ve zordur. Çünkü birçok teknoloji ve özel yapıları bir arada kullanmak gereklidir.[7]
- Dinamik olarak oluřturulan sayfa tarayıcının “geçmiř” motoruna kaydedilmez. Dolayısı ile tarayıcıda “geri” tuřu ile ancak Ajax ile oluřturulmuř sayfadan önceki içerięe ulařılabilir. Yani “geri” tuřu ajax kullanılan sayfada düzgün bir biçimde kullanılamaz.[7]
- Dinamik sayfa güncellemesi, sayfalara “yer imi(bookmark)” koymayı güçleřtirmiřtir. Bu problem sayfanın durumunu URL bölüm iřaretleyicisi ile iřaretlenerek kısmen çözülmüřtür.[7]
- İnternet arama motorlarının, sayfaları indekslemek için kullandıkları robotlar, sayfalardaki javascript gibi kodları çalıştıramazlar. Bu sebeple javascript ile çağırđığımız dinamik içerikler, arama motorları tarafından algılanamazlar.[8]
- Javascript çalıştırma özellięi kapatılmıř tarayıcılar yada mobil telefonlar, PDA'lar ve ekran okuyucu gibi javascript'i çok kısıtlı kullanabilen ilkel tarayıcıya sahip cihazlar ajax ile yapılan uygulamaları çalıştıramaz.[9]
- Ajax için isteklerin aynı bölgede olma zorunluluęu bulunmaktadır. [10]

3. HTML DOKÜMAN OBJE MODELİ (HTML DOM):

Ajax tekniğinde dinamik içerik güncelleme ve kullanıcı etkileşimi için DOM(Document Object Model) kullanıldığından DOM ile alakalı bir bölümün hazırlanma ihtiyacı, bu tez çalışmasında geliştirilen çoktan seçmeli sınav sisteminin anlaşılabilmesi için oluşmuştur. DOM HTML dokümanlarına erişilebilmesi ve bunlar üzerinde değişiklik yapabilmesi için geliştirilen W3(Word Wide Web Konsorsiyumu) standartlarına uygun bir standartlar topluluğudur. Bir başka deyişle, DOM, arayüzün bir parçası olarak, programların ve scriptherin dinamik olarak doküman üzerindeki stil, yapı ve içeriklerine erişilebilmesini ve bunların güncellenebilmesini sağlar. Böylelikle DOM standardına göre geliştirilmiş doküman akıllı bir belge haline dönüşür ve ilgili doküman hem daha fazla işlem yapabilir hemde işlem sonuçlarını sunulmuş olan sayfa içerisinde gösterebilir[11].

DOM temel olarak üç seviyeye ayrılabilir. Bunlar:

- Core DOM : Yapısal dokümanlar için standart model
- XML DOM : XML dokümanları için standart model
- HTML DOM : HTML dokümanları için standart model

Şeklinde. Aşağıda her bir seviye ayrı ayrı açıklanmıştır.

3.1. Core DOM

Core DOM, DOM için ana fonksiyonları barındırır ve diğer seviyelere barındırdığı bu fonksiyonlara servis eder. Başka bir deyişle DOM' un temel seviyesidir.

3.2. XML DOM

XML DOM, XML dokümanların obje ve özelliklerini tanımlar ve metotların (arabirimlerin) dokümana erişimini sağlar.

3.3. HTML DOM

HTML DOM, HTML için standart obje modeli olup, standart bir programlama dili olarak kullanılabilir. Bu dil platformdan bağımsızdır ve W3 standartlarına uygundur. HTML DOM' un kullanımına örnek HTML dokümanı Şekil 5.a da verilmiştir

HTML DOM kullanımına örnek olarak:

HTML Dokümanı

```
<div id="ornek">  
</div>
```

(a)

```
var element_eris= document.layers["ornek");//Netscape için  
var element_eris= document.all["ornek");//Internet Explorer için
```

(b)

```
İf (document.all)  
{  
//internet explorer için kodlar  
}  
else if(document.layers)  
{  
//netscape için kodlar  
}
```

Şekil.5 (c) HTML DOM örnek dokümanı[1]

(a) HTML örnek kodu, (b) Javascript kullanarak bu koddaki “div” elementine erişebilmek için kullanılan yapı, (c) İki tarayıcıda da çalışabilmesi için şu şekilde kullanılan yapı.

3.4. DOM Metot ve Özellikleri

DOM metot ve özellikleri iki ana guruba ayrılabilir. Bunlar sırası ile

- ✓ DOM düğümlerinde gezinebilmeyi ve düğüm değer ve özelliklerini çağırabilmeyi sağlayan metot ve özellikler,
- ✓ DOM düğümleri ekleyebilmeyi, silebilmeyi, üzerlerinde işlem yapabilmeyi sağlayan metot ve özellikler

‘dir . Yukarıda belirtilen her iki gurup için belirtilen metot ve özelliklerin DOM destekli herhangi bir dil içerisinde kullanılabilmesi için DOM ağacı için yapacağı işlemlerde aynı

metot ve özellikler kullanılmalıdır. Aşağıda bazı DOM metod ve özellikleri, bu tez çalışmasında yapılan projenin anlaşılabilmesi için sıralanmıştır.

getElementById

“ornek” id sine sahip “div” elementine DOM ile erişmek için “document” objesinin metodlarından biri olan getElementById metodunu kullanılabilir. Şekil 6’da örnek kullanım gösterilmiştir[1].

```
var element_eris = document.getElementById(“ornek”);
```

Şekil.6 *getElementById* kullanımına örnek

getElementsByTagName

Bu komut getElementById’ye benzer, aralarındaki fark, elementin id si yerine “tag”ının ismi kullanılarak erişim sağlanmasıdır. Bu metodun kullanımı için Şekil 7’de örnek bir html kodu verilmiştir[1].

```
<p>merhaba  
</p>  
<p>dunya  
</p>
```

Şekil.7 *Örnek HTML kodu*

Şekil 8’de verilen koddaki “p” elementine javascript kullanılarak ulaşmak için;

```
var paragraf=document.getElementsByTagName(“p”);
```

Şekil.8 *Elemente Erişmek için örnek kod*

Şekil 8’ de verilen ifade kullanılabilir. Paragraf değişkeni, doküman içerisindeki bütün “p” elementlerini içerisinde barındıran bir element dizisidir. Örneğin bu dokümanın içerisinde kaç tane paragraf olduğunu öğrenilmek istenirse[1];

```
alert(paragraf.length);
```

Şekil.9 *Paragraf sayısının ekrana yazdırılması örneği*

Şekil 9 ifadesinin kullanılması yeterlidir. Eğer dokümanın içerisindeki her bir paragrafa tek tek erişmek istenirse Şekil 10’ daki kod kullanılabilir.

```
for(i=0;i<paragraf.length;i++){  
    paragraf[i] .....  
}
```

Şekil.10 Parağraflara erişim yöntemi

getAttribute

Elemente ait öznitelikleri getirir. Örnek kullanım Şekil 11’de verilmiştir

```
<p id="giris" title="giris mesajı">  
</p>
```

Şekil.11 *getAttribute* ile erişilecek örnek html kodu

getAttribute ile “p” elementinin özniteliklerine erişmek için javascript tarafında Şekil 12’deki kodların yazılması gerekir.

```
var elementimiz= document.getElementById("giris");  
var oznitelik=elementimiz.getAttribute("title");
```

Şekil.12 “title” özelliğine erişen örnek kod

Bu şekilde “p” elementinin title özniteliği alınmış oluyor.

3.4.1. Dğümler

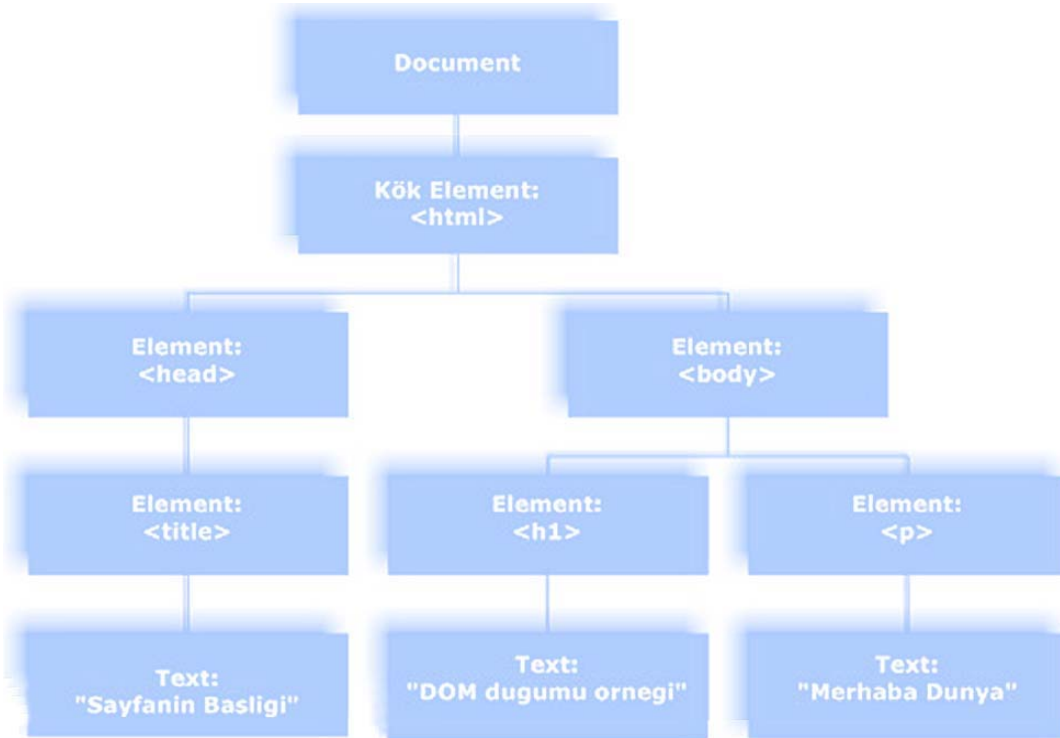
DOM’a göre bir HTML dokümanındaki her şey aslında bir düğüme bağlıdır. Buna Göre

- ✓ Bütün doküman,”document” düğümüdür.
- ✓ Her HTML tagı bir “element” düğümüdür.
- ✓ HTML elementlerinin her bir metni, “text” düğümüdür.
- ✓ Her HTML özniteliği bir “attribute” düğümüdür[12].

getElementById, getElementsByTagName, getElementsByAttribute metotları ile aslında dokümanın düğümlerine erişilmiş olur. Şekil 13’de örnek html dokümanı ve Şekil 14’de de aynı dokümanın düğüm yapısı verilmiştir.

```
<html>
<head>
<title>Sayfanin Basligi</title>
</head>
<body>
<h1>DOM dugumu ornegi</h1>
<p>Merhaba Dunya</p>
</body>
</html>
```

Şekil.13 Örnek HTML kodu



Şekil.14 Örnek HTML kodunun düğüm yapısı

Şekil 14’de verilen düğümleri açıklarsak;

- ✓ <html> düğümünün bir ebeveyn düğümü yoktur, yani bir kök düğümdür.
- ✓ <head> ve <body> düğümlerinin ebeveyn düğümü <html> dir.
- ✓ “Merhaba Dünya” “text” düğümünün ebeveyn düğümü <p> düğümüdür.

Birçok element çocuk (child) düğümüne sahiptir. Şekil 14 bu bağlamda tekrar incelenirse;

- ✓ <html> iki tane çocuk düğüme sahiptir. Bunlar <head> ve <body> düğümleridir.
- ✓ <head> düğümü bir çocuğa sahiptir. Bu düğüm <title> düğümüdür.
- ✓ <title> düğümü bir çocuğa sahiptir. Bu düğüm bir “text” düğümü olan “Sayfanın Başlığı” dır.
- ✓ <h1> ve <p> düğümleri kardeş (siblings) düğümlerdir. Her ikisinde <body> düğümünün çocuklarıdır[1].

Aşağıda DOM’ un düğümler üzerinde gezmek için kullandığı komutlar verilmiştir.

parentNode

parentNode özelliği referans olarak ebeveyn elementini döndürür. Sadece element düğümler bu özelliğe sahiptir. Aşağıda bu komuta örnek verilmiştir.

```
var elm = document.getElementById("elm");  
var par=elm.parentNode;
```

Şekil.15 parentNode kullanım örneği

childNodes

childNodes özelliği çocuk elementlerin koleksiyonunu (dizi olarak) geriye döndürür. childNodes özelliği “length” özelliği ile sahip olunan çocuk sayısını verir. Eğer bağlı herhangi bir çocuk element yok ise sıfır döndürür. Bu özellik sadece element düğümleri için geçerlidir.

“length” özelliği bazı tarayıcılarda farklı şekilde hesaplanabilen bir özelliktir. Bazı tarayıcılar elementler arasındaki boşlukları ve line break leri sayıya dahil ederken bazıları bunları dahil etmezler. Aşağıda childNodes örnek kullanımı verilmiştir[15].

```
var elm = document.getElementById("elm");  
var par = elm.parentNode;  
var cocuklar= par.childNodes;  
alert(cocuklar.length);
```

Şekil.16 childNodes kullanım örneği

firstChild

firstChild özelliği elementin ilk çocuğunu referans olarak geri döndürür. childNodes[0] ile aynı işi yapar[1].

lastChild

Elementin son çocuğunu referans olarak geriye döndürür[1].

nextSibling

Eğer iki düğüm aynı ebeveyne sahipse bunlar kardeş düğümlerdir. Bu ikisinin de aslında ebeveyn düğümün childNodes dizisinin elemanları olduğu anlamına gelir. nextSibling özelliği kendinden bir sonraki kardeşi referans olarak geriye döndürür[1].

previousSibling

previousSibling kendisinden bir önceki kardeş düğümü referans olarak geriye döndürür.[1]

nodeValue

Eğer bir düğümün içeriği getirilmek istenirse bunun için nodeValue özelliği kullanılır.

nodeValue: “Element” düğümleri için “undefined”, “Text” düğümleri için bu düğümün metninin kendisini ve “Attribute” düğümleri için ise bu özniteliğin değerini döndürür. Aşağıda örnek kullanım verilmiştir [12].

```
<html>
<body>
<p id="intro">W3Schools example</p>
<div id="main">
<p id="main1">The DOM is very useful</p>
<p id="main2">This example demonstrates how to use the
<b>nodeValue</b> property</p>
</div><script type="text/javascript">
x=document.getElementById("main2").firstChild;
document.write("First paragraph text: " + x.nodeValue);
</script>
</body>
</html>
```

Şekil.17 nodeValue kullanımı için örnek kod [12]

createElement

createElement ile “frame” ve “iframe” elementleri hariç bütün elementleri oluşturabilmek ve programatik olarak bu elementlere erişip üzerlerinde okuma/yazma işlemlerini yapabilmek mümkündür. IE4 ve öncesinde ise createElement ile sadece “img”, “area” ve “option” elementlerinin oluşturulmasına imkân verir. [13]

Mevcut dokümanda oluşturulan bu elementi eklemek için kullanabilecek iki metot vardır. Bunlar insertBefore ve appendChild metotlarıdır. Şekil 18’de metotlara örnek kullanım verilmiştir

```

<SCRIPT>
function fnCreate(){
    oData.innerHTML="";
    var oOption=oSel.options[oSel.selectedIndex];
    if(oOption.text.length>0){
        var aElement=document.createElement(oOption.text);
        eval("aElement." + oOption.value + "=" + oText.value + "");
        if(oOption.text=="A"){
            aElement.href="javascript:alert('A link.');"
        }
        oData.appendChild(aElement);}
</SCRIPT>
<SELECT ID="oSel" onchange="fnCreate()">
<OPTION VALUE="innerText">A
<OPTION VALUE="value">&lt;INPUT TYPE="button"&gt;
</SELECT>
<SELECT ID=oText onchange="fnCreate()">
<OPTION>
<OPTION VALUE="Text">Text
<OPTION VALUE="More and More Text">More and More Text
</SELECT>
<SPAN ID="oData" >
</SPAN>

```

Şekil.18 createElement için örnek kod [13]

CreateTextNode

Dokümanda “Text” düğümü oluşturmak için kullanılan metottur. Şekil 19’da örnek kullanım verilmiştir.

```
<SCRIPT>

function fnChangeNode(){

    var oTextNode = document.createTextNode("Yeni Text");

    var oReplaceNode = oSpan.childNodes[0];

    oSpan.replaceChild(oTextNode,oReplaceNode);

    alert(oReplaceNode.nodeValue);}

</SCRIPT>

<SPAN ID="oSpan" onclick="fnChangeNode()">

    Orijinal Text

</SPAN>
```

Şekil.19 CreateTextNode için örnek Kod[13]

setAttribute

setAttribute metodu dokümandaki elementlere özellik eklenebilmesini sağlar. Şekil 20’de örnek kullanım verilmiştir.

```
<input type="button" value="style Ekle" onClick="styleEkle()"/>

<input type="button" value="style Sil" onClick="styleSil()"/>

<div id="ornek" >Özellikliğini değiştirmek istediğimiz bölüm

</div>

<script type="text/javascript">

function styleEkle() {

    var styleText= 'background-color: #f1c2b3;';

    var divimiz = document.getElementById('ornek');

    divimiz.setAttribute('style',styleText);

}

function styleSil()

{
```

```
var divimiz = document.getElementById('ornek');  
divimiz.removeAttribute('style');  
}  
</script>
```

Şekil.20 *setAttribute için örnek kod*

appendChild

DOM ağacına oluşturulan bir çocuk elementin eklenebilmesini sağlayan metottur. Şekil 21’de örnek kullanım verilmiştir.

```
var newP = document.createElement("p");  
var txt = 'ogo was here.';  
var newT = document.createTextNode(txt);  
newP.appendChild(newT);
```

Şekil.21 *AppendChild için örnek kod [14]*

Örnekte görüldüğü gibi öncelikle bir “p” elementi oluşturulur, daha sonra bir “text” düğümü oluşturulur ve son olarak da “p” elementine oluşturulan “text” düğümü eklenmiş olur.

removeChild

removeChild metodu DOM ağacından bir çocuk düğümü silmek için kullanılan bir metodudur. Şekil 22’de örnek kullanım verilmiştir.

```
<input type="button" value="İlk Hücreyi Sil"
onClick="birinciSil()"/>
<table id="ornekTablo">
  <tr id="ornekSatir">
    <td>Hücre 1</td>
    <td>Hücre 2</td>
    <td>Hücre 3</td>
    <td>Hücre 4</td>
  </tr>
</table>
<script type="text/javascript">
function birinciSil()
{ var Satir = document.getElementById('ornekSatir');
  Satir.removeChild(Satir.firstChild);
}
</script>
```

Şekil.22 removeChild için örnek kod[15]

4. XMLHttpRequest

4.1. GİRİŞ

XMLHttpRequest (XHR) web server ve tarayıcı arasında, metin ve XML bilgisi transfer etmek için kullanılan web tarayıcı script dillerinin (mesela javascript)kullandığı bir uygulama arabirimidir (API). XMLHttpRequest kullanılarak verilere senkron veya asenkron olarak ulaşmak mümkündür.[16].

XMLHttpRequest arka planda veritabanlarından bilgi getirmek ya da bilgileri değiştirmek için kullanılır. XML ve XMLHttpRequest bilgi getirmek için HTML, JSON veya sadece metin formatlarını kullanabilir.

XMLHttpRequest ajax web geliştirme tekniğinin en önemli parçasıdır ve birçok web sitesi esnek ve dinamik web uygulamaları için XMLHttpRequest nesnesini kullanmaktadır. Örnek web siteleri arasında Google Maps, Windows Live's Virtual Earth, MapQuest ve Facebook sayılabilir.

4.2. METOTLAR

Bu bölümde XMLHttpRequest nesnesinin tez çalışmasında kullanılan bazı metotları verilmiştir.

- ✓ **abort()** : Mevcut isteği iptal eder.
- ✓ **getAllResponseHeaders()**: HTTP başlık bilgisinin tamamını geriye döndürür.
- ✓ **getResponseHeader(başlıkAdı)**: İstenilen başlık bilgisini geriye döndürür.
- ✓ **open(method, URL, async, userName, password)**: Bu metot için parametreler ayrı ayrı incelenirse;
- **Method**: open metotunun ilk argümanı olan “metot”, “GET” veya “POST” gibi metin değerler alır. Eğer sunucudan bir dosya yada bilgi alınacak veya herhangi bir bilgi gönderilmeyecek ise bunun için “GET” kullanılır. Eğer sunucuya bilgi gönderilecek ise bunun için ise “POST” kullanılır. “GET” ve “POST” örnekleri istekte bulunma metodunu belirtir, bunların obje metotları ile karıştırılmaması gerekir. Bu bağlamda basit olarak istek şeklinin belirtildiği argümandır denilebilir. Bunların dışında “PUT”, ”DELETE”, ”HEAD” gibi metotlarda vardır.

- **URL:** URL sunucu üzerinde istekte bulunulacak dosyanın adresini içerir. Bu adres “../ogo.html” gibi bağıl olabileceği gibi “*http://www.beykent.edu.tr/dosya.html*” gibi tam adres şeklinde de olabilir. URL argümanı için önemli özellik, istekte bulunan javascript dosyası ile istekte bulunulan dosyanın aynı etki alanında olması gerektiğidir. Eğer javascript foo.com adresinden çalıştırılıyorsa, bar.com’dan herhangi bir istekte bulunamaz.
- **async:** “true” ya da “false” değerlerini alabilir. Bu özelliği “true” yapılması isteğin asenkron olarak çalışmasını sağlayacaktır. Bu da scriptin geri kalanının bu isteğe gelecek cevabı beklemeden devam etmesini sağlar. Eğer bu özellik “false” yapılırsa script bu noktada cevap gelene kadar bekleyecektir. Async argümanı genellikle “false” değeri ile kullanılmaz. Eğer sunucunun cevap verme süresi uzun ise bu değerin “false” olması durumunda tarayıcı bu sürede duracaktır. Bu çok istenilen bir durum değildir. Ajaxın en büyük avantajlarından birisi olan, kullanıcının sayfa üzerinde etkileşimli olarak gezinirken, isteklerin arka planda çalışması, işlemini yapabilmek için bu özellik “true” olarak “*Request.open(‘GET’, ‘file.html’, true)*” şeklinde kullanılmalıdır.
- **userName ve password:** Bu iki isteğe bağlı argüman en az kullanılan argümanlardır. Javascript, CSS gibi istemci tarafında çalışan yapıların üzerinde kullanıcı adı ve şifre kullanılması sistem güvenliği açısından sakıncalı bir yol olduğu için günümüzde pek tercih edilmemektedir.
- ✓ **Send(content):** Open metodu bir ajax isteği için özellikleri ve detayları belirlemesine rağmen istek için bir hazırlık yapmaz. İsteği gerçekleştirmek için send metodu kullanılır. Eğer istek metodu olarak “GET” kullanılır ise sunucuya veri gönderilmez ve bu durumda *request.send(null)* şeklinde kullanılabilir.

4.3. ÖZELLİKLER

onreadystatechange: Her olay değişiminde yapılmak istenilen işler için kullanılır.

readyState: Aşağıda belirtilen şekilde objenin durumunu geri döndürür.

- ✓ 0=uninitialized-open() henüz çağırılmamış
- ✓ 1=open-send() henüz çağırılmamış
- ✓ 2=send-send() çağırılmış, header ve status uygun
- ✓ 3=receiving-downloading responseText içerisinde bilgi tutuyor
- ✓ 4=load-finished

responseText: Cevabı metin olarak geri döndürür.

responseXML: W3C DOM düğüm-ağaç metotları ve özellikleri kullanılarak ayrıştırılabilir halde XML formatında cevabı geri döndürür.

status : Http durum kodunu numara olarak geriye döndürür.(“not found” için 400, “ok” için 200 vs.)

statusText: Http durum kodunu metin olarak geriye döndürür. (“not found”, ”ok” vs.)

5. ÇOKTAN SEÇMELİ SINAV SİSTEMİ

5.1. Giriş:

Bu projenin amacı güncel web tabanlı programlama teknikleri kullanılarak internet bant genişliğini verimli bir şekilde kullanan ve dinamik soru üretim mekanizmasına sahip bir çoktan seçmeli sınav sistemi oluşturmaktır. Projenin amacından da anlaşılacağı gibi tez konusu iki önemli ana hedefi bulunmaktadır. Bunlar:

- ✓ Bant genişliğinin verimli bir şekilde kullanılması
- ✓ Soru bankasının geliştirilmesi ve yeni sorular üretilmesi kolay bir soru üretim mekanizmasının geliştirilmesi.

Yukarıda belirtilen birinci hedef için ajax tekniği kullanılmıştır. Bu tez çalışmasında kullanılan ajax tekniğinin ayrıntıları bölüm 2,3 ve 4 de verilmiştir. Bu bölümde öncelikle geliştirilen soru üretim mekanizmasından bahsedilecek, daha sonra dosya ve ilişkileri ile devam edilecektir. Bölümün devamında projede kullanılan veritabanı yapısı verilip bölümün sonunda geliştirilen projenin kullanımı ayrıntıları ile verilecektir.

5.2. Soru Üretim Mekanizması

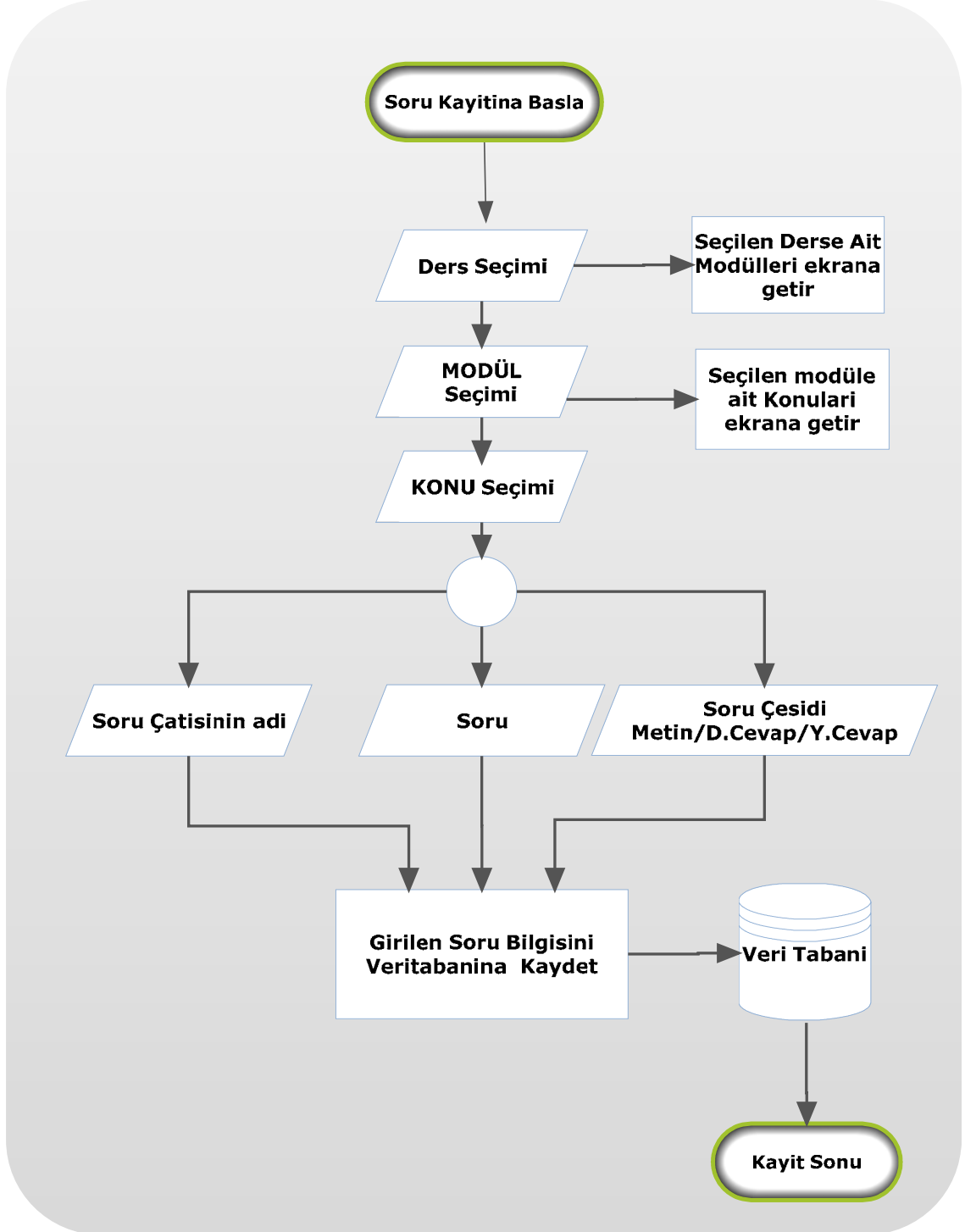
Sınav yapabilmek için bir soru bankasına ihtiyaç vardır. Tasarlanılan sistem bir soru bankası oluşturabilmesine olanak sağlamaktadır. Bu projede gerçekleştirilen soru üretim mekanizması blok diagramı Şekil 23 de gösterilmiştir. Şekil 23 de açıkça görüldüğü gibi soru bankasındaki her bir sorunun istenilen kadar soru metni, doğru cevap ve yanlış cevap seçeneği olabilmektedir. Sistem tarafından sorular oluşturulurken önce muhtemel soru metinlerinden bir tanesi, daha sonra muhtemel doğru cevaplardan bir tanesi ve son olarak yanlış cevaplardan dört tanesi rastgele seçilir ve harmanlanır. Bu sayede aynı soruyu birçok farklı biçimde oluşturmak mümkün olmakta ve bu da soru bankasının yüksek verimle kullanılmasını sağlar. Ayrıca tasarlanılan sistem soru bankasını resim ya da belge olarak değil metin modunda web tabanlı saklamaktadır.

Soru üretme mekanizmasının bir sonraki adımı ise Şekil 23 deki gibi hazırlanıp veritabanına kaydedilen soru bulutunun seçilen sınava göre soru bankası haline dönüştürülebilmesidir. Bu yapının blok diagramı Şekil 24 de verilmiştir. Şekil 24 den anlaşılacağı gibi “Sınava Başla” butonuna basıldıktan sonra sınav ayarları veritabanından

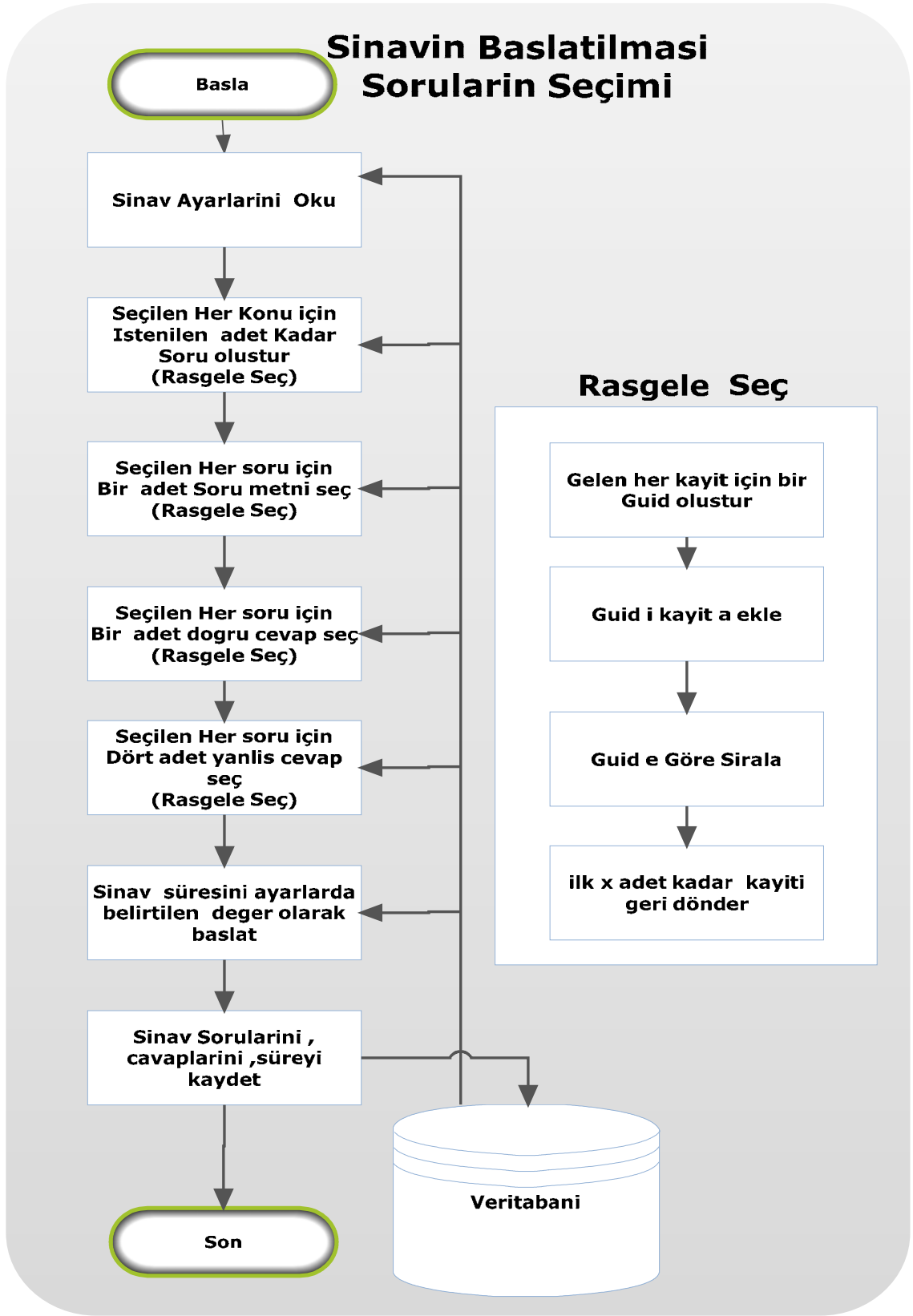
okunur. Sınav ayarlarında belirtilen konu başlıkları için istenilen soru adedi kadar Soru çatısı Şekil 24 de verilen “Rasgele Seç” algoritmasına göre seçilir. Rasgele seçilen soru çatılarına göre muhtemel soru metinlerinden bir soru metni, muhtemel doğru cevaplardan bir doğru cevap, muhtemel yanlış cevaplardan belirtilen adet kadar rasgele seç algoritmasına göre ayrı ayrı seçilir ve bu seçim veri tabanına kaydedilir.

Öğrenci sınav için sisteme girdiğinde daha önce rasgele algoritmasına göre seçilip veritabanına kaydedilen sorular yeniden rasgele sıralanıp öğrencinin karşısına getirilir. Bu işlem öğrencinin sınava her girişinde tekrarlanır. Dolayısı ile öğrenci sayısı kadar grup oluşturularak muhtemel kopya çekmenin önüne geçilmiş olur.

Soru Girisinin Yapilmasi



Şekil.23 Soru Bankasına Soru Giriş



Şekil.24 Sinavin başlatilmasi ve sorularin harmanlanmasi

5.3. Uygulamanın Özellikleri:

Sınav için en önemli bilgi sınava giren öğrencinin verdiği cevaplardır. Bu cevapların, oluşabilecek bağlantı problemlerinden, en az şekilde etkilenecek bir sistem oluşturulması gerekir. Bunun için cevapların kayıt işlemi üç şekilde olmaktadır. Birincisi cevap işaretlendiğinde sadece o soruya verilen cevap ajax tekniği kullanılarak veritabanına kayıt işlemi ya da değiştirme için sunucuya gönderilir. İkincisi bu işlem sırasında ağ bağlantısının kesik olabileceği düşünülerek belirli periyotlar ile o ana kadar verilen cevapların sunucuya gönderilmesi işlemidir. Böylece cevaplar en güncel hali ile ve bağlantı kesikliklerinden en az etkilenecek şekilde veritabanımıza kayıt edilmiş olur. Üçüncü kayıt aşaması ise sınavın bitirilmesi için “Sınavı Bitir” tuşuna basılması ile verilen cevaplar, son olarak veritabanımıza kayıt veya değiştirme işlemi için gönderilir.

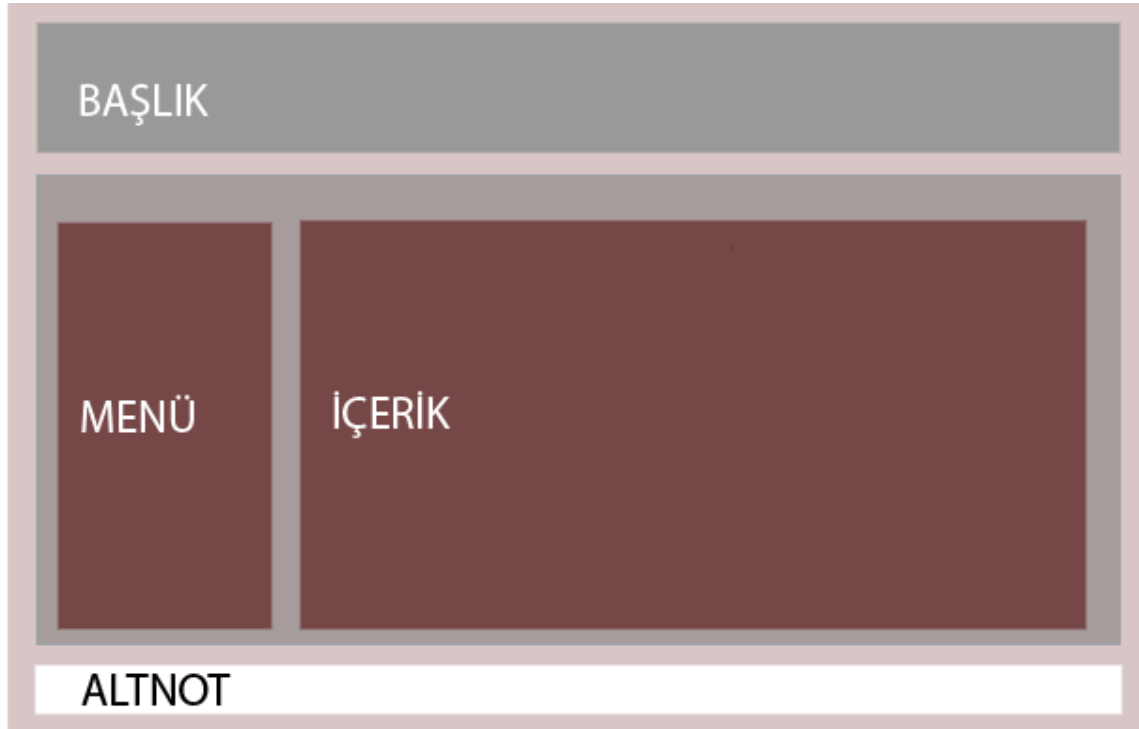
Sınav için diğer önemli nokta ise sınava giren kişilerin birbirlerinden kopye çekmesidir. Bunun en aza indirilmesi için sınava her girişde sınav soruları ve cevapları yer değiştirerek her öğrencinin karşısına farklı dizilişde bir sınav gelmesi sağlanmıştır. Öğrenci için ayrıca cevapladığı soruları başkasının görmesini engellemek için soruları gizleyebilme seçeneği sağlanarak daha güvenli bir sınav ortamı sağlanmıştır. Uygulamamızın bahsettiğimiz özellikleri sağlayabilmesi için bazı temel işlemlere ihtiyacı vardır. Bunlar:

- ✓ Ders İşlemleri
- ✓ Modül İşlemleri
- ✓ Konu İşlemleri
- ✓ Soru İşlemleri
- ✓ Sınav İşlemleri
- ✓ Öğrenci İşlemleri
- ✓ Sınav Değerlendirme İşlemleri
- ✓ Sınav işlemleri

Olarak özetlenebilir.

5.4. Dosya Yapısı ve Özellikleri

Index.aspx dosyamız şekil olarak dört kısımdan oluşmaktadır (Şekil 25). Bunlar başlık, menü, içerik ve alt not kısımlarıdır. Bu bölümlerin içerikleri ajax ile sunucu tarafından gönderilmektedir.



Şekil.25 Sitemin web şablonu

Bu kısımlar normal HTML dokümanı formatında hazırlanmış ve içerikleri için ise sunucu ile ajax tekniğini kullanarak haberleşen javascript fonksiyonu kullanılmıştır. Bu fonksiyon(“*agetir*(‘x’, ‘y’))” iki tane parametre almaktadır. Bunlardan birincisi getirilecek içeriğin hangi DOM elementi içerisine yerleştirileceği, ikincisi ise sunucudan istenilen içeriktir.

```
<body >
  <form id="form1" runat="server">
    <div>
      <div id="header" style="height: 150px ;background-
color:Blue;" >
        <script type="text/javascript">
          agetir('header','header');
        </script>
      </div>
      <div id="anadiv" >
        <table border="0" >
          <tr>
```

```

        <td style="width: 100px" valign="top">
            <div id="menu" >
                <script type="text/javascript">
agetir('menu','menu');
                </script>
            </div>
        </td>

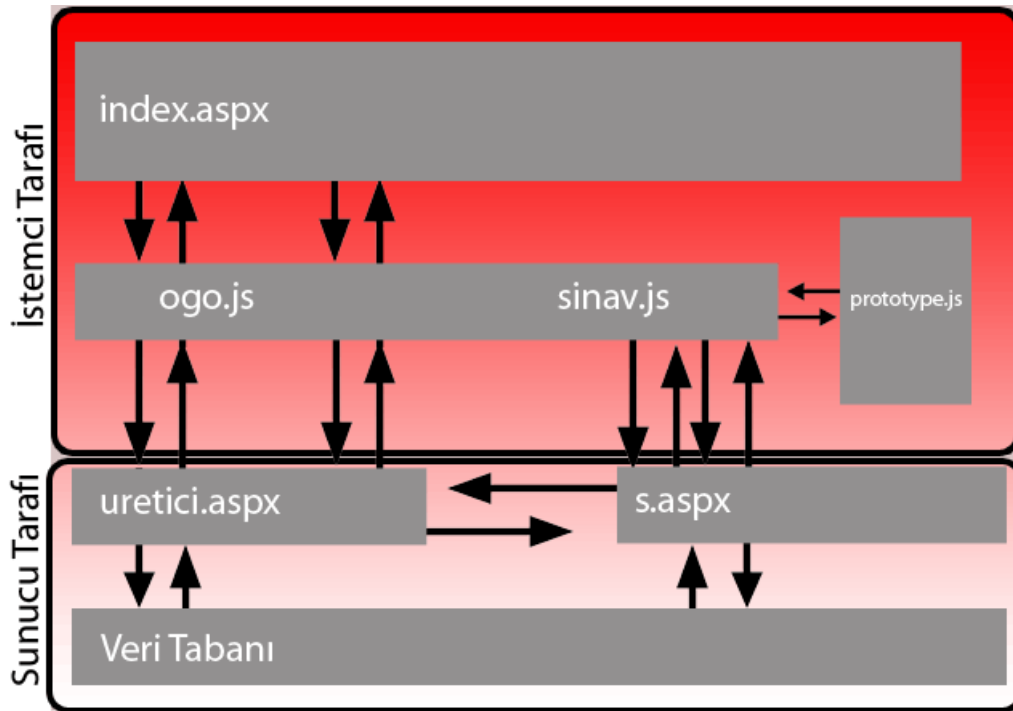
        <td align="center" >
            <div id="icerik">
                <form id="icerikform" action="uretici.aspx" >
                    <script type="text/javascript">
agetir('icerik','ana');
                    </script>
                </form>
            </div>
        </td>
    </tr>
</table>
</div>
</div>
<div id="footer" style="height: 50px;" >
    <script type="text/javascript">
        agetir('footer','footer');
    </script>
</div>
</form>
</body>

```

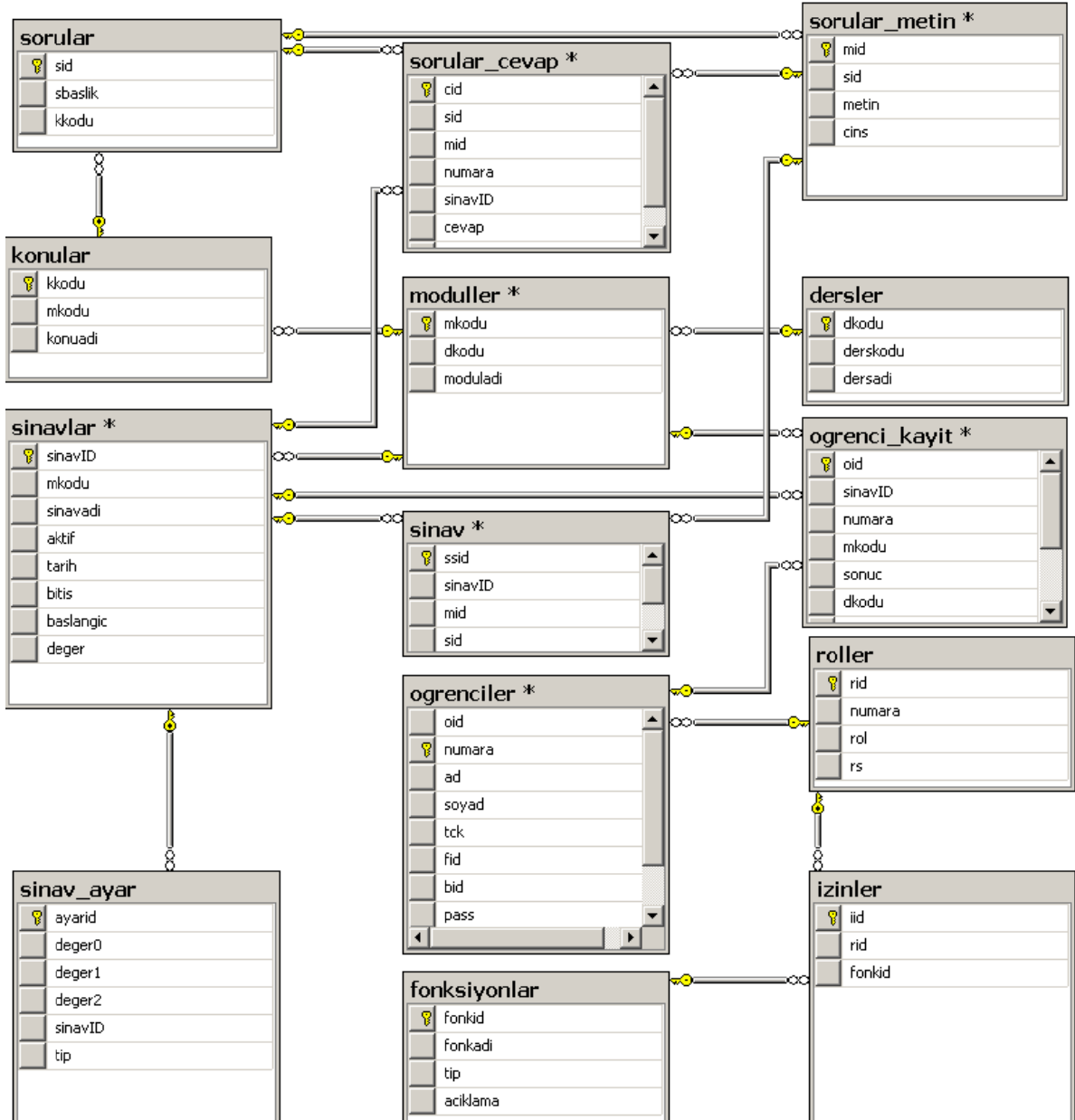
Şekil.26 Index.aspx dosyasının bir kısmı

Tasarladığımız uygulamada ajax tekniğini kullandığımız için sayfamızın dinamik içeriğini yönetebilmek, kullanıcının gördüğü sayfa ile veritabanı ve sunucular ile haberleşmeyi sağlayabilmek gibi görevleri yerine getirebilecek bir javascript arabirimi hazırladık. Bu arabirimler uygulamamızdaki dosyaların birbiri ile haberleşmesini gösteren Şekil 27’de de görülebileceği gibi, ogo.js ve sinav.js dosyalarıdır. Bu dosyalar gerektiğinde bir olaya bağlı çalışıp, gerektiğinde de belirli zaman aralıkları ile, sunucu tarafında hizmet sağlayan uretici.aspx ve s.aspx dosyalarını kullanarak, haberleşmek için hazırlanmış fonksiyonları barındırır ve istemci tarafında çalışır.

Sunucu tarafında çalışan uretici.aspx dosyamız, uygulamamızın çalışması sırasında ihtiyaç duyduğu her türlü veriyi, gerektiğinde veri tabanı ile bağlantı kurarak üretir. Örneğin yetki seviyesine göre oluşturulan menüyü, girilecek sınavların listesini yada bir modüle bağlı konuların listesini uygun formatta geriye döndürür. Aynı şekilde sunucu tarafında çalışan s.aspx dosyamız ise veritabanına kayıt ekleme, silme, değiştirme gibi işlemler için ihtiyaç duyulan fonksiyonlardan oluşur. Sınav ekleme, soru değiştirme yada bir sınava öğrencileri atama gibi işlemler buna örnek olarak verilebilir.



Şekil.27 Kullanılan bazı dosyaların birbiri ile ilişkisi



Şekil.28 Veritabanı şeması ve ilişkileri

“**ogrenciler**” tablosu: Bu tablo öğrencilerin temel özelliklerini tutan tablodur. Ad,soyad, öğrenci numarası, T.C. kimlik numarası, sisteme giriş için şifresi, sistemdeki rolü gibi bilgileri bu tabloda tutulur.

“**ogrenci_kayit**” tablosu: Sistemimizde kayıtlar derse değil de modüllere yapıldığı için ayrıca öğrenciler kayıt oldukları module ait bir sınava kayıt yaptıracakları için bu tabloda öğrencilerin modüllerle olan ilişkisi ve varsa kayıt olduğu sınav bilgisi bu tabloda bulunmaktadır.

“sinavlar” tablosu: Sınavlar tablomuz, sınavların modüllere bağlı olmasından dolayı modüller tablomuza bağlıdır. Sınavın adı, başlangıç zamanı gibi bilgiler bu tabloda bulunmaktadır.

“sinav” tablosu : Sınav başlatıldığı anda sınav soruları ve cevap seçenekleri soru bankasından rastgele seçilerek bu tabloya kaydedilir. Öğrenci sınav ekranı bu tablodan oluşturulur.

“sinav_ayarlar” tablosu: bu tabloda sınav için ihtiyaç duyduğumuz çeşitli parametreleri tutuyoruz. Örneğin sınav süresi, hangi konudan kaç soru sorulacağı gibi.

“sorular” tablosu: Sınav oluşturabilmek için ihtiyaç duyduğumuz soru bankası için kullandığımız tablolardan biridir. Bu tabloda bir soru kümelerini saklıyoruz. Bu kümenin elemanları ise “sorular_metin” tablomuzdadır.

“sorular_metin” tablosu : “sorular” tablosundaki bir soruya ait soru metinlerini ve cevap seçeneklerini bu tabloda tutuyoruz. Bu bilgi düz metin halinde değilde HTML şeklinde saklanmaktadır. Bu sayede sorulara ve cevaplara resim ekleyebiliriz yada HTML’in bize sunduğu her türlü yapıyı kullanabiliriz. Ayrıca bir soruya ait birden çok soru metni ve seçeneği olabilmektedir. Sınavlar oluşturulurken bir soruya ait bir soru metni, dört yanlış cevap ve bir doğru cevap, soru kümesinden seçilir ve soru oluşturulur.

“sorular_cevap” tablosu: Öğrencilerin sınavdaki sorulara verdikleri cevapları bu tabloda tutulur.

“fonksiyonlar” tablosu: Uygulamamızda dışarıdan erişilebilecek olan fonksiyonlar birer “id”ye sahiptir. Bu fonksiyonlar kullanılmak istendiğinde yetki kontrolü için fonksiyonun kontrolü için bu “id”ler “fonksiyonlar” tablosunda tutulur.

“roller” tablosu: Uygulamamızda yetkilendirme için kullanacağımız rolleri bu tabloda tutulur

“izinler” tablosu: Bu tabloda hangi rol hangi fonksiyonu kullanacağına dair bilgiler “izinler” tablosunda tutulur.

5.5. Uygulamanın Ekran Görüntüleri ve Kullanımı:

5.5.1. Sisteme giriş:

The image shows a login interface for 'SanalSınavMerkezi'. It features a blue header with the text 'SanalSınavMerkezi | Giriş'. Below the header, there is a yellow padlock icon on the left. To the right of the padlock, there are two input fields: 'Öğrenci Numaranız : 1' and 'Şifreniz :'. Below these fields is a 'Giriş' button. The background of the interface includes a large, stylized 'B' and 'Ü' logo and a set of keys in the top right corner.

© 2007 | Beykent Üniversitesi | Ar-Ge Merkezi

Şekil.29 Giriş Ekranı

Giriş ekranında kullanıcı adı ve şifremizi ilgili alanlara girerek sisteme erişilmiş olur.



Şekil.30 Yönetici Ekranı

Yönetici ekranı Şekil 30 sisteme yönetici olarak girdiğimizde oluşan menüler görülmektedir.

5.5.2. Ders Ekleme

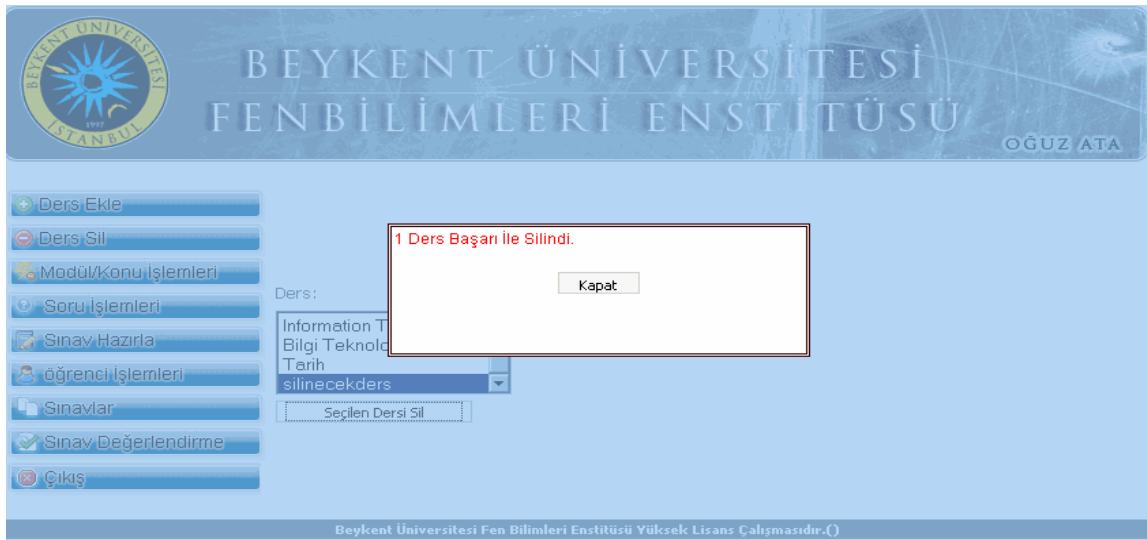
Sol taraftaki menülerden ders ekle menüsü seçildiğinde eklemek istediğimiz dersin adını girceğimiz bir alan gelmektedir. Bu alana dersin adını girip “Dersi Ekle” tuşuna basarak sisteme yeni bir ders ekleyebiliriz.



Şekil.31 Ders Ekleme Ekranı

5.5.3. Ders Silme

Sol menüden “Ders Sil” seçilir. Listelenen dersler arasından silmek istediğimiz dersi seçerek “Seçilen Dersi Sil” tuşuna basılarak ders silinir.



Şekil.32 Ders Silme Ekranı

5.5.4. Modül ekleme/silme

Modül eklemek yada silmek için soldaki menüden “Modül/Konu İşlemleri” seçilir. Açılan ekranda Ders listesi vardır. Modüllerimiz derslere bağlı olduğu için önce ekleyeceğimiz modülün hangi derse ekleneceğini seçeriz. Daha sonra “Modül/Konu Adı:” yazan alana modülün adını gireriz ve “Derse Bu Modülü Ekle” tuşuna basarız. Silmek istediğimiz modül için ise önce dersi seçeriz. Modül alanında o derse ait bütün modüller listelenir. Silmek istediğimiz modülü buradan seçerek “Seçilen Modülü Sil” tuşuna basarak modülü silmiş oluruz.

BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
OĞUZ ATA

Ders Ekle
Ders Sil
Modül/Konu İşlemleri
Soru İşlemleri
Sınav Hazırla
Öğrenci İşlemleri
Sınavlar
Sınav Değerlendirme
Çıkış

Ders: Modül: Konu:

Information Technology I	Bilgisayar Kullanımı ve Dosya Yönetimi	
Bilgi Teknolojileri	Kelime İşlemci	
Tarih	Hesap Çizelgesi	
	Sunum	

Modül/Konu Adı: EKLENECEK MODUL

Derse Bu Modülü Ekle
Seçilen Modülü Sil

Seçilen Modüle Konu Ekle
Seçilen Konuyu Sil

Beykent Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Çalışmasıdır.Q

Şekil.33 Modül Ekleme-Silme Ekranı

5.5.5. Konu Ekleme silme

Konu eklemek yada silmek için soldaki menüden “Modül/Konu İşlemleri” seçilir. Eklemek istediğimiz konunun önce hangi derse ait olacağını daha sonra hangi modüle ait olacağını seçeriz. “Modül/Konu Adı” yazan alana konunun adını girdikten sonra “Seçilen Modüle Konu Ekle” tuşuna basarak seçtiğimiz modüle konu eklemiş oluruz. Konu silmek için de ders ve modül seçilir. Daha sonra bu modüle bağlı konular listelenir. Bu konulardan silmek istediğimizi seçerek “Seçilen Konuyu Sil” tuşuna basarak konuyu silmiş oluruz.

BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
OGUZ ATA

+ Ders Ekle
- Ders Sil
Modül/Konu İşlemleri
? Soru İşlemleri
Sınav Hazırla
Öğrenci İşlemleri
Sınavlar
Sınav Değerlendirme
Çıkış

Ders: Information Technology I
Modül: Bilgisayar Kullanımı ve Dosya Yönetimi
Konu: Bilgisayar Cihazı[8]
Masalı[7]
Dosya Yönetimi[1]
Virüsler[1]

Modül/Konu Adı: EKLENECEK KONU

Derse Bu Modülü Ekle
Seçilen Modülü Sil
Seçilen Modüle Konu Ekle
Seçilen Konuyu Sil

Beykent Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Çalışmasıdır.Q

Şekil.34 Konu Ekleme-Silme Ekranı

5.5.6. Soru işlemleri

Soru işlemleri yapmak için soldaki menüden “Soru İşlemleri” seçilir. Açılan ekranda Ders listesinden işlem yapacağımız ders, o derse ait modül ve o modüle ait konu seçilir. Daha sonra mevcut sorulardan birisi seçilir yada yeni bir soru oluşturulur. Seçilen soru için soru metni, doğru cevap ya da yanlış cevap ekle tuşuna basarak eklenir.

BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
OĞUZ ATA

1.Dersi Seçin: 2.Modül Seçin: 3.Konuyu Seçin:

4.Soruyu Seçin:

5.Soruya ait Metni Seçin:

Soru Çatısı:

Oluştur
Sil

Soru/D.cevap/Y.Cevap

Soru Metni

Ekle
Sil
Değiştir

Path:

Editorü Aç/Kapat

Şekil.35 Soru İşlemleri Ekranı

5.5.7. Sınav hazırlama

Sınav hazırlamak için sol menüden “Sınav Hazırla” menüsü seçilir. Sınavı hangi modül için yapacağımızı seçeriz. Sınav ismini girerek “Sınav ekle” tuşuna basarız.

Seçtiğimiz modül için sınav oluşturulmuş olur. Daha sonra sınav için ayarları yapmamız gerekir.

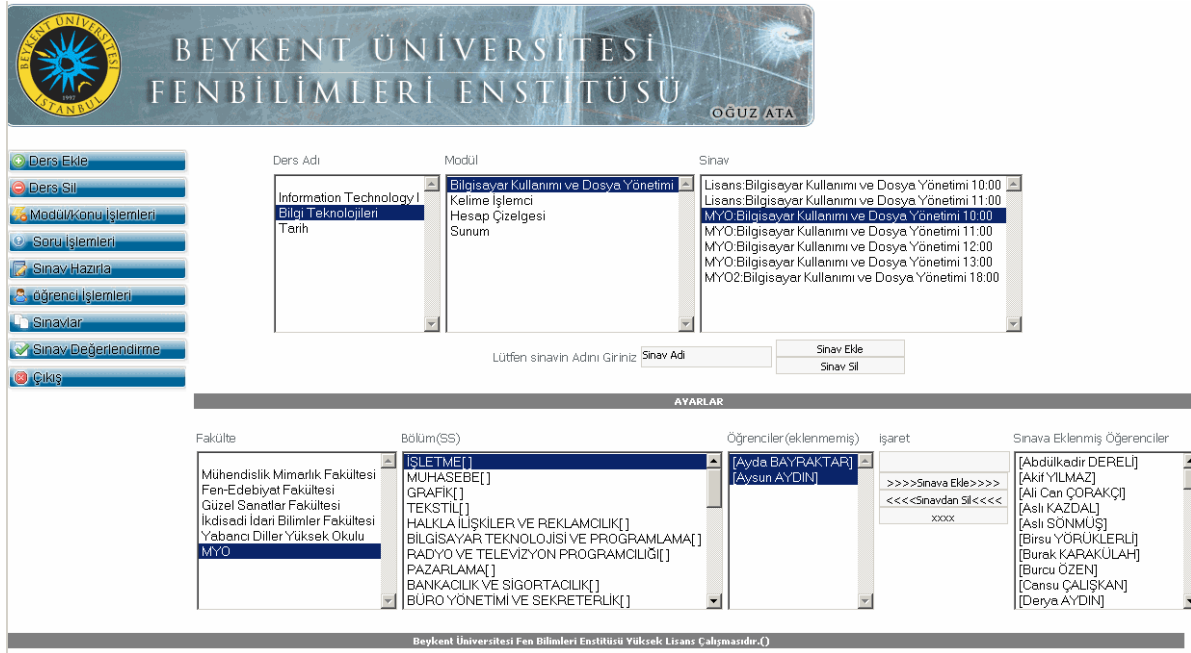
Konular bölümünden, o modüle bağlı konulardan, istediğimiz konuyu ve adedi girerek sınav için soruları oluştururuz. Aynı şekilde sınavın süresini de ilgili alana girerek “Süre Ekle(dakika)” tuşuna basarak ekleyebiliriz. Sınavı başlatmak için ise “Bu Sınavı Başlat” tuşuna basarız.

The screenshot shows the 'Sınav Hazırlama' (Exam Preparation) interface of Beykent University. The header includes the university logo and name. The left sidebar contains buttons for 'Ders Ekle', 'Ders Sil', 'Modül/Konu İşlemleri', 'Soru İşlemleri', 'Sınav Hazırla', 'Öğrenci İşlemleri', 'Sınavlar', 'Sınav Değerlendirme', and 'Çıkış'. The main area has three columns: 'Ders Adı' (Course Name) with a list including 'Information Technology I', 'Bilgi Teknolojileri', and 'Tarih'; 'Modül' (Module) with a list including 'Bilgisayar Ku', 'Kelime İşlem', 'Hesap Çizelc', and 'Sunum'; and 'Sınav' (Exam) with a list including 'Lisans:Sunur', 'Lisans:Sunur', 'MYO:Sunum', 'MYO:Sunum', 'MYO:Sunum', 'MYO:Sunum', and 'MYO2:Sunum'. Below these columns is a text input field for 'Lütfen sınavın Adını Giriniz' (Please enter the exam name) and a 'Sınav Adı' (Exam Name) field. To the right are 'Sınav Ekle' (Add Exam) and 'Sınav Sil' (Delete Exam) buttons. A section titled 'AYARLAR' (Settings) contains three columns: 'Konular' (Topics) with a list including 'Başlangıç[10]', 'Bir Sunum Oluşturma[7]', 'Metin ve Görüntüler[7]', 'Grafik ve Çizim Nesneleri[4]', 'Ekran Gösterisi Efektleri[2]', and 'Çıktı Hazırlama[1]'; 'Soru Sayısı' (Question Count) with buttons for 'Adet Soru Ekle', 'Süre Ekle(Dakika)', 'Başarı Oranı Ekle(yüzde)', 'Bu Sınavı Başlat', and 'AYAR SİL'; and 'Sınav Eklenecekler' (Exams to be added) with a list including '[Sınav Süre=50]', '[SoruAdet=5]Başlangıç', '[SoruAdet=5]Metin ve Görüntüler', and '[SoruAdet=5]Bir Sunum Oluşturma'. The footer of the interface reads 'Beykent Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Çalışmasıdır.(.)'.

Şekil.36 Sınav Hazırlama Ekranı

5.5.8. Öğrencileri sınavlara atama

Hangi sınava hangi öğrencinin gireceğinin ayarlandığı bölüm “Öğrenci İşlemleri” menüsüdür. İşlem yapılmak istenilen sınav seçilir. Seçilen sınava girecek öğrencilerde listeden seçilerek “Sınava Ekle” tuşuna basarak ekleriz. Aynı şekilde sınava girecek öğrencileri sınav listesinden silebiliriz.



Şekil.37 Öğrencileri Sınavlara Atama Ekranı

5.5.9. Sınav Modülü

Öğrenci ilgili sınavı seçtikten sonra ekrana Şekil 36’daki ekran gelir. Burada sınav soruları ve seçeneklerin dışında, sınav süresi ve sınavı bitirdiğini beyan etmesi için “Sınavı Bitir” tuşu bulunmaktadır. Bu ekrandaki sorular ve seçenekler, sayfanın her çağırılışında yer değiştirmektedir.



BEYKENT ÜNİVERSİTESİ
FENBİLİMLERİ ENSTİTÜSÜ

Kullanıcı: Oğuz
Numara: 1
Rol: admin
37.32
SINAVI BİTİR
Log

Beykent Sınav Merkezi

Soru:1 [-]

Aşağıdakilerden hangisi giriş birimi değildir?

a-	☐ Kamera
b-	☐ Klavye
c-	☐ CD- ROM
d-	☐ Yazıcı
e-	☐ Mikrofon

Soru:2 [-]

Bilgisayarın tüm sistem kaynaklarının takılı olduğu elektronik devre aşağıdakilerden hangisidir?

a-	☐ Modem
b-	☐ Klavye
c-	☐ Seri bağlantı Noktası
d-	☐ RAM
e-	☐ Anakart

Soru:3 [-]

Aşağıdakilerden hangisi genellikle modem in bilgi transfer oranı için kullanılır?

a-	☐ GB
b-	☐ Bit
c-	☐ Khz
d-	☐ Bps
e-	☐ Ghz

Soru:4 [-]

Donanım bileşenlerinin kordinasyonu hangi birim tarafından yapılır.

a-	☐ ROM
b-	☐ Merkezi İşlem Birimi(MİB)
c-	☐ BIOS
d-	☐ Hard-disk
e-	☐ RAM

Soru:5 [-]

Bilgisayarın çalışabilmesi için gereken temel işletim sistemi hangisidir?

a-	☐ RAM
b-	☐ Linux
c-	☐ Adobe
d-	☐ Windows XP
e-	☐ BIOS

Soru:6 [-]

Ofis binamızda bulunan bilgisayarları dosya ve yazıcı paylaşımı gibi nedenlerden dolayı birbirine bağlarız. Bu bağlantı şekli için aşağıdakilerden hangisi söylenebilir?

a-	☐ LAN (Yerel Ağ Bağlantısı)
b-	☐ İnternet
c-	☐ ISDN
d-	☐ WAN
e-	☐ ADSL

Soru:7 [-]

Yazılımların oluşturduğu verilerin işlemci tarafından hızlı ve kolayca erişilebilmesi için saklandığı geçici bellek hangisidir?

a-	☐ Hard-Disk
b-	☐ Paralel bağlantı noktası
c-	☐ ROM
d-	☐ RAM
e-	☐ Tarayıcı

Soru:8 [-]

..... özel şirket ağlarında kullanılan internet teknolojileridir.Aşağıdakilerden hangisi boşluk için en uygun seçenektir?

a-	☐ ADSL
b-	☐ World Wide Web(WWW)
c-	☐ Intranet
d-	☐ Dial-up(Çevirmeli Bağlantı)
e-	☐ İnternet

Şekil.38 Öğrencinin Sınav Ekranı

6. SONUÇ

Bu projede güncel web tabanlı programlama teknikleri kullanılarak internet bant genişliğini verimli bir şekilde kullanan ve dinamik soru üretim mekanizmasına sahip bir çoktan seçmeli sınav sistemi oluşturulmuştur. Projede amaçlanan iki hedefe de ulaşılmıştır. Ulaşılan bu hedefler:

- ✓ Bant genişliğinin verimli bir şekilde kullanılması
- ✓ Soru bankasının geliştirilmesi ve yeni sorular üretilmesi kolay bir soru üretim mekanizmasının geliştirilmesi.

'dir. Ayrıca bağlantı kesikliği veya enerji kaybı gibi sorunlar nedeniyle doğabilecek veri kayıpları da bu tez çalışmasında sunulan yöntemlerle minimuma indirilmiştir. Proje

2007-2008 ve 2008-2009 öğretim yıllarında “bilgi teknolojileri kullanımı” dersi muafiyet ve final sınavlarında sınav başına ortalama 1000 kullanıcı üzerinde başarı ile kullanılmış ve elektrik ve geçici bağlantı kopukluklarından kaynaklanan yaşanmış olup herhangi bir veri kaybının olmadığı görülmüştür.

REFERANSLAR

- [1]. J. Keith. Bulletproof AJAX. New Riders, 2007
- [2]. J.J Garrett, A New Approach to Web Applications
<http://www.adaptivepath.com/ideas/essays/archives/000385.php>
- [3]. Wikipedia Ajax (programming). <http://en.wikipedia.org/wiki/AJAX>
- [4]. AJAX Introduction. http://www.w3schools.com/php/php_ajax_intro.asp
- [5]. Wang, H.; Yang, J. "Research and application of web development based on ASP.NET 2.0+Ajax". Industrial Electronics and Applications. ICIEA 2008. 3rd IEEE Conference on 3-5 June 2008 Page(s):857 - 860
- [6]. Bergasa-Suso J., Sanders D.A. Tewkesbury G.E., "Intelligent browser-based systems to assist Internet users", Education, IEEE Transactions on, vol.48, no.4, Nov. 2005.
- [7]. Why use AJAX?. InterAKT Nov.2005. <http://www.interaktonline.com>
- [8]. The Disadvantages of Using AJAX. <http://ajaxwebhosting.lunarpages.com/ajax-disadvantages.php>
- [9]. James Edwards.AJAX and Screenreaders: When Can it Work?
<http://www.sitepoint.com/article/ajax-screenreaders-work> May. 2006
- [10]. Access Control for Cross-Site Requests. World Wide Web Consortium.
<http://dev.w3.org/2006/waf/access-control/>
- [11]. Document Object Model (DOM). <http://www.w3.org/DOM/>
- [12]. DOM. <http://www.w3schools.com/>
- [13]. createElement Method. Microsoft MSDN. [http://msdn.microsoft.com/en-us/library/ms536389\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms536389(VS.85).aspx)
- [14]. Document Object Model. <http://javascript.about.com>
- [15]. Element.removeChild().http://www.mynotlar.com/html_dom/element/removechild.aspx
- [16]. AJAX. [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))