



**BURSA TEKNİK ÜNİVERSİTESİ MÜHENDİSLİK VE DOĞA
BİLİMLERİ FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ
BÖLÜMÜ**

**BLM0471 Web Uygulamaları Tasarımı
(BAHAR 2021)**

HAYDE

Teknik Raporu

Takım Üyeleri

Adı Soyadı: Hatice Çelik	Görevi: Dokümantasyon
Adı Soyadı: Bayram Atakan Koşdurma	Görevi: Backend
Adı Soyadı: Yahya Can Özdemir	Görevi: Frontend
Adı Soyadı: Dilara Kanalcı	Görevi: Backend
Adı Soyadı: Erdoğan Turpcu	Görevi: Veri Tabanı

ÖNSÖZ

Bilindiği üzere PHP dilinin içerisinde kullanıcıların işlerini kolaylaştıran ve güvenlik açısından destek sağlayan birçok framework bulunur. "Web Sanatçılarının PHP Framework'ü" sloganıyla Laravel bu framework yapılarında ilk sıralarda gelir. Web uygulamaları geliştirilirken büyük bir kullanım oranına sahiptir. Laravel, yerleşik özellikler aracılığıyla web uygulamaları geliştirmeyi daha kolay ve daha hızlı hale getirmek için tasarlanmış açık kaynaklı bir PHP framework'tür.

Piyasada yaygın olarak kullanılan ve biz öğrenciler için çok önemli olan PHP Laravel ve MVC mimarisiyle HAYDE isimli sosyal medya uygulamasını hazırlama fırsatını bizlere sağladığı için Bursa Teknik Üniversitesi Bilgisayar Mühendisliği Öğretim Üyesi Sayın Dr. Volkan Altuntaş hocamıza teşekkür eder, saygılarımızı sunarız.

Grup 1
(Hatice Çelik
Yahya Can Özdemir
Dilara Kanalıcı
Erdoğan Turpçu)

Bursa 2021

İÇİNDEKİLER

ÖNSÖZ.....	2
İÇİNDEKİLER	3
ŞEKİLLER	4
ÖZET	5
1.GİRİŞ	5
1.1 HAYDE Projesi	5
1.2 Uygulamanın Hedef Kullanıcı Kitlesi	5
1.3 Projenin Geliştirici Takımı	6
2. GELİŞTİRİLME ORTAMI ve TEKNOLOJİSİ	6
2.1 Sistem Mimarisi.....	6
2.2 Yazılımın Tasarım Süreci	7
3.FRONTEND	8
3.1 Tasarım Desenleri.....	8
3.2 Html ve Css Nedir? Neden Kullanılır?	9
3.3 Javascript ve Bootstrap’ın Kullanımı.....	10
3.4 Web Uygulamaları Bileşenleri(UI/UX).....	10
4.DATABASE.....	17
4.1 Nosql and Mysql Farkları Nedir? Neden Mysql?	17
5.BACKEND.....	18
5.1 OOP Nedir?	18
5.2 Laravel MVC Modelleme	18
5.2.1 Models	19
5.2.2 Views	19
5.2 Controller(Denetleyiciler).....	20
5.3 Provider	20
5.4 Router	20
6.SONUÇ.....	21
7.KAYNAKLAR	21

ŞEKİLLER

1- Şekil 2.2 Yazılım Tasarlama Süreci	7
2- Şekil 3.4.1.a Kaydol Ekranı	10
3-Şekil 3.2.1.b Giriş Yap Ekranı	11
4- Şekil 3.4.2 Anasayfa Ekranı	12
5- Şekil 3.4.3 Arkadaşlık İstekleri	13
6- Şekil 3.4.4 Bildirim Ekranı.....	14
7- Şekil 3.4.5.a Profil Seçenekleri	15
8- Şekil 3.4.5.b Hesap Ayarları	15
9- Şekil 3.4.5.c Güvenlik Ayarları	16
10-Şekil 3.4.5 Kullanıcı Profil Ekranı.....	18
11- Şekil 5.2 MVC Yapısı.....	19

ÖZET

Sosyal medya; internet teknolojilerini kullanarak iletişim kurmayı, bilgi paylaşmayı ve içerik üretmeyi sağlayan bir platformdur. Sosyal Medya, bireylerin internette yazdıkları yazılardan, videolardan fotoğraflardan ve diğer materyallerden oluşur. Sosyal medya araçları kullanıcıları sosyal medyayı iletişim ve gündemi takip etme, kişilere ulaşmak, kişisel sunum ve bilgi paylaşmak ve insanları daha iyi tanımak için kullanmaktadırlar. Oluşturmuş olduğumuz “HAYDE” adlı sosyal medya uygulaması sayesinde Bursa Teknik Üniversitesi öğrencileri ve personel kadrosuyla gerek etkinlikler gerekse kişilerin kendi yaptıkları paylaşımlarla sürekli etkileşim halinde olmayı hedefleyen bir uygulama olmasıdır. Uygulamanın içeriğinden ve yazılım dilinden raporda detaylıca anlatılmış olup frontend alanında kullanıcıların kullanacağı ekran tasarımını backend alanındaysa yazılımcıların ihtiyacı olan materyaller içeren alan bulunmaktadır. Veri tabanı kısmına ayrı bir yer verilmiş olsa da backend kısmında laravelin kendi içeriğinde bulunmuş olan bir veritabanı alanı da bulunmaktadır.

Oluşturmuş olduğumuz uygulama sayesinde uygulamayı kullanmaya başlayan kişiler isterlerse içerik üretebilecek, istenirse etkinlikler düzenleyip bu etkinliklere katılma fırsatı yakalamış olacaklardır.

1.GİRİŞ

1.1 HAYDE Projesi

Projede ortaya çıkacak “Hayde” platformunun ismi; bu uygulamayı geliştirecek takım üyelerinin isimlerinin baş harflerinin birleşiminden oluşmaktadır. Aynı zamanda kelime anlamı olarak bir işte hızlanmayı sağlamak, işe isteklendirmek gibi anlamlar içeren bu kelimeyle kullanıcıların her anlamda harekete ve sosyal yaşama teşvik edilmesi amaçlanmaktadır. Geliştirilen platform; arkadaşlık kurma, konum ve fotoğraf paylaşma, etkinlik oluşturma, etkinliklere katılma ve bunlar gibi birçok özelliklerine sahiptir.

1.2 Uygulamanın Hedef Kullanıcı Kitlesi

Uygulamayı Bursa Teknik Üniversitesi öğrencileri ve üniversitenin akademik kadrosu kullanacaktır. Öğrenciler ortak dersleri, projeleri, topluluk etkinliklerini ve bölümleri hakkında iletişim kurabileceklerdir. Bunlarla birlikte projede birbirlerine akademik ve sosyal olarak (yeni gelen öğrencilere şehrin tanıtımı vb.) destek olmaları hedeflenmektedir.

1.3 Projenin Geliştirici Takımı



Bayram Atakan Koşdurma
Backend



Dilara Kanalıcı
Backend



Erdoğan Turpcu
Veri Tabanı



Yahya Can Özdemir
Frontend



Hatice Çelik
Dokümantasyon

2. GELİŞTİRİLME ORTAMI ve TEKNOLOJİSİ

2.1 Sistem Mimarisi

Sistem mimarisi terimi, bir bilgisayar ağının veya sistemin genel tasarımını ve yapısını tanımlamak için kullanılır. Bilgi teknolojisi çok çeşitli fiziksel cihazları içerecek şekilde genişlediğinden, bu maddeleri uyumlu bir şekilde bir araya getirmek ve bağlamak için bir yöntem gerekir.

Herhangi bir sistem mimarisinin dört ana bileşeni vardır: işlem gücü, depolama, bağlantı ve kullanıcı deneyimi. Sistemin karmaşıklığı çok değişkendir ve kullanıcı ihtiyaçlarına, iş gereksinimlerine, fonlamaya ve kaynak kullanılabilirliğine bağlıdır. Sistem mimarisinin esnek olması ve değişen ihtiyaçları hızla karşılayabilmesi gerekmektedir. Çok katı bir yapı, yeni yazılım veya donanımı barındıramaz.

İşlem gücü, bilgisayar veya sunucuya dayanmaktadır. Bu donanım sistemin beynine benzer. İşlemcilerin sistemi doğru kurmak, yazılımın özelliklerine, eşzamanlı kullanıcı sayısına, bağlantının gücüne ve uygulamalara dayanmalıdır. Bir sistem tasarlarken, ölçeklenebilirlik çok önemlidir.

Depolama alanı, sistemde yerleşik olan sabit sürücülerin ve ilgili cihazların sayısına ve kapasitesine bağlıdır. Maliyet bu tür ekipman için belirleyici bir faktördür, çünkü

kapasite arttıkça maliyet sürekli olarak düşer. Bu, üretim sürecinde devam eden gelişmelerden kaynaklanmaktadır. Ancak, mimari açıdan bakıldığında, bu sürece başka bir öge ekler. Kapasite arttıkça, genel fiziksel şekil de değişebilir ve bu da ekipmanı eski kılar.

Ağ trafiğini ve bağlantısını yönetme, sistem tasarımının önemli bir parçasıdır. Günlük yaşamdaki yollar gibi, sistemin performansı da sistemin tüm yönleri arasındaki bağlantının doğru şekilde boyutlandırılmasına ve korunmasına bağlıdır. Ağ kablosunu, anahtarları, yönlendiricileri ve diğer ekipmanları yükseltmek pahalı ve zaman alıcıdır, ancak sistem performansı üzerinde büyük bir etkisi vardır. Kullanıcı deneyimi, sistem mimarisi ve performansın bir kombinasyonuna dayanmaktadır. Hedef kitlemiz Bursa Teknik Üniversitesi öğrencileri ve akademik kadrosunu kapsadığı için burada uygulamamızı deneyimleme fırsatı sunacak kişilerin fikirleri oldukça önemlidir. Kullanıcıların geri dönütlerine göre uygulama içerisinde geçirdikleri sürelerle de baz alınarak gerekli iyileştirmeler yapılmaktadır. Çünkü iyi tasarlanmış bir destek sistemi, kullanıcıların ihtiyaçlarına cevap verir ve uzun vadede işlemi destekleyebilir.

2.2 Yazılımın Tasarım Süreci



Şekil 2.2 Yazılım Tasarlama Süreci

Bir yazılım geliştirme işlemi yalnızca kodlamadan oluşmamaktadır. Kodlama aşamasına gelene kadar geçen süreçler, projenin gerekliliklerinin öğrenilmesinden başlayıp kullanıcının kullanımı ve bakımına kadar birçok aşamayı barındırır. Yazılım talepleri ile alakalı gelişmeler sürekli olarak değiştiği ve genişlediğinden, söz konusu evreler devamlı bir döngü biçiminde ele alınır. Döngü içinde her hangi bir evrede geriye dönmek ve bir daha ilerlemek mevzu bahistir. Bu döngünün de temel bazı adımları vardır. Bunlar; Talep Alma, Planlama, Analiz, Tasarım, Kodlama, Test ve Bakım olarak sıralanırlar.

Projenin gerekliliklerini şöyle sıralayabiliriz; bir sosyal medya uygulaması olması, içeriğinde kişilerin kendi profilleri, ana sayfaları ve arkadaşlık istekleri, etkinlik oluşturma ve oluşturulmuş etkinliklere katılma, konum ve fotoğraf paylaşma ve bunlar gibi özelliklerdir.

Planlama ve proje analizi kısmında öncelikle yazılım projesinin ilk etabında projenin ihtiyaç duyduğu ana modüller analiz edilmeli, (veritabanı, UI/UX tasarımı, gerekli backend yazılımları vb.) proje amaçları, hedefleri ve yapılacak işler detaylandırılmalıdır. Proje varsayımları göz önüne alınmalı kullanıcıya yönelik faydaları gözden geçirilmelidir.

Projede kullanılacak en doğru yazılım dili, yazılım mimarisi, sunucu gereksinimleri belirlenmelidir.

Tasarım ve kodlama kısmında oluşturulmak istenilen proje web tabanlı olacağı için görsel arayüzde mobil, tablet ve masaüstüne opsiyonel olarak ayarlanabilmelidir. Bu doğrultuda oluşturduğumuz proje tasarımının bu platformlara veya cihazlara uygun olması gerektiği için tasarım aşamasında dikkat ettiğimiz kısımlar buralardı.

Güçlü bir yazılım mimarisi ile çalışılmalı ve sonradan çıkabilecek tüm isteklere kolaylıkla cevap verebilecek şekilde kodlama yapılabilmesi için artırimsal yazılım geliştirme modelini uyguladık.

Artırimsal yazılım geliştirme modelinde, yazılımın küçük parçalara ayrılarak döngüsel olarak geliştirilmesi amaçlanmaktadır. Proje süresi, döngü olarak tanımlanan küçük zaman dilimlerine bölünür. Proje birçok döngüden oluşur. Her döngünün sonunda, projeye ait planlanmış çıktılar elde edilir ve yazılıma yeni fonksiyonlar eklenir. Bu sayede yazılım artırimsal olarak geliştirilmesi planlanır ve biz de bu bağlamda yazılımımızı gerçekleştirdik.

Geliştirdiğimiz yazılımımızda gerek backend gerekse frontend alanında gerekli kullanıcı testlerini gerçekleştirerek kullanıma hazır duruma getirmiş bulunmaktayız.

3. FRONTEND

3.1 Tasarım Desenleri

Tasarım desenleri, yazılım tasarımında sürekli karşılaşılan genel sorunlara esnek, yeniden kullanılabilir, başarılı çözümler getiren bir takım hazır kalıplardır. Çeşitli durumlarda sorunların nasıl giderileceğini açıklayan, bunlara yol gösteren açıklamalardır. Nesneye dayalı programlamada, tasarım desenleri sınıf ve nesneler arasındaki ilişkilerin en iyi şekilde nasıl olmaları gerektiğini açıklayan yöntemlerdir.

Yazılım parçalarının tekrar kolayca kullanılabilir olmaları, kolayca genişleyebilir veya sistemden kolayca çıkarılabilir olmaları yani kısaca esnek olmaları beklenir. Yeni ihtiyaçların, yazılımın diğer kısımlarını en az biçimde etkileyerek yazılıma kolayca dahil olmaları beklenen kalıplardır. Tasarım desenleri, daha önce sınanmış, ispatlanmış ve yararı görülmüş yazılım yöntemleri önerdiği için geliştirme sürecini hızlandırmasına olanak sağlar.

Tasarım desenlerinin belirlenmesi ve uygulanması aşamasında arayüz tasarımı planlanırken belli başlı kalıplarla çalışmak arayüzün tasarım kısmı için harcanan zamandan tasarruf etmemize olanak sağlar. Arayüzün işlevselliğini arttırmak için birçok desen kullanılır bunlardan bazıları, formlar, kullanıcı giriş ekranı işlemleri, mesaj kutusu ve bazı kart tasarımları olarak söylenebilir. Şimdi detaylıca bunları inceleyelim.

3.2 Html ve Css Nedir? Neden Kullanılır?

Web sitelerinin vazgeçilmezi olan HTML (Hyper Text Markup Language), bir programlama dili olarak tanımlanamaz. Zira HTML kodlarıyla kendi başına çalışan bir program yazılamaz. Ancak bu dili yorumlayabilen programlar aracılığıyla çalışabilen programlar yazılabilir. Programlama dili denilememesinin nedeni tam olarak buna dayanır. HTML dosyalarının aktarımı için HTTP (Hyper Text Transfer Protocol) kullanılır. Temel gereği yazı, görüntü, video gibi değişik verileri ve bunları içeren sayfaları birbirine basitçe bağlamak, buna ek olarak söz konusu sayfaların web tarayıcısı yazılımları tarafından düzgün olarak görüntülenmesi için gerekli kuralları belirlemektir. HTML kodunu web tarayıcıları okur, yorumlar ve görsel hale dönüştürürler, dolayısıyla aynı HTML kodunun farklı tarayıcılarda farklı sonuç vermesi olasıdır. Bu farklılıkları, tasarımı, düzeni vb. sağlayan ve HTML yanında mutlaka CSS kullanılması gerekmektedir.

Cascading Style Sheets (Basamaklı Stil Şablonları ya da Basamaklı Biçim Sayfaları, bilinen kısa adıyla CSS), HTML'e ek olarak metin ve format biçimlendirme alanında fazladan olanaklar sunan bir Web teknolojisidir. Bir web sitesi içerisinde birbiriyle uyumlu renk ve yazı tipi gibi öğelerin her sayfa için ayrı ayrı değil de tek bir yerden yönetilmesi anlamına gelen bir çeşit kodlama türüdür.

Proje içinde html ve css şu şekilde kodlanmıştır.

```
<!-- \\\\\\\\\\\Create Post-->
<div id="createpostarea" class="card gedf-card rounded ">
  <p id="create-post-card-title"> Gönderi Oluştur</p>
  <div class="card-header">
```

Burada html içinde var olan classların içindeki yapılar css klasöründe belirtilmiş olup şu şekilde kodlanmıştır;

```
.gedf-card {  
  background-color: #263258;  
  animation: 1s fade;  
  -webkit-box-shadow: 0 0 20px rgba(0, 0, 0,  
1.8);  
  -moz-box-shadow: 0 0 20px rgba(0, 0, 0,  
1.8);  
  box-shadow: 0 0 20px rgba(0, 0, 0, 1.8);  
  margin-bottom: 2.77rem;  
}
```

3.3 Javascript ve Bootstrap Kullanımı

Bootstrap, HTML, CSS ve JavaScript ile yazılmış kullanışlı, yeniden kullanılabilir kod parçalarından oluşan dev bir koleksiyondur. Ayrıca, geliştiricilerin ve tasarımcıların hızla tam olarak duyarlı web siteleri oluşturmasını sağlayan bir frameworktür. Bootstrap web uygulaması geliştirme araç takımı, eski Twitter çalışanları Mark Otto ve Jacob Thornton tarafından oluşturulmuştur.

Bootstrap'ın projede kullanılmasının sebeplerini şöyle sıralayabiliriz.

- 1- Responsive Görünüm sağlar. Responsive gösterimi, bir web sitesinin tüm cihazlarda iyi görünmesini sağlayan web sitesi tasarımına yaklaşmanın nispeten yeni bir yoludur. Duyarlı web sitelerinde, hangi cihazda olursanız olun aynı bilgi ve sayfa öğeleri görünür, ancak boyutlandırma ve düzenleme yöntemleri ekran boyutunuza göre değişir.
- 2- Bootstrap Faydalı Komponentler İçerir. Bu komponentler; navbarlar, dropdown yapıları, Thumbnails... vb yapıları sağlar.
- 3- Bootstrap İhtiyaca Göre Özelleştirilebilir. Frameworkler karşılaştırılırken en önemli parametrelerden birisi performanstır. Eğer framework uygulama üzerinde ağırlık yaratıyor ise uygulamanın yavaş açılmasına neden olabilir örn; Bootstrap'ın CSS dosyasının boyutu 119 KB'dır. Bu bir CSS dosyası için büyük bir boyuttur. Bootstrap bu sorunla başa çıkabilmek için frameworkü özelleştirebilmenize imkan sağlar.

3.4 Web Uygulamaları Bileşenleri(UI/UX)

Arayüz olarak web sitelerinin kullanıldığı uygulamalar web uygulaması olarak adlandırılır. Daha açık bir ifadeyle dinamik web sayfalarının ve genellikle bir veritabanının kullanıldığı sistemlere web uygulamaları denir

3.4.1 Kullanıcı Giriş Ekranı

Projemizde bulunan kullanıcı giriş ekranı 2 ana bölümden oluşmaktadır. Birinci bölüm kayıt olma bölümüdür. Bu bölümde kullanıcılar ad, soyad, e-posta adreslerini ve şifrelerini girerek kayıt olurlar. Kullanıcılar kaydolduktan sonra yazdıkları e-posta adreslerine birer doğrulama maili gönderilir böylece oturum açma iznine sahip olurlar.

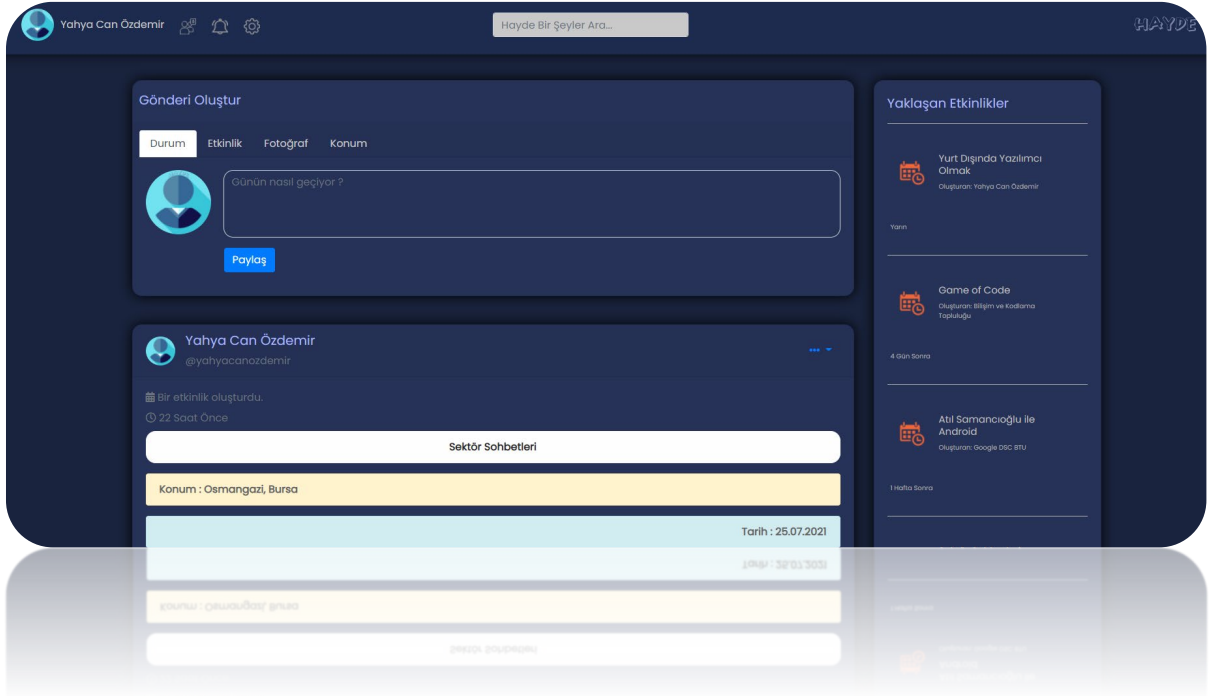
Şekil 3.4.1.a Kaydol Ekranı

İkinci kısımdaysa kayıtlı kullanıcılar için giriş yapma ekranı yer alır. Kullanıcılar e-posta ve şifrelerini girerek uygulamamızı kullanmaya başlayabilirler.

Şekil 3.2.1.b Giriş Yap Ekranı

3.4.2 Anasayfa

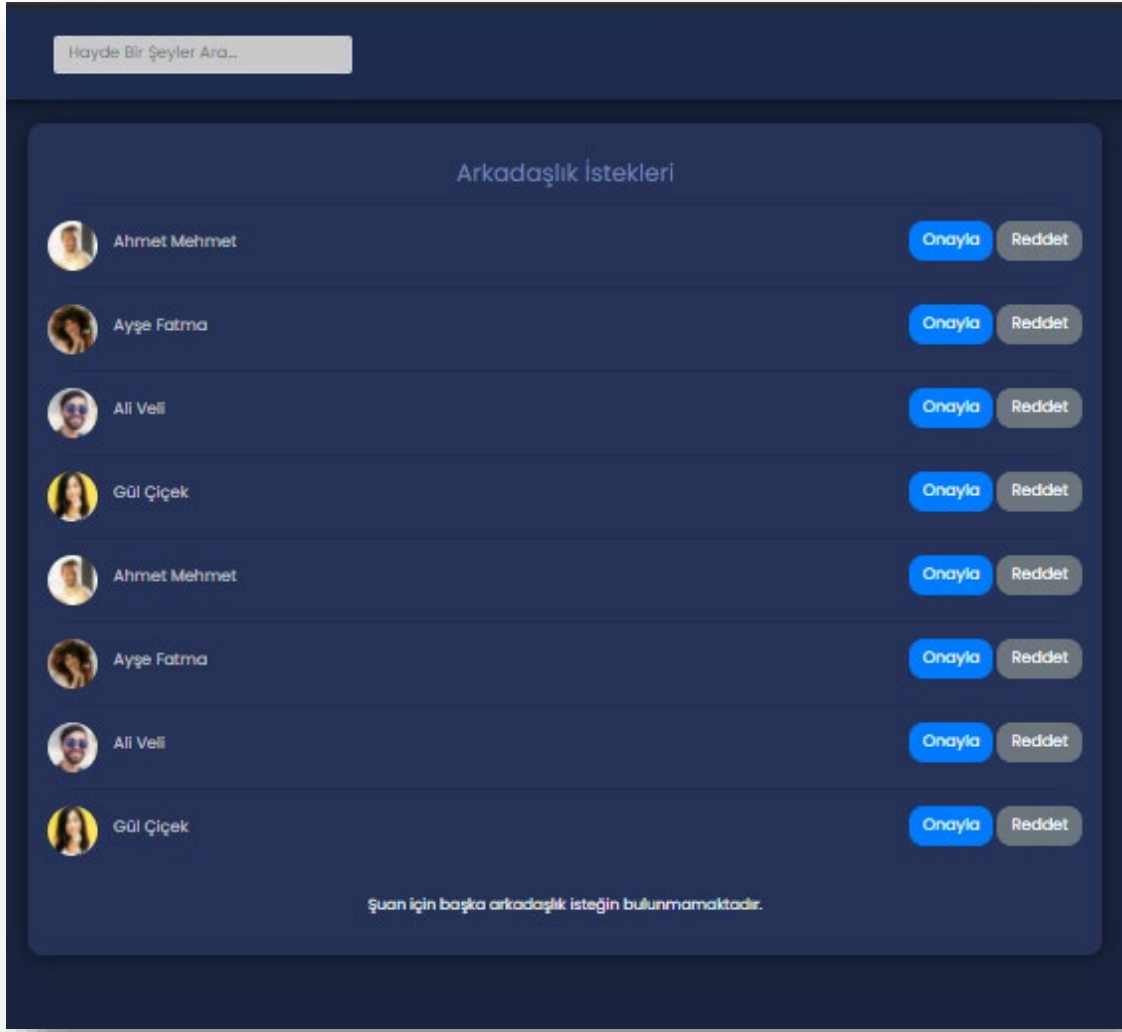
Kullanıcıların giriş yaptıktan sonra açılan pencerede sağ üst köşede uygulamanın logosu bulunmaktadır böylece kullanıcılar hangi sayfaya girerlerse girsinler ana sayfaya dönmek isteyecekleri zaman bu logoya tıklamaları yeterli olacaktır. Sayfanın sol üst köşesindeyse kişinin profil resmi, ismi, arkadaşlık istekleri, bildirimler ve ayarlara erişebilmek için küçük ikonlar bulunmaktadır. Tam ortadaysa bir arama çubuğu yer almaktadır. Sayfanın alt kısımlarına indikçe en üstte gönderi oluşturmak için bir alan bulunmaktadır. Bu alanda kişiler durum, fotoğraf, konum ve etkinlik oluşturup paylaşım yapabilmektedir. Bu alanın sağ yanında uygulama içindeki oluşturulmuş etkinlikler tarihe göre sıralanmaktadır ve en yakın tarihli dört etkinlik, etkinlik adı ve tarihiyle birlikte yer almaktadır. Sayfanın alt kısımlarına inildikçe kullanıcının ve arkadaşlarının paylaşımları görülmektedir.



Şekil 3.4.2 Anasayfa Ekranı

3.4.3 Arkadaşlık İstekleri

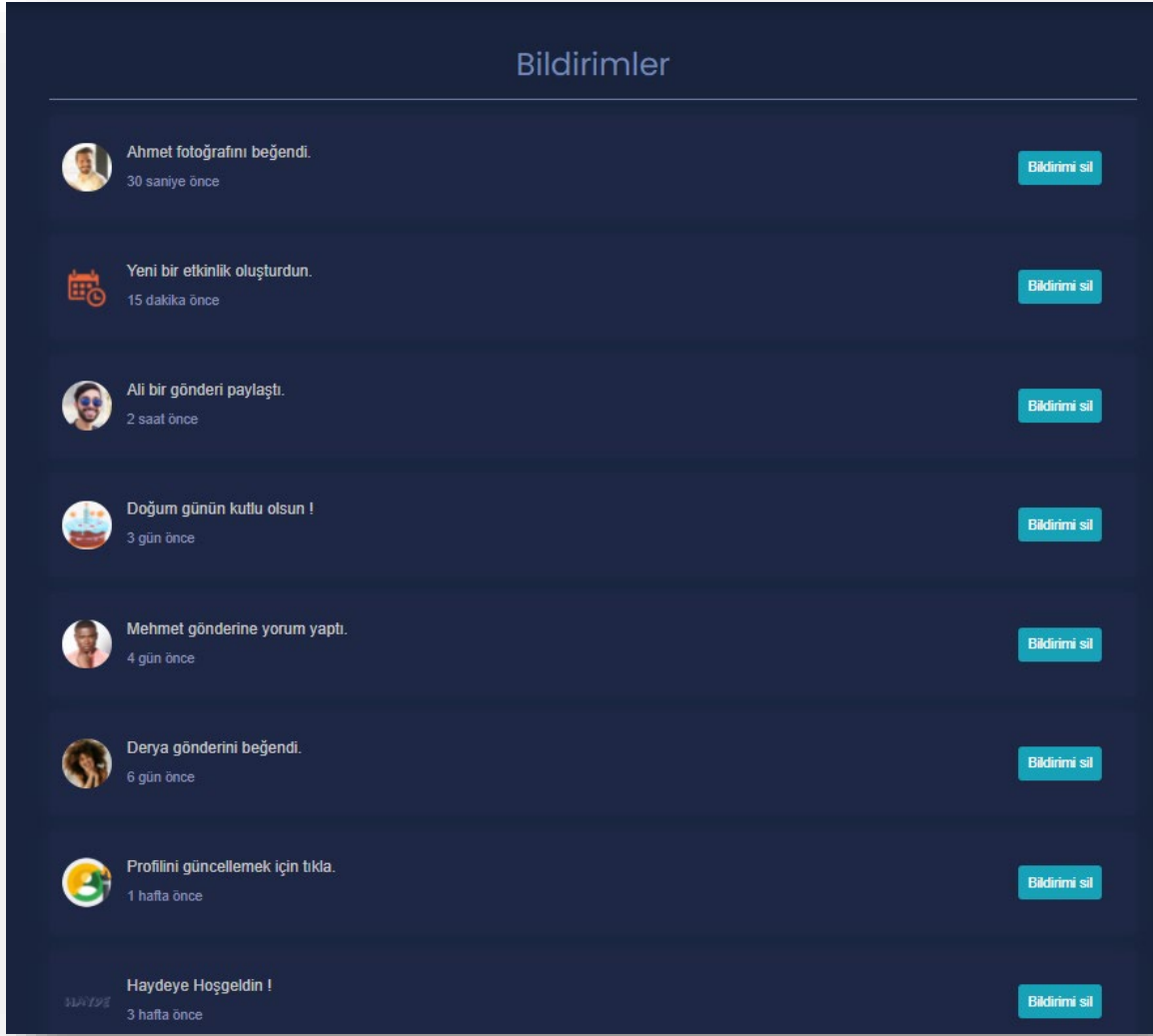
Üst kısımda bulunan arkadaşlık istekleri ikonuna tıklanıldığında kullanıcı aşağıdaki resimde görünen ekranla karşılaşmaktadır. Bu ekran sayesinde kimlerin arkadaşlık isteğini görebilir dilerse onaylar veya reddedebilir.



Şekil 3.4.3 Arkadaşlık İstekleri

3.4.4 Bildirimler

Ana sayfada sol üst kısımda bulunan bildirimler düğmesine tıklanıldığında kullanıcıyı aşağıdaki bildirimler ekranı karşılamaktadır. Açılan ekranda kullanıcı istediğinde bildirimi silebilmesi için buton eklenmiştir.



Şekil 3.4.4 Bildirim Ekranı

3.4.5 Ayarlar


Ana sayfada sol üst kısımda bulunan bildirim ikonun yan tarafında ayarlar kısmı yer almaktadır bu ikona tıklanıldığında kullanıcının profil ayarları, hesap ayarları ve güvenlik ayarları olmak üzere üç ana başlık altına alınan bir panel ekranı bulunmaktadır. Profil ayarları kısmında kullanıcının profil resmi, adı, soyadı, yaşı, uyruğu, yaşadığı ili ve ilçesi, bölümü ve bölümünde kaçınıcı sınıfta olduğu, ilgi alanı ve diğer sosyal medya uygulamalarındaki kullanıcı ismi yer almaktadır. Hesap ayarları kısmındaysa şifre, kullanıcı adı, e-posta ve telefon için değişiklik yapma alanları bulunmaktadır. Üçüncü kısımdaysa kişilerin güvenlikleri için gerek oturum açma gerekse şifre değişikliklerinde mail gönderimini sağlamak amacıyla konulmuş butonlar bulunmaktadır. Bu ekranın en alt kısmında çıkış yap düğmesi yer almaktadır.

Ayarlar

Profil Seçenekleri

Hesap Ayarları

Güvenlik Ayarları



yahyacanozdemir

İsim

Yahya Can

Soyisim

Ozdemir

Yaş

21

Uyruk

Türkiye

Şehir

İstanbul

İlçe

Maltepe

Bölüm

Bilgisayar Mühendisliği

Sınıf

3

İlgili Alanı

Mobile Development

Facebook

@yahyacanozdemir

Twitter

@yahyacanozdemir

LinkedIn

@yahyacanozdemir

Instagram

@yahyacanozdemir

Kaydet

İptal

Çıkış Yap

HAYDE 2021 Tüm Hakları Saklıdır


Şekil 3.4.5.a Profil Seçenekleri

Ayarlar

Profil Seçenekleri

Hesap Ayarları

Güvenlik Ayarları



Mevcut Şifre

Yeni Şifre

Yeni Şifreyi Doğrula

Kullanıcı Adı

yahyacanozdemir

Birincil E-Posta

18360859048@ogrenci.btu.edu.tr

İkinci E-Posta

yahyacanozdemir@gmail.com

Telefon Numarası

+90532532532

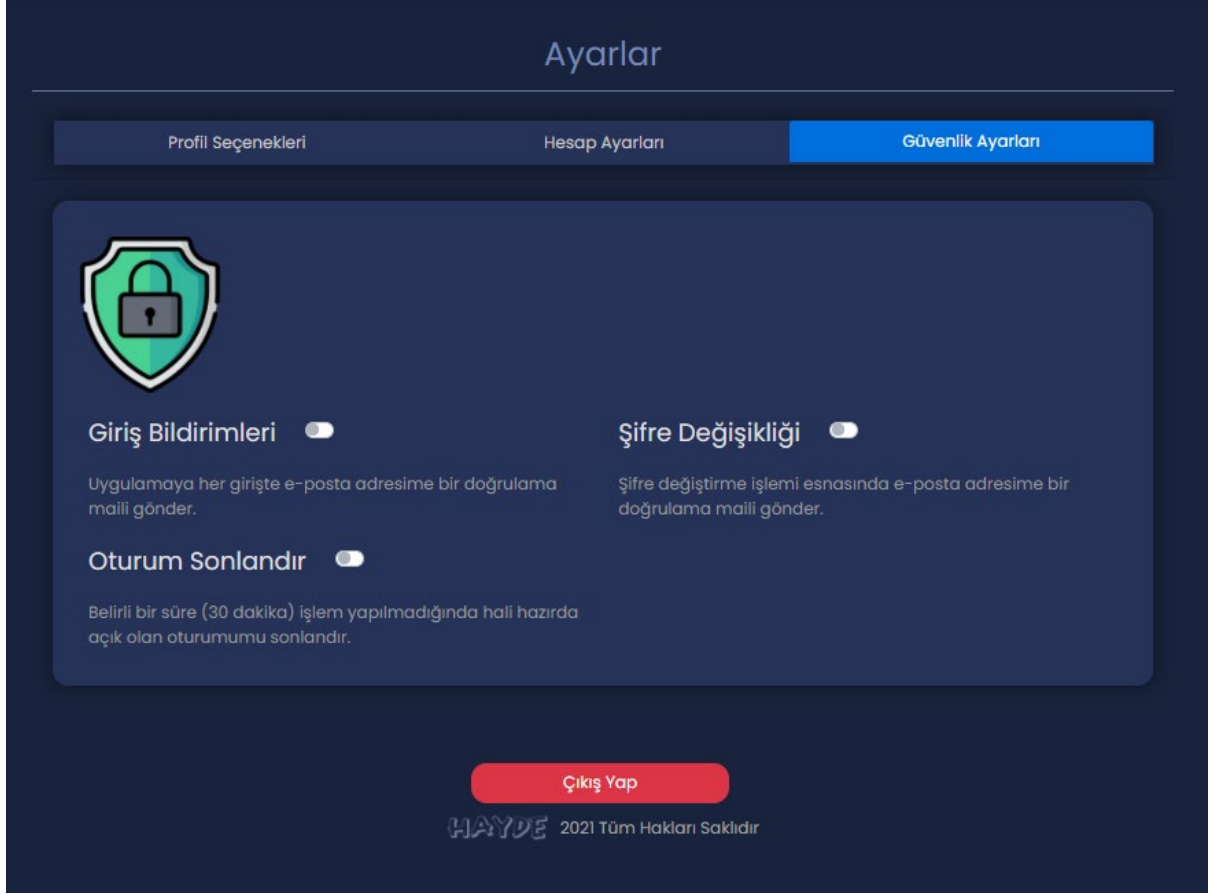
Kaydet

İptal

Çıkış Yap

HAYDE 2021 Tüm Hakları Saklıdır

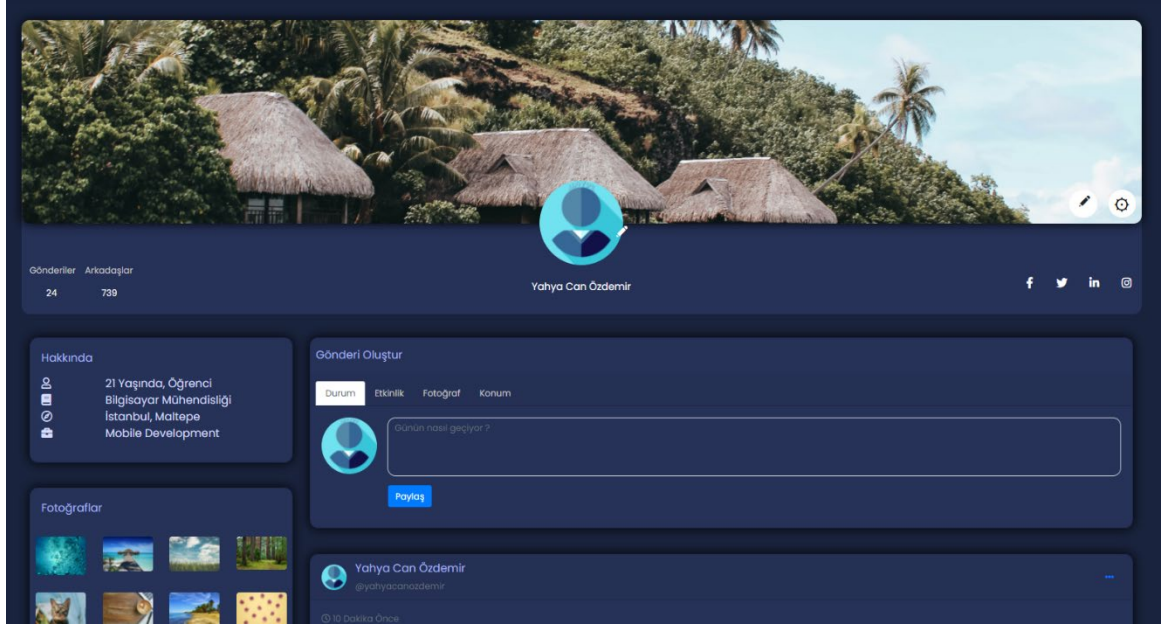
Şekil 3.4.5.b Hesap Ayarları



Şekil 3.4.5.c Güvenlik Ayarları

3.4.6 Kullanıcı Profil Ekranı

Kullanıcıların profil ekranında üst kısımda diğer sosyal medya araçlarında olduğu gibi bir profil fotoğrafı ve bir adet arka plan fotoğrafı bulunmaktadır. Fotoğraf alanlarının sağ alt taraflarında fotoğrafı değiştirmek için ayarlar bulunmaktadır.



Şekil 3.4.6.a Güvenlik Ayarları

4.DATABASE

4.1 Nosql and Mysql Farkları Nedir? Neden Mysql?

Veritabanı sistemlerinde amaç, kod satırlarına yerleştirilemeyecek büyüklükte ve değişken değerlere sahip olan verilerin, uygun koşullarda saklanarak, dilediği zaman istenen veriye erişimin sağlanmasıdır. Sunucuya yüklenen bir yazılım olan MySQL'e kullanıcılar, hazırladıkları dinamik kod sayfaları aracılığıyla sorgular gönderir. Böylece veritabanı ile etkileşime geçmiş olurlar.

1. Kolay Sorgulama Altyapısı: MySQL'in geniş komut dizinleri, kullanıcıların onlarca farklı sorgu yapmasına olanak sağlıyor. Ayrıca, yazılımın sahip olduğu komut dizinlerinin açıklamaları, web sayfaları üzerinde detaylı bir biçimde anlatılıyor, pek çok örnekle de destekleniyor.

2. Güçlü Karakter Seti Desteği: Dünya üzerinde yer alan tüm dillerin ve alfabelerin depolanabileceği şekilde tasarlanan MySQL, alfabeler arasında yapılacak geçişlerde de büyük kolaylık sağlıyor. Örnekleyecek olursak, kod sayfaları yazılırken, Türkçe karakterlerin internet sayfalarında görünmesi için UTF-8 ya da 1259-9 karakter setleri kullanılıyor. Bu karakterleri setleri ve daha binlercesi MySQL tarafından destekleniyor.

3. Sorgu Önbellekleme: Kullanıcıların daha önce yaptıkları aramaları önbelleğe alarak, daha sonra yapacağı aramalar için daha hızlı ve sunucunun işlemcisini daha az kullanarak sayfaya gönderiyor.

4. phpmyAdmin Arayüzü: Web hosting firmaları tarafından sunuculara kurulan MySQL yazılımı, kullanıcılara gösterilecek olan front-end yazılımıdır. phpmyAdmin sayesinde veritebanınıza erişerek bilgileri görebilir, değiştirebilir ve ayarları düzenleyebilirsiniz.

NoSQL veritabanı ilişkisel değildir. NoSQL DB'de, sorgulara odaklanmak ve belge toplamadır. Bu bazen Yapılandırılmamış Sorgu Dili (UnQL) olarak adlandırılır. UnQL sözdizimi bir veritabanından diğerine değişir.

SQL DB'ler, karmaşık sorgu ve yoğun ortamlar için daha uygundur. Bunun nedeni, NoSQL'in yüksek düzeyde karmaşık sorguları gerçekleştirmek için standart bir arayüze sahip olmaması ve NoSQL üzerindeki sorguların SQL sorguları kadar güçlü olmamasıdır.

Ortalama işlemler için NoSQL kullanmak mümkün olsa da, SQL, yoğun işlem tipi uygulamalar için en iyi seçimdir. Bunun nedeni, NoSQL'in karmaşık işlem uygulamaları ve yüksek trafik ile yüklendiğinde yeterince kararlı olmamasıdır. SQL, veritabanı kararlı olduğu ve verilerin bütünlüğünü ve atomikliğini vaat ettiği için iyi bir uyum sağlar.

5.BACKEND

5.1 OOP Nedir?

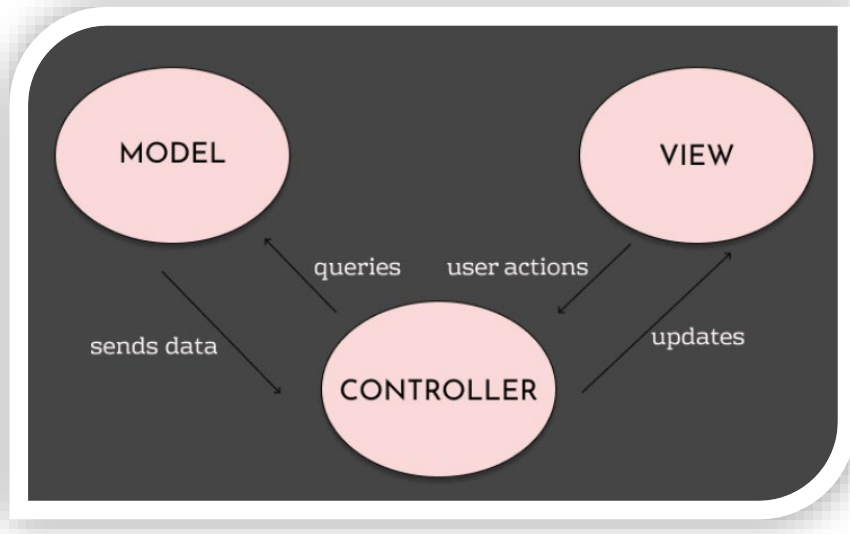
OOP, bir tasarım modelidir. OOP kavramı şu ihtiyaçlardan dolayı doğmuştur: Eğer bir kod yığını sürekli olarak kullanılıyorsa, aynı işlemi yapan kodu her projede tek tek yazmak veya kopyalayıp yapıştırarak modifiye etmek yerine OOP mantığı ile bir Class oluşturup, bu Class'ta sadece istediğiniz işi yapacak olan fonksiyonlara (Class içindeki fonksiyonlar metot olarak adlandırılır) gerekli parametreler gönderilir. Örneğin; projemizde uyguladığımız doğrulama e-maili göndermek için bir e-mail gönderme sınıfı oluşturup ve bu Class'a sadece alıcı adresi, mesaj içeriği, başlık gibi bilgiler gönderilmektedir. Oluşturulan bir Class belirli tasarım standartlarına göre hazırlanır. Bu Class'lar bir araya gelir ve daha önce işlemiş olduğumuz Framework'leri oluşturur. Yani Framework'ler birer Class yığınıdır. OOP sisteminin amacı daha az zamanda, daha verimli bir şekilde ve daha kısa kod yazabilmektir. OOP'nin diğer bir faydası da kompleks, yani karmaşık projelerde çalışırken sağlamış olduğu kodlama hakimiyetidir.

5.2 Laravel MVC Modelleme

MVC, 'Model View Controller' için bir kısaltmadır. Geliştiricilerin uygulama oluştururken benimsediği mimariyi temsil eder. MVC mimarisi ile uygulamamızın veri akışının nasıl çalıştığı ile ilgili olarak uygulama yapısına bakıyoruz.

MVC, etki alanını/uygulamayı/iş mantığını kullanıcı arabiriminin geri kalanından ayıran bir yazılım mimarisidir. Bunu uygulamayı üç parçaya ayırarak yapar: model, görünüm ve denetleyici.

Model, kod tabanımızdaki gerçek hayattaki bir örneğin veya nesnenin temsilidir. View, kullanıcının uygulamamızla etkileşime girdiği arayüzü temsil eder. Bir kullanıcı bir işlem yaptığında, Controller eylemi gerçekleştirir ve gerekirse Modeli günceller.



Şekil 5.2 MVC Yapısı

5.2.1 Models

Model, uygulamanın temel davranışlarını ve verilerini yönetir. Bilgi taleplerine cevap verebilir, bilgilerinin durumunu değiştirmek için talimatlara cevap verebilir ve hatta bilgi değiştiğinde olay güdümlü sistemlerde gözlemcileri bilgilendirebilir. Bu bir veritabanı veya herhangi bir sayıda veri yapısı veya depolama sistemi olabilir. Kısaca uygulamanın veri ve veri yönetimidir.

Laravel'e dahil olan Eloquent ORM, veritabanınızla çalışmanızı sağlayan kullanışlı ve basit ActiveRecord uygulamasıdır. Her veritabanı tablosuna karşılık gelen "Model", tablolarla etkileşim için kullanılır. Modeller veritabanındaki tablolarımızdaki veriler için sorgular yapmayı, yeni kayıtlar girmeyi sağlar.

5.2.2 Views

View, görüntülenebilen bir şablondur ve controller tarafından değiştirilebilir. View, uygulamanızın tüm HTML bileşenlerinden meydana gelmektedir. Views, yeniden kullanılabilir HTML kodu parçaları ve genel mizanpaj dosyaları yazmamıza olanak tanıyan Blade şablon oluşturma motorunu desteklemektedir.

Blade, Laravel'de bulunan basit ama güçlü bir şablonlama motorudur. Bazı PHP şablonlama motorlarından farklı olarak Blade, şablonlarınızda düz PHP kodu kullanmanızı kısıtlamaz. Aslında, tüm Blade şablonları düz PHP kodunda derlenir ve değiştirilene kadar önbelleğe alınır, bu da Blade'in uygulamanıza temel olarak sıfır ek

yük eklediği anlamına gelir. Blade şablon dosyaları .blade.php dosya uzantısını kullanır ve kaynaklar/görünümler dizininde depolanır.

5.2.3 Controller(Denetleyiciler)

Denetleyiciler, ilgili istek işleme mantığının tek bir sınıfta gruplanmasına izin verir. Tüm istek işleme mantığını tek bir route.php dosyasında tanımlamak yerine, bu davranışı Controller sınıflarını kullanarak düzenlemek gerekir. Denetleyiciler, ilgili HTTP istek işleme mantığını bir sınıfta gruplanabilmektedir. Denetleyiciler app/Http/Controllers dizininde depolanır. Middleware, uygulamamıza giren HTTP isteklerini filtrelemek için uygun bir mekanizma sağlar. Örneğin Laravel, uygulamamızın kullanıcısının kimlik doğrulamasını doğrulayan bir ara katmanı içeriyor. Kullanıcı kimliği doğrulanmazsa, katman kullanıcıyı oturum açma ekranına yönlendirir. Bununla birlikte, kullanıcı kimliği doğrulanırsa, ara katman, isteğin uygulamanın ilerleyişine izin vermesine izin verir.

5.3 Provider

Providerlar, tüm Laravel uygulamasında önyüklemelerinin merkezi yeridir. Laravel'in tüm temel hizmetleri, hizmet sağlayıcılar aracılığıyla önyüklenmektedir.

Önyükleme'nin amacı hizmet kapsayıcı bağlamalarını, olay dinleyicilerini, ara katman yazılımlarını ve hatta rotaları kaydetme dahil olmak üzere şeyleri kaydetmektir. Servis sağlayıcılar, uygulamamızı yapılandırmak için merkezi yerdir.

Uygulama içerisindeki config/app.phplaravel dosyasını açtığımızda providerlar bulunmaktadır. Bunlar, uygulamanız için yüklenecek olan servis sağlayıcı sınıflarının tamamıdır. Varsayılan olarak, bu dizide bir dizi Laravel çekirdek hizmet sağlayıcı listelenir. Bu sağlayıcılar, posta, kuyruk, önbellek ve diğerleri gibi temel Laravel bileşenlerini önyükler. Bu sağlayıcıların çoğu "ertelenmiş" sağlayıcılardır, yani her istekte değil, yalnızca sağladıkları hizmetlere gerçekten ihtiyaç duyulduğunda yüklenirler.

5.3 Router

Router: Türkçe karşılığı rota olan ve istemcinin gönderdiği adreslerin gerekli Controller ve Action yönlendirilmesi sağlayan yapılardır. Kaynak rotası, bir rotalar dosyasında tanımlanabilen bir kısayol rotasıdır. Tüm Laravel rotaları, rotalar dizininde bulunan rota dosyalarınızda tanımlanır. Bu dosyalar çerçeve tarafından otomatik olarak yüklenir. route/web.php dosyası, web arayüzümüz için olan yolları tanımlamaktadır. Bu rotalara, oturum durumu ve CSRF koruması gibi özellikler sağlayan web ara katman yazılımı grubu atanır. Hayde uygulaması için, route/web.php dosyasındaki rotalar tanımlanır. Routes/web.php'de tanımlanan rotalara, tarayıcıya tanımlanan rotanın URL'sini girerek erişilebilir.

6.SONUÇ

Sonuç olarak laravel, php, bootstrap, html,css ve mysql kullanılarak oluşturulmuş olan HAYDE adlı sosyal medya uygulaması içerisinde durum,etkinlik, fotoğraf ve konum paylaşma özelliği bulunmaktadır. Aynı zamanda kullanıcılar birbirlerini arkadaş olarak ekleyebilir, şifrelerini unuttuklarını veya yeni kayıt olmak istediklerinde maillerine doğrulama kodu gönderimi yapılabilmektedir. Bu proje kapsamında gerek backend gerek frontend gerekse database kısmında bir çok yeni yapılar öğrenilmiş olup proje içinde uygulanmaya çalışılmıştır. Böylelikle üniversitemize yeni bir sosyal medya uygulaması kazandırılmıştır.

7.KAYNAKLAR

1. <https://github.com/laravel/laravel>
2. <https://laravel.com/docs/8.x/providers>
3. <https://laravel.com/docs/5.6/routing>
4. <https://medium.com/why-i-chose-to-switch-career-paths-and-pursue/intro-to-bootstrap-baa07f334233>
5. <http://ogretmenkoyu.baskabirokulmumkun.org/category/kutuphane/dokumantasyon/>
6. <https://www.netinbag.com/tr/internet/what-is-system-architecture.html>
7. <http://iibf.erciyes.edu.tr/guven/web/webuygulamalari.pdf>