



T.C.
KARABÜK ÜNİVERSİTESİ

Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Lisans Bitirme Projesi

Asp.Net MVC ile Uzaktan Eğitim Sistemi

ALPER KUYUCU

2015210205007

TAHA ABDURRAHİM DURSUN

2015010205026

Bitirme Projesi Danışmanı: Yüksel Çelik

İÇİNDEKİLER

KISALTMALAR.....	iii
ŞEKİLLER	iv
1. GİRİŞ.....	5
1.1. Bitirme Projesinin Amacı	5
2. LİTERATÜR ÖZETİ.....	9
3. KULLANILACAK YÖNTEM VEYA TENKİKLER.....	9
3.1. Mimari (MVC).....	9
3.2. OWIN	12
3.3. Güvenlik	14
3.4. Membership	15
3.5. Identity	16
3.6. Area	16
3.7. Entity Framework	17
4. KULLANILACAK TEKNİK DONANIMLAR, YAZILIMLAR	19
4.1. ASP.NET MVC.....	19
4.2. Web Tasarımı	21
4.3. JQUERY.....	22
5. ZAMAN PLANLAMA	24
6. REFERANSLAR.....	25

KISALTMALAR

MVC: Model View Controller

HTML: Hypertext Markup Language

CSS: Cascading Style Sheets

HTTP: Hyper Text Transfer Protocol

ORM: Object Relational Mapping

XML: eXtensible Markup Language

JS: Javascript

SSL: Secure Sockets Layer

SEKİLLER

Şekil 1. MVC Mimarisi	9
Şekil 2. Routing Kod Yapısı	11
Şekil 3. OWIN Katmanı	12
Şekil 4. Şifreleme Yapısı.....	14
Şekil 5. Asenkron Programlama.....	17
Şekil 6. Entity Framework Katmanları.....	18

1. GİRİŞ

1.1. Bitirme Projesinin Amacı

Günümüzde birçok uzaktan eğitim sistemi bulunmaktadır. Bu sistemler genelde ticari amaç yürüten firmaların ortaya çıkardığı sistemler ve yabancı kaynaklardır. Yeterli yerli ve ücretsiz, kolay erişilebilir bir eğitim sistemi ihtiyacı web tabanlı bu projeyi hayata geçirmek için önemli bir neden olmuştur. Bu projede yer alan katmanlı mimari sayesinde sistem daha düzenli bir hale getirilmiştir. Öğretmen ve öğrenciler ortak bir noktada buluşturulmuştur.

Proje özetle, 3 farklı rol tanımlanmasından oluşmaktadır:

Yönetici, öğretmen ve öğrenci ile ilgili izin ve kontrol mekanizmasına sahiptir.

Öğretmen, kurs oluşturma, ders ekleme, dosya gönderme ve öğrencileriyle iletişim kurma yetkisine sahiptir

Öğrenci, kursa ücretsiz kayıt olup, öğretmenle iletişime geçebilme yetkisine sahiptir.

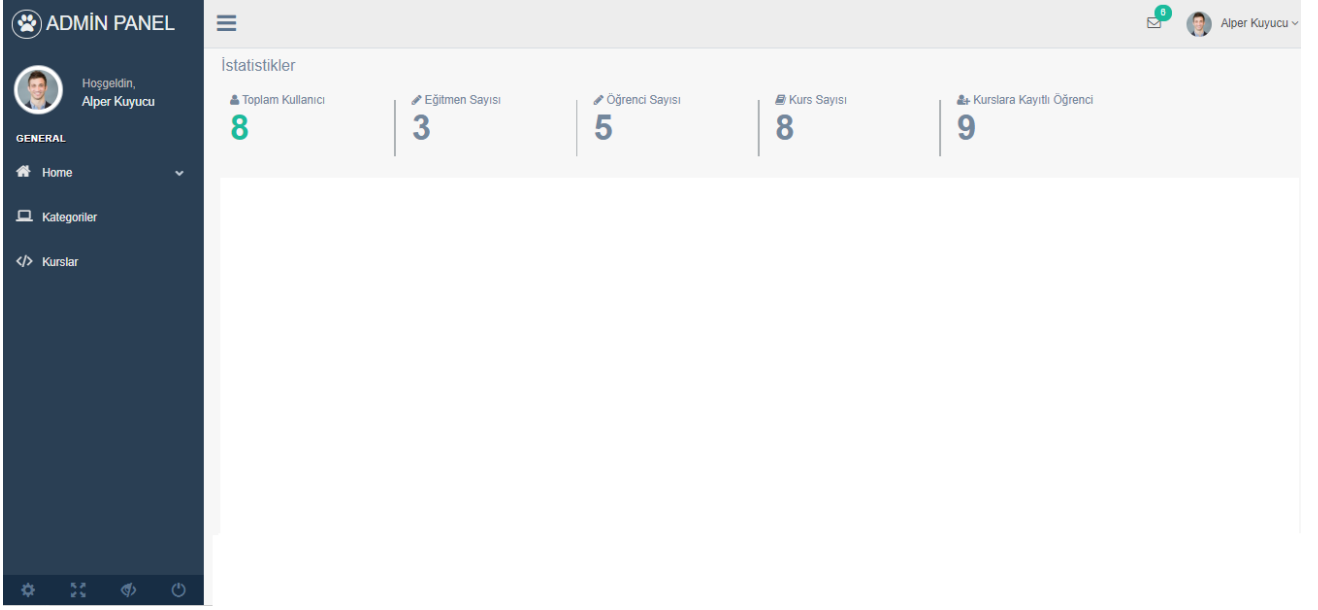
Bu sistem sayesinde öğrenciler kendi dilinde ve ücretsiz olarak istediği kaynağa ulaşabilmektedir. Öğrenciler kurslara kaldığı yerden devam edebilmesi için işlediği dersler ve kalan dersleri görme imkanına kolayca ulaşmaktadır. Öğretmenler ise kurslarına kaç öğrencinin kayıt olduğu, kaç kişinin kursları bitirdiği gibi istatistiklere genel ve kurs bazlı erişebilmektedir.

Projede web teknolojisi olarak Microsoft tarafından geliştirilen ve birçok mimariye sahip olan Asp.Net MVC yapısı kullanılmaktadır. Bu dilin kullanılmasının sebebi günümüzde birçok geliştirici tarafından kullanımı fazlasıyla artan ve yazılım geliştirmede önem taşıyan MVC mimarisi ile uyum içinde çalışmasıdır. MVC mimarisi yazılımcı ile tasarımcıyı ayırarak back-end ve front-end yapısına uyumluluk sağlamaktadır. Bu sayede yazılımcı bir sorunla karşılaştığından bunu kendine back-end üzerinden düzenleyerek işlemlerini rahatlıkla yapabilmektedir. Aynı zamanda modelleme yöntemi sayesinde veri tabanı üzerinden tablo oluşturmak ve bu tablolar arasındaki yönetimi sağlamak için Entity Framework kullanılmaktadır. Bu yöntem sayesinde doğrudan veri tabanı içerisindeki tablolara dokunmadan sadece model üzerinden bir değişiklik yaptığımızda entity framework otomatik olarak veri tabanı üzerinden modelle ilgili tabloyu düzenler.

Proje Kapsamı

Sisteme yeni kayıt olan bir kullanıcı normal bir üye olarak kaydolar. Üye olurken e-mail adresi sisteme kayıtlı ise uyarı verip, farklı bir e-mail adresiyle kayıt olması gerekmektedir.

Yönetici Paneli;



Sistemde yönetici , site ile ilgili verilere ulaşabilir.

Kategoriler bölümünden kategori ekleyip silebilir , gerekli güncellemeleri yapabilir.

Kurslar bölümünde eğitmenlerin yüklediği kursları herhangi bir aykırı durum olmadığına kanaat getirdiğinde kursun paylaşılmasına izin verir.

Öğretmen Paneli;



Yüksel Çelik

Teacher

[Verdiğim Kurslar](#)

[Aldığım Kurslar](#)

[Mesajlar](#)

[Yorumlarım](#)

[Dosyalarım](#)

Instructor Courses

Kurs Ekle

#	Kurs Adı	Kurs AltBaşlık	Acıklama	Kategori	Kurs Güncelle	Ders İşlemleri	Kurs Sil	Dosya Paylaşım	İstatistik
3	Php Symfony	Php Symfony	Php Symfony	Yazılım Geliştirme	Güncelle	Dersler	Sil	Dosya Gönder	Görüntüle
33	Django A'da	Django A'da	Django A'dan Z'e Web Programlama	Yazılım Geliştirme	Güncelle	Dersler	Sil	Dosya Gönder	Görüntüle
35	Django E-Co	Django E-Co	Django E-Commerce Project (ENGLISH)	Yazılım Geliştirme	Güncelle	Dersler	Sil	Dosya Gönder	Görüntüle

Sistemde Eğitimci Kurs ekleyebilir. Eklediği kursu daha sonra gerekli güncelleme işlemlerine tabi tutabilir.

Eklediği kursa derslerini ekleyerek kurs içeriğini istediği gibi genişletebilir. Dersleri daha sonra isteğine göre güncelleyebilir.

Eğitimci kursu alan öğrencilerine dosya paylaşımı yapabilir.

Eğitimci görüntüle kısmından kursu alan öğrencilerin ne durumda olduğuna ilişkin verilere ulaşabilir(Kursu bitiren,devam eden ve kursa kayıtlı öğrenci).

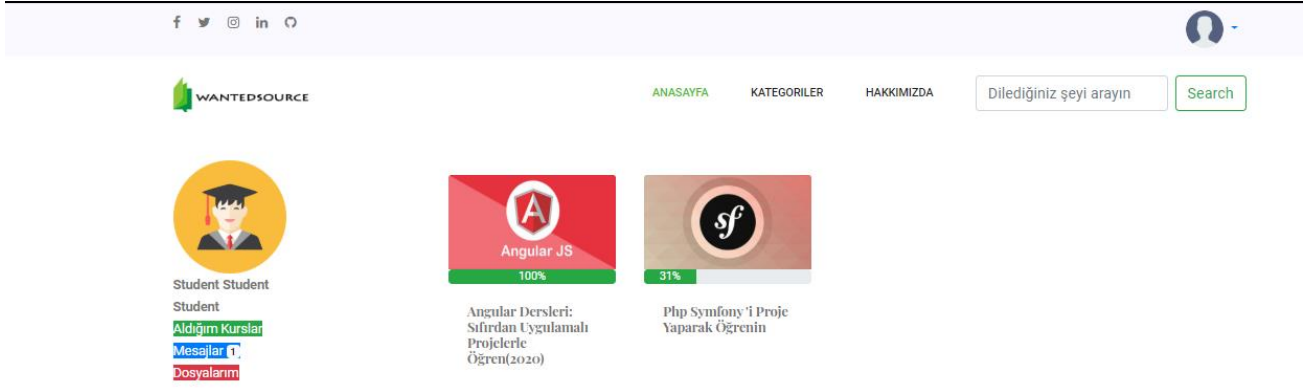
Eğitimci aynı zamanda yeni bir hesap almadan sistemde öğrenci olarak devam edebilir yani istediği kurslara kayıt olabilir.

Eğitimci öğrencileri ile mesajlaşabilir.

Eğitimci derslerine yapılan yorumları bir panelden erişebilir ve ilgili yoruma gidebilir.

Eğitimci de bir öğrenci olduğu için dosyalarım kısmında eğitimci tarafından paylaşılan dosyalara ve kendisinin paylaştığı dosyalara erişebilir ve indirebilir.

Öğrenci Paneli;



Sistemde öğrenci , aldığı kursları listeleyebilir.Kursun ne kadarını tamamladığına dinamik olarak takip edebilir.

Kursun içinde ki derslere erişebilir ve ders hakkındaki yorumlarını yapabilir, yapılan yorumu silebilir ve dilerse eğitmenine bu panelden mesaj gönderebilir.

Mesajlar bölümünden eğitmeniyle iletişime geçebilir.

Dosyalarım bölümünden aldığı kursa dair eğitmen tarafından paylaşılan dosyalara erişebilir.

2. LİTERATÜR ÖZETİ

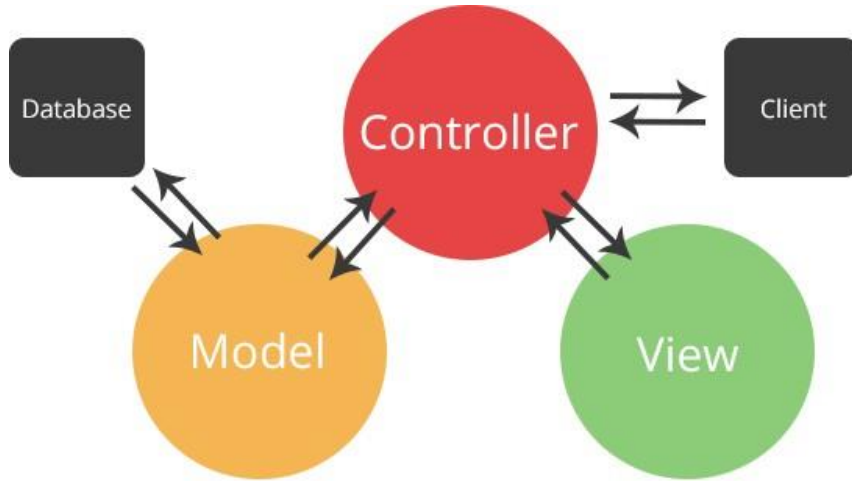
Bu çalışma ile ilgili daha önceden yapılmış sistemler ve kullanılan teknolojiler;

- Web Server: IIS
- Frameworks : ASP.NET
- JavaScript Libraries : jQuery

3. KULLANILACAK YÖNTEM VEYA TENKİKLER

3.1.Mimari (MVC)

[10] MVC, uygulama geliştirmede (özellikle web uygulaması geliştirmede) önemli yere sahip mimari desenlerden biridir. Günümüzde MVC denince akla Microsoft'un geliştirdiği ASP.NET MVC Framework gelmektedir, oysa 1979 yılından beri (Microsoft 1975 yılında kurulmuştur) yazılım dünyasında yer almaktadır. MVC mimarisi Java ve Php gibi diller içinde yer almaktadır.



Şekil 1. MVC Mimarisi

MVC, Model, View, Controller kelimelerinin baş harflerinden oluşur ve her kelime MVC'nin farklı bir katmanını ifade eder.

Model

MVC dünyasında model, uygulama verisinin veya durumunun saklandığı yerdir, genellikle nesne yönelimli programlamada kullanılan nesnelere benzemektedir. Modeller veri tabanında oluşturulacak veri nesnesini model olarak oluşmayı sağlıyor. Aynı zamanda web sayfasında görüntülenecek listeyi yada form elemanlarını da model üzerinden gösteriri. Modeller veritabanı veya xml/json dosyası formatındadır.

Model, veri katmanını (database, xml, json dosyası, vb.) uygulamadan izole eder, böylece diğer katmanlarda veri katmanının neresi olduğunun bilinmesine gerek kalmaz.

Model katmanı sıklıkla Entity Framework, Nhibernate, LLBLGen, vb. gibi araçlar kullanılarak oluşturulur. Bu sayede veri tabanı üzerinden etkileşim sağlanarak veri tabanı yerine modeller üzerinden işlem yapılır..

View

View, istemcinin gördüğü arayüzü içeren katmandır, genellikle Model katmanındaki verinin kullanılması ile oluşturulur. View katmanının Model ve Controller katmanlarından ayrılması ile arayüz değişikliklerinin uygulamanın diğer katmanlarını değiştirmeye gerek kalmadan yapılabilmesi sağlanmıştır.

View katmanında HTML5, CSS3, JavaScript, Angularjs, jQuery gibi son versiyon teknolojiler kullanmak mümkündür. HTML5, CSS3, JavaScript, Angularjs, jQuery ile masaüstü ve mobil tarayıcılarda çalışabilen uygulamalar geliştirmek çok kolaylaşmıştır. Hatta Windows Store uygulamaları geliştirmek için HTML5, CSS3, JavaScript, Angularjs, jQuery teknolojilerinden yararlanılabilir.

Controller

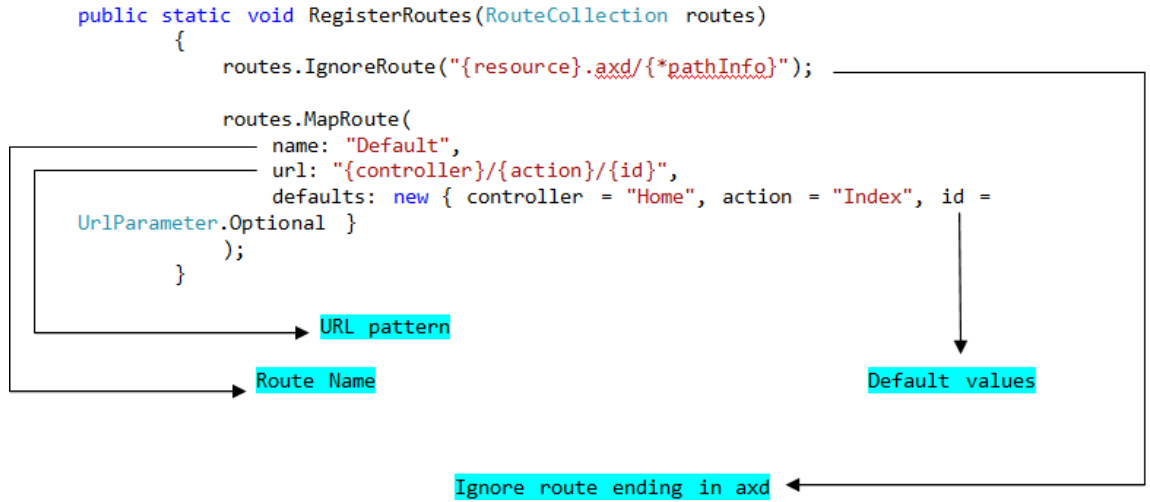
Controller, istemciden gelen isteği işlemek, Model ve View katmanları arasında köprü olmak gibi görevleri yerine getirir. Controller içerisinde bir veya daha fazla Action olabilir, genellikle her Action bir web sayfası üretmek için kullanılır. Action içine yazılan komutlar sayfa üzerinde oluşacak işlemleri tetiklemeyi sağlar. Bu sayede listeleme,

oluşturma, düzenleme, detayı görüntüleme ve silme gibi önemli fonksiyonları oluşturmada kolaylıklar sağlar.

Controller'ın en iyi yapı taşı routing'dır.

Routing

Routing, istemci'nin uygulamaya yaptığı isteği uygun Controller ve Action'a yönlendiren yapıdır. İstemci, isteği uygulamanın belli bir adresine gönderir, routing mekanizması sayesinde ilgili adres için en uygun Controller ve içerisindeki Action tespit edilir ve çalıştırılır. Aşağıdaki resimde görüldüğü gibi Route name bizim ana area alanımızı belirtir. Url pattern ise, hangi controller üzerinden hangi action'a ait sayfa açılacağını belirtir. (Örnek olarak; "home/Index") id ise id değerini göndermeyi sağlar. Bu sayede ilgili id'e ait data üzerinden işlem yapılabilir. (örneğin; "home/edit/1")



Şekil 2. Routing Kod Yapısı

Gerçek hayattan bir örnek

[11] Satranç uygulamasını ele alırsak;

Model'de oyunun "durumu" barındırılacaktır. Oyunun durumunu değiştirecek etkiler (örneğin bir taşın hareketinin doğru olup olmadığı) veya oyunun bitip bitmediği gibi bilgiler model üzerinde yer alacaktır. (Kısaca, oyunun tüm bilgileri ve yapılacak işlemlerin validasyonu Model üzerinde barınacak)

View kısmında satranç tahtasının görünümü, yönettiğimiz piyonların şekilleri ve piyonları hareket ettirdiğimizde hangi piyonun nereye gittiğini söyleyen bildirimler yer alacaktır.

Piyonların nasıl hareket ettiđi, oyunun durumuyla ilgili mantıksal bilgilerin View ile hiçbir işi olmayacaktır. View sadece ve sadece görselliđi barındıracaktır. Kullanıcının gördüğü kısım sadece burasıdır. Diğer katmanlardan haberi yoktur.

Controller ise View ve Model arasında haberleşmeyi sağlayacaktır. Örneđin, kullanıcı View’da yer alan “Yeni oyun başlat” butonuna bastığında Controller, Model’e giderek böyle bir isteđin geldiđini söyleyecektir. (Tüm bu işleri yapan Model olacaktır, Controller’ın amacı böyle bir isteđin geldiđini ve alakalı isteđin detaylarını Model’e iletmektir) Bu sayede Controller yeni oyun için gerekli kodları çalıştırarak yeni oyunu başlatacaktır. Kullanıcı her oyun hamlesinde gerekli controller, modeller üzerinden bilgiyi alıp gerekli işlemleri yerine getirerek view’e son halini iletecektir.

Önemli noktalar

Controller bizim için yalnızca aracı görevi görüyor. İş mantığı Model’de, görsel mantık View’da olmalı, Controller sadece haberleşmeyi sağlamalı. Controller’a, Model’in veya View’ın sorumlulukları yüklenirse MVC kullanmanın hiçbir anlamı yok.

Yukarıda bahsettiğimiz sebepten ötürü Fat Model, Skinny Controller yapısını kurgulamamız lazım. Algoritmayı Controller’a sıçartırsak eđer MVC kullanmamızın hiçbir kazancı kalmıyor.

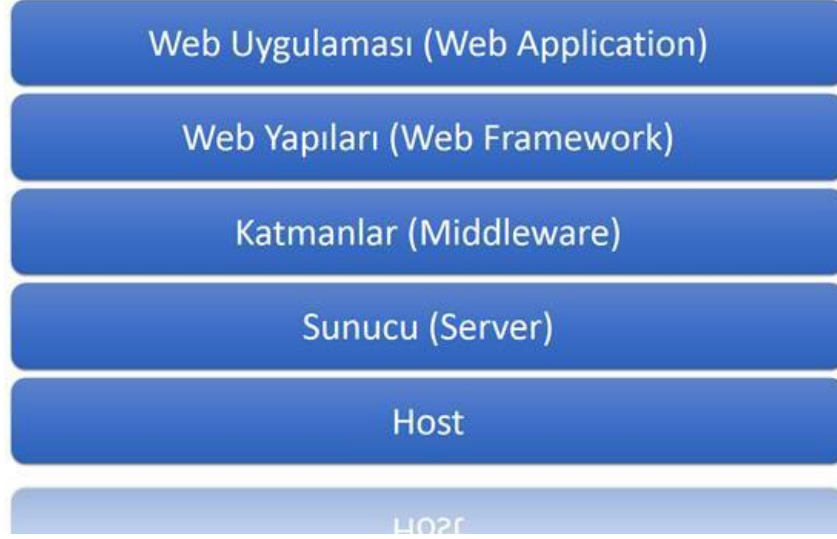
Pek çok MVC nedir makalesi Web tabanlı projelerden örnek verse de, Desktop/Mobil uygulama geliştirirken de MVC kullanılabilir. Zira MVC bir mimari biçimdir. (Örneđin iOS geliştiriciler iPhone uygulaması geliştirirken MVC modelini kullanır)

3.2. OWIN

[6] OWIN, Open Web Interface for .Net (.Net için açık web arayüzü) kelimelerinden gelmektedir. Bir teknoloji olmaktan ziyade bir standart olan OWIN, .Net web sunucuları ile yine .Net ile geliştirilmiş web uygulamalarının birbirleriyle nasıl iletişim kuracağını anlatmakta, bir arayüzü tanımlamaktadır.

OWIN tanımladığı standart ile uyumlu geliştirilen web sunucuları, IIS gibi kompleks ve pek çok senaryoyu aynı anda destekleyen sunucuları yerine, geliştiricilerin istediğı bileşenleri uygulamaların ihtiyaçları doğrultusunda birleştirdiğı ve yalın HTTP’nin hemen

üzerinde yer alan ince bir katman olarak düşünülebilir. Sadece geliştirici olarak bizlerin ihtiyacı olan modüllerin yüklenmesi beraberinde önemli bir performans kazanımı getirmekle birlikte, aynı zamanda sunucu yüzeyini küçülterek olası saldırıların sayısını da azaltacaktır. Buda ekstra bir güvenlik önlemi sağlayacaktır.



Şekil 3. OWIN Katmanı

OWIN, temelde host, server (sunucu), middleware (katmanlar) web frameworks (web yapıları) ve web application (web uygulaması) rolleri etrafında toplanmıştır;

Host : Uygulamayı başlatmaktan sorumlu olan süreç/yapıdır. IIS gibi bazı web sunucuları hem host hem de server rollerini üstlenmektedirler. Bir web sitesinde yayınlanmak istenen sayfaların, resimlerin veya dokümanların internet kullanıcıları tarafından erişebileceği bir bilgisayarda tutulmayı da sağlar.

Server (Sunucu) : İstemcilerle HTTP üzerinden doğrudan iletişim içerisinde olan yapıdır. Sunucu rolü HTTP protokolü üzerinden gelen isteklerin takip eden rollerce işlenmesinden ve çıktının istemciye iletilmesinden sorumludur.

Middleware (Katman) : Sunucu ile uygulama arasında yer alan ve denetleme, yetkilendirme, doğrulama, yönlendirme v.b. özel işlemleri gerçekleştiren rollerdir.

Web Frameworks (Web Yapıları) : Web uygulaması geliştirme sırasında gerçekleştirilen ortak işlerin daha hızlı ve kolay gerçekleşmesinden sorumlu rollerdir.

Web Application (Web Uygulaması) : OWIN uyumlu sunucularda ve web yapıları üzerinde özel bir amaç için geliştirilmiş uygulama rolleridir.

3.3. Güvenlik

Authentication

[7] Authentication bir kullanıcının web sitemizde bir sayfaya erişme yetkisinin olup olmadığını anlamak ve erişime izin vermek demektir. Sadece Asp.Net MVC ya da web için değil herhangi bir sistem için de geçerli bir durumdur. Bir isteğin erişme yetkisinin olup olmadığına karar vermemiz her zaman gerekir.

Authentication, Son kullanıcıya gösterilecek olan sayfaların kimler tarafından görüntülenebileceği hangi roller yardımı ile görüntülenebileceği konusunda yapılan güvenlik işlemlerine de denmektedir.

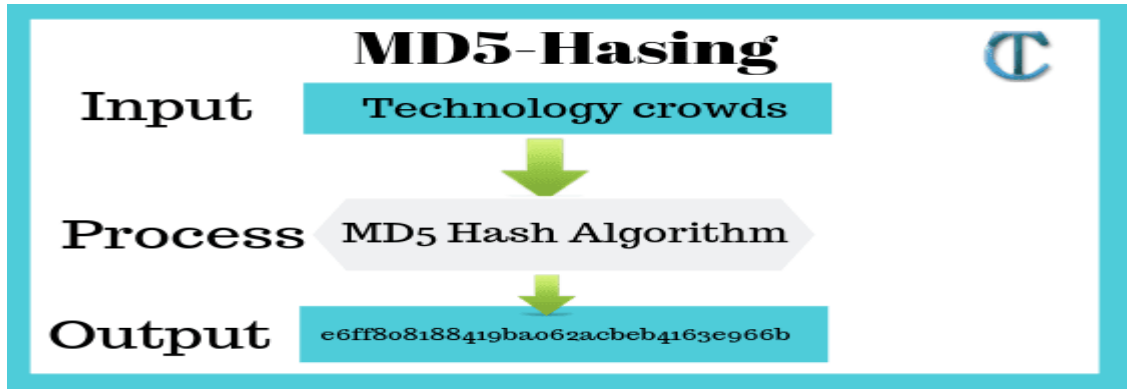
ASP.NET Membership sisteminde log-in/log-out işlemleri için Forms Authentication kullanılırken, **ASP.NET Identity** OWIN Authentication kullanır. Cookie yönetimi Membership'te Forms Authentication ile gerçekleştirilirken OWIN'de Cookie Authentication ile sağlanır.

Authorization

Sisteme giriş yapan kullanıcının izin verilen işlemleri gerçekleştirmesi kontrol etme işlemi burada gerçekleşir. Üye girişi yapmayan ya da yetkisi olmadan bir sayfaya girmek isteyen kullanıcının erişiminin annotation ile engelleyerek giriş yapması için giriş sayfasına yönlendirme yapılır yada mesaj döndürülür. Authorize metodu ile üye tiplerine göre sayfa giriş yetkilendirmelerini ayarlayabilir yada üye olmayanları özel sayfaya girişini engelleyebiliriz. Üye tipleri dışında belirli üyeler için özel sayfalarda oluşturulabilir. Bu üyelere göre özel sayfalar oluşturulabilir.

Şifreleme (Encryption)

Gizlenmek istenen bir bilginin (metin, fotoğraf, ses kaydı, kişisel bilgiler vb.) bir algoritma yardımıyla bir başkası tarafından okunmasını ya da değiştirilmesini engellemek için veri üzerinde yapılan işleme şifreleme denir. Şifrelenen bir veri ancak şifreleme işleminde kullanılan anahtar ya da bu anahtara karşılık gelen ikincil anahtar bilindiğinde açılabilir. Şifreleme İnternet üzerinde yapılan veri transferlerinde çokça kullanılan bir güvenlik yöntemidir. Eticaret sitelerinde de kullanıcıların verileri (kişisel bilgileri, kredi kartı bilgileri) SSL sertifikalarıyla şifrlenerek üçüncü şahıslar tarafından erişilmesi engellenir.



Şekil 4. Şifreleme Yapısı

Kullanıcı şifre (password) bilgilerini şifrelemek için kullanılan yöntem MD5,SHA1 ve SHA2 algoritmasıdır. Projemde kullandığım MD5'tir. "Hash" fonksiyonlarına dayalı veritabanı yönetimine (database management) imkan sağlar.

Özellikleri

- MD5 algoritması, e-posta şifreleme uygulamaları, güvenli uzaktan erişim uygulamaları, özel bilgisayar ağları ve daha bir çok alanda kullanılabilir.
- Günümüzde güvenliği arttırmak amacıyla MD5ve SHA1 algoritmaları birbiri ardına kullanılarak veriler şifrelenir.

.NET teknolojisi ile yazılım geliştirilirken, herhangi bir veride şifreleme yapmak için .NET Framework içerisinde yer alan System.Security.Cryptography kütüphanesi kullanılır. Bu kütüphane içerisinde yer alan fonksiyonlar sayesinde, yazılımcının isteği doğrultusunda güvenli bir şekilde veri şifreleme ve şifre çözümü yapılabilir. SHA1 algoritması kullanılarak şifreleme yapılabilmesi için SHA1CryptoServiceProvider sınıfının kullanılması gereklidir.

3.4. Membership

ASP.NET Identity ilk olarak 2013 yılında tanıtılan ve ASP.NET Membership'in devamı sayılabilecek bir framework'tür. Membership, geliştirilen bir web sitesinde üyelik işlemleriyle ilgili gereksinimleri çözen, Forms Authentication mantığıyla çalışan ve SQL Server'da kullanıcıyla ilgili bilgileri (username, password vb) saklayan bir yapıdır.

Bununla birlikte uzun yıllar geliştiricilerin yükünü hafifleten bu sistem geçen süre zarfında web teknolojilerindeki değişimle birlikte bazı gereksinimleri karşılayamaz hale gelmiştir.. SQL Server üzerinde yer alan veri tabanı şablonunun değiştirilememesi, Facebook, Twitter ya da Google gibi dışarıdan sistemlerle authenticate olmaya izin vermeyen yapısı belirttiğimiz yetersizliklere örnek gösterilebilir.

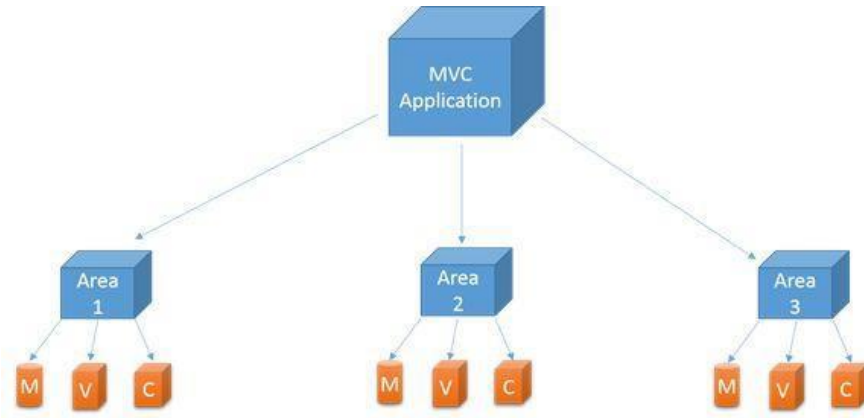
3.5.Identity

[13] Membership'in yukarıda belirttiğimiz olumsuzlukların yanı sıra, uzun yıllar boyunca kullanım sırasında geliştiricilerin geri dönüşlerini de dikkate alan ASP.NET ekibi bizlere daha esnek, dinamik ve daha sosyal bir authentication sistemi hazırladı.

ASP.NET Identity default olarak SQL Server üzerinde çalışır ve Code First yaklaşımı ile hayata geçirilir. ASP.NET Identity'nin yeniliklerini başlıca şu şekilde sıralayabiliriz:

- ASP.NET MVC, Web Forms, Web Pages, Web Api başta olmak üzere dilediğimiz ASP.NET Framework'ünde ya da mobil, Windows Store ve hybrid uygulamalarda kullanılabilmesi
- Veri tabanında yer alan tabloların genişletilebilmesi (kullanıcının doğum tarihini de tutmak isterseniz tabloya bunun için bir kolon ekleyebilirsiniz)
- İstedığınız kadar role eklenmesi (Admin, operator, user vb)
- Facebook, Twitter, Google hesaplarıyla entegre üyelik işlemleri gerçekleştirilebilmesi

3.6. Area



Şekil 6. Area Mimarisi

Area aynı proje içerisinde farklı birimler (roller) tarafından kullanılacak olan kısımları diğerlerinden ayırmak için kullanılan bir yapıdır. Bu yapı sayesinde istenilen bölümlere diğerlerinden bağımsız olarak kurallar tanımlanabiliyor. Özellikle içerik yönetim sistemlerinde siteyi ziyaret edecek kişilerin görüntüleyeceği sayfa ile site içeriklerini düzenleyecek kişilerin görüntüleyeceği sayfa farklıdır. Sayfayı ziyaret edecekler için oluşturulan proje içerisine bir Area ekleyerek sayfa yöneticisi yani admin sayfasını bu area içinde yapılabilir. Bu sayede iki birim ayrılmış olur. Normal proje içinde yer alan Mvc mimarisi aynı şekilde Area içerisinde de yer almaktadır. Bu sayede hem kullanıcı sayfasını hem de admin sayfasının ayırırken her iki birim içinde tasarım ve kodlama kısımlarını ayırmış oluruz.

3.7. Entity Framework

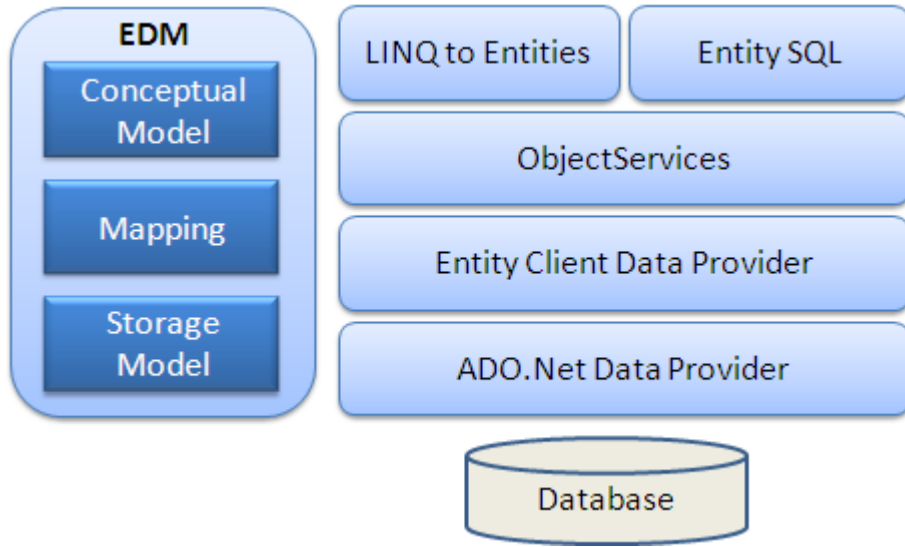
[3] Entity Framework Microsoft tarafından geliştirilen ve yazılım geliştiricilerin katı sql sorguları yazmalarını ortadan kaldırarak bir ORM imkanı sağlayan framework'tür. ORM ise ilişkisel veritabanı yönetim sistemlerine direkt olarak müdahale yerine nesneler aracılığı ile müdahale edilmesini sağlayan bir köprüdür diyebiliriz.

Piyasada bir çok ORM Framework'leri bulunmaktadır. Örnek olarak; DataObjects.Net, NHibernate, OpenAccess, SubSonic etc. Entity Framework vs..

Entity framework ile birlikte LINQ (Language Integrated Query) sorgularını kullanarak nesneler üzerinde güçlü bir sorgulama imkanına sahip oluruz. Entity framework'ün bize sunduğu hizmetler başlıca change tracking, identity resolution, ve query translation.

Entity framework'ün aslında temel amacı uygulama geliştiricinin data işlemleri ile çok uğraşmadan uygulama tarafına odaklanmasını sağlamaktır. Çok basit bir örnek olarak, klasik ADO.NET uygulamalarında bir bağlantının açılmasından ve kapatılmasından tamamen biz geliştiriciler sorumludur. Ancak entity framework bu işlemleri geliştirici yerine yapmaktadır. Sorguyu hazırlar ver entity framework aracılığı ile bunu database'e iletir. Bu sayede sql sorguları ve sql bağlantıları olmadan direk olarak Linq üzerinden data işlemleri yapılabilir. Bu sayede sql sorguları ve sql bağlantıları olmadan direk olarak Linq üzerinden data işlemleri yapılabilir.

Entity Framework Mimarisi



Şekil 7. Entity Framework Katmanları

EDM (Entity Data Model): EDM 3 bölümden oluşur. Conceptual Model, Mapping, Storage Model.

Conceptual Model: Bu alanda model sınıflarımız ve bu sınıfların ilişkileri yer alacaktır. Bu sınıflar veritabanı tasarımınızdan bağımsız olacaktır.

Storage Model: Bu alanda veritabanı tasarım modelimiz yer alır. Bu model içerisinde veritabanımıza ait tablolar, view'lar, stored procedure'ler ve bunlara ait ilişkiler ve key'ler yer alır.

Mapping: Bu alan ise model sınıflarımız ile tasarım modelimiz arasındaki haritalama işlemlerinin bilgilerinin tutulduğu alandır.

LINQ to Entities: Nesneleri sorgulamada kullanacağımız sorgulama dilidir. Bu sorgular bize model sınıflarını döndürecektir. Bu alanda LINQ'in yeteneklerini de kullanabiliriz.

Entity SQL: Yine LINQ to Entities gibi sorgulama yapabileceğimiz bir sorgulama dilidir. Ancak Linq To Entities'e göre daha zordur.

Object Services: Veritabanından sorgulama sonucunda alınan verilerin geri döndürülme aşamasında gereken convert (dönüştürme) işlemlerinin yapıldığı alandır.

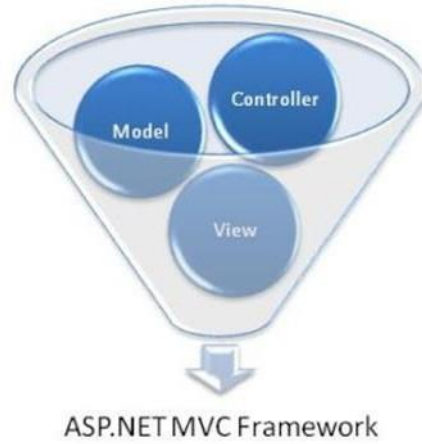
Entity Client Data Provider: Bu alanın sorumluluğu hazırlanan LINQ to Entities yada Entity SQL sorgularını ADO.Net Data Provider'a gönderilmek üzere anlaşılır SQL sorgularına dönüştürmektir.

ADO.Net Data Provider: Bu katman ise standart ADO.NET kullanarak veritabanı ile iletişim kurar.

4. KULLANILACAK TEKNİK DONANIMLAR, YAZILIMLAR

4.1. ASP.NET MVC

[12] Asp.Net MVC, Microsoft'un Asp.Net teknolojisi üzerine MVC tasarımını giydirmesi ile oluşturduğu yeni bir yapıdır. Bu yapı sayesinde büyük çaplı projelerde (eğer desene uygun şekilde çalışılırsa) projenin yönetimi kolaylaşacak, yazılımın performansında artış görülebilecek, yazılan kodların yeniden kullanımı (reusable) artacağından hem daha hızlı yazılım geliştirme yapılabilecek, hem de testleri daha kolaylaşacaktır. Asp.Net MVC öğrenmek ve bu teknolojiyle yazılım geliştirmek için daha bir çok neden sıralayabiliriz.



Asp.Net MVC, Asp ile başlayan Microsoft web platformu yazılım geliştirme teknolojilerinin şu anda geldiği son noktadır. Günümüzde Asp teknolojisini kullanan proje sayısı gitgide azalmakta ve projenin kapsamına, içinde kullanılması gereken teknolojilerine, yazılım geliştiricilerin alışkanlıklarına ve bazı gereksinimlere göre Asp.Net ve Asp.Net MVC teknolojilerinden birisi kullanılmaktadır.

Asp.Net vs Asp.Net MVC

[8] “Asp.Net Web Forms ve Asp.Net MVC teknolojilerinden hangisini seçmeliyiz ve neden?” sorusu, yazılım geliştiricilerin sıklıkla sorduğu bir sorudur. Proje ve altyapı (donanım, yazılım, yazılım geliştirici vs) hangi teknolojiye daha uygunsa onunla yazılım geliştirmenizde fayda vardır. Asp.net kendine has yapısı ve özellikleri varken Asp.Net MVC ise açık kaynak sistem kullanan web programcılarının sıklıkla karşılaştığı bir yapıya benzemektedir. Buda kolay kullanım sağlamaktadır. MVC mimari sayesinde işlemler hızlıca yapılmaktadır.

Neden ASP.NET MVC tercih etmeliyim?

MVC ile istemci'nin isteğine karşılık üretilen çıktı üzerinde çok büyük kontrol imkanı vardır. Bu sayede her alanına müdahale edebildiğimiz ve isteğe en uygun çıktının üretilmesi sağlanmıştır.

MVC ile tekrar kullanılabilir (reusable) kod üretmek mümkündür. MVC'nin katmanları birbirinden ayrıldığı için her bir katmanın başka projelerde kullanılabilmesi sağlanmıştır.

MVC ile istemci'nin istek göndereceği adresler üzerinde çok büyük kontrol imkanı vardır. Bu sayede adres'in içerik ile tam bir ilişki içerisinde olması sağlanmıştır. Arama motorları için adres-içerik ilişkisi önemli olduğu için uygulamanın bulunabilirliğine katkısı yüksektir.

MVC ile test edilebilir uygulamalar geliştirme çok kolaylaşmıştır. Katmanların birbirinden ayrı olması ve Test Driven Development yapmayı kolaylaştırıcı mimarisi sayesinde test edilebilir uygulama geliştirilebilmesi sağlanmıştır.

[9] Asp.Net MVC'nin Asp.Net Forms'a üstünlük sağladığı konular:

Performans

Asp.Net MVC'de ViewState kavramı bulunmadığı için durum yönetimi yazılım geliştirici tarafından yapılmak durumundadır. Web Forms'ta durum yönetimleri otomatik olarak yapıldığından dolayı sayfaların performansı Asp.Net MVC sayfalarına göre daha düşüktür.

Reusabisay (Yeniden Kullanılabilirlik)

Asp.Net Web Forms'ta sayfa içerisine yazılan kodlar (sunucu tarafı) o sayfaya ya da kontrole özgü olmak durumundadır. Asp.Net MVC'de yazılan kodlar ise kontrollerden tamamen bağımsız olduğu için birden çok yerden aynı kod çağırılıp kullanılabilir. Bu da kodların yeniden yazılmasının önüne geçer.

Test Driven Development

Asp.Net MVC’de yazılan kodlar kontrollere ya da sayfaya özgü olmadığı (bağımsız yazıldığı) için kolaylıkla test edilebilir durumdadır. Bu sebeple Asp.Net MVC ile geliştirilen projeler Test Driven Development’a uygundur.

Yazılım Geliştirme Süresi Daha Kısa

Katmanlar birbirinden farklı olduğundan farklı yazılım geliştiriciler farklı katmanlarda aynı işlem için eş zamanlı olarak kodlama yapabilirler. Örneğin bir kişi yapılacak işlemin Modelini kodlarken biri Controllerı bir diğeri de View katmanını eş zamanlı olarak kodlayabilir. Asp.Net Web Forms’ta bir işlem için sadece bir kişinin kodlama yapması uygundur.

Kontrollerin Yönetimi İçin Client Side Development (İstemci Taraflı Geliştirme)

Sayfa içerisine yerleştirilen HTML kontrollerinin yönetimi ve sunucuya istek gönderimi için Asp.Net MVC’de Javascript ve Javascript tabanlı istemci taraflı teknolojiler kullanılmaktadır.

Sonuç

Sonuç olarak MVC uzun yıllardır bir çok projede kullanılmış ve (yönetimi ve yazılım geliştirme ekibi başarılı ise) başarılı olmuş bir desendir. Bu desen Asp.Net teknolojisiyle birleşip gün geçtikçe gelişen bir yapı (Asp.Net MVC) haline gelmiştir.

Asp.Net Web Forms ve Asp.Net MVC teknolojileri arasındaki seçim için, yukarıda da bahsedildiği gibi projeleriniz ve altyapınız (donanım, yazılım, yazılım geliştirici vs) hangi teknolojiye daha uygunsa onunla yazılım geliştirmenizde fayda vardır.

4.2. Web Tasarımı

Herhangi bir kurumun kurumsal kimliğinin yansıtılması ve tanıtım faaliyetlerinin gerçekleşmesi amacı ile ulusal ve uluslararası ağ ortamında yayınlanan online sistemlere web sitesi ve bu sitelerin oluşturulma sürecine web tasarımı denir.

Metin, video, ses kaydı veya fotoğraf gibi tüm işitsel ve görsel araçları kullanarak oluşturulan parçalar web sayfalarını, aynı sistem altındaki bütün web sayfalarının bir araya gelmesi ile oluşan bütün de internet sitesini oluşturur.

Bootstrap (Tasarım framework)



[2] Bootstrap, front-end kodlamayı kolaylaştıran ve hızlandıran ücretsiz bir framework’tür. İçerisinde HTML, CSS ve JS kullanılarak geliştirilen yazı biçimleri, formlar, butonlar, tablolar, menüler, modal yapılar ve bir çok yapı bulunmaktadır. Bu

yapının en önemli özelliği responsive (mobile ve tabletlere uyumlu) olmasıdır. Geliştirilen web sitesinin, görüntülediği tüm cihazlarda otomatik olarak en uygun görünümü almasını sağlar. Mobil uyumluluğu çok daha ön plana çıkmasıyla birlikte daha fazla önem kazanmıştır.

Bu yapının tercih edilme nedeni basit düzeyde html ve css bilgisi olan herkesin kullanabilmesidir. Responsive yapısı ile mobil cihazlardan bile yazılar net bir şekilde okunabilir. Mobil öncelikli yapısı vardır. Tüm modern tarayıcılarına uygun olarak çalışır.

4.3. JQUERY

[1] jQuery bir Java Script kütüphanesidir. 2006 yılına kadar çok fazla karmaşık Java Script kütüphanesi kullanılıyordu. John Resig ise jQuery'yi geliştirerek bu Java Script kütüphanelerini daha sade ve anlaşılır bir hâle getirdi. Ardından web geliştiricilerinin hizmetine sundu. Birkaç sene içerisinde de jQuery tüm dünyada popüler oldu.



[4] jQuery'nin sloganı "az kod çok iş"tir. Bu şekilde birkaç kod ile animasyonlar, slider efektleri ve gizleme - gösterme efektleri yapılabilir. HTML5 ve CSS3 desteği oldukça fazladır. Bugün birçok web sitesinin neredeyse tamamı jQuery kullanmaktadır. JQuery kütüphanesi bedava ve açık kaynaklıdır. Projelerde kullanımı hızlı bir şekilde artmaktadır. Bununla birlikte ilk kullanıldığı günden bugüne büyük bir gelişim göstermiştir ve hala gelişmekte olup yeni versiyonları yayınlanmaktadır.

Avantajları

- Nesne seçimlerini ve etkilenmelerini hızlandırır.
- Html içinde animasyon yapımını kolaylaştırır.
- Css ile birlikte kusursuz çalışır.

AJAX

Ajax, “Asynchronous JavaScript and XML” (Türkçe: Eşzamansız JavaScript ve XML) anlamına gelen ve bir çok programlama dili ile uyumlu çalışan bir tekniktir. Amaç; bütün sayfayı kullanıcıya tekrar yükletmeden, sadece gerekli olan veriyi dinamik olarak ekrana getirmek veya sunucuya veri göndermektir. Örneğin iletişim formu doldurulduğunda sadece datayı göndererek o bölüme sonucu yazdırmak için kullanılabilir.

Ajax hemen hemen her web projesinde kullanılabilir ancak desteklemeyen tarayıcılar içinde bir versiyon oluşturulması gerekir. Gelişmiş arama motorları ajax içeriğini okuyabilirken bazıları kaynak kodlarında gözükmeyen bu içeriği okuyamaz.

Sadece istenen kısmın yüklenmesi hız açısından bir avantajken, kodların kullanıcı bilgisayarını tarafından yorumlanması sebebiyle yavaş bilgisayarlar sorun yaşamaktadır. Ajaxı kullanmak ana web programlarının yanı sıra iyi bir javascript bilgisi gerektirmektedir.

5. ZAMAN PLANLAMA

Tablo 1. Çalışma Zaman Planı

1. Araştırma
1.1 Kullanılacak Teknolojiler
1.2 Örnek Sistemleri İnceleme
1.3 Proje Analizi
1.4 Model Çıkarma
1.5 Temel Görevlerin Belirlenmesi
2. Veri Tabanı (VT) Tasarımı
2.1 Modelleri Tasarımı
2.2 Modellerin İlişkilendirilmesi
3. Web Tasarımı
3.1 Tasarım Kalıpların Çıkarılması
3.2 Kullanıcı Arayüzlerin Çıkarılması
3.3 JavaScript'lerin Yerleşimi
3.4 Performansa Yönelik Tasarımlar
4. Programlama
4.1 Kullanıcı İşlemleri
4.2 Güvenlik
4.3 Yetkilerin Ayarlanması
4.4 Arayüzlerin Hazırlanması
4.5 Yetkilere Göre Görev Dağılımı
4.6 Gerekli Değişikliklerin Yapılması

6. REFERANSLAR

[1] <https://jquery.com>

[2] <http://trbootstrap.com>

[3] <http://oguzkurtcuoglu.com/blog/post/entity-framework-makale-serisi---entity-framework-nedir>

[4] <http://www.ismailgursoy.com.tr/jquery-nedir/>

[5] <https://ferhatkortak.wordpress.com/2014/06/11/c-asenkron-programlamaasync-ve-await-anahtar-kelimeleri/>

[6] <http://www.enterprisecoding.com/post/yeni-baslayanlar-icin-owin>

[7] <http://www.csharpnadir.com/articles/read/?id=992>

[8] <http://koddit.com/yazilim/mvc-nedir-gercek-orneklerle-mvc-nedir-anlayalim/>

[9] <http://www.ugurkizmaz.com/YazilimMakale-1868-MVC-Pattern-ve-Asp-Net-MVC-Nedir--Asp-Net-MVC-vs-Web-Forms.aspx>

[10] <http://www.borakasmer.com/asp-net-mvc-nedir-ne-ise-yarar/>

[11] <http://www.iztim.com/blog/yazilimteknolojisi/mvc-nedir>

[12] <https://www.asp.net/mvc>

[13] Veysel UĞUR KIZMAZ, ASP.NET MVC5, Kodlab Yayıncılık