

Umuttepe Turizm Uygulaması

*Umuttepe Turizim ile güvenli seyahat edin

İsmail Aydın
211207067

*Kocaeli Üniversitesi Bilişim Sistemleri
Mühendisliği*

Elif Özkan
211307057

Kocaeli Üniversitesi Bilişim Sistemleri
Mühendisliği

Tayfun Açııcı
211307022

*Kocaeli Üniversitesi Bilişim Sistemleri
Mühendisliği*

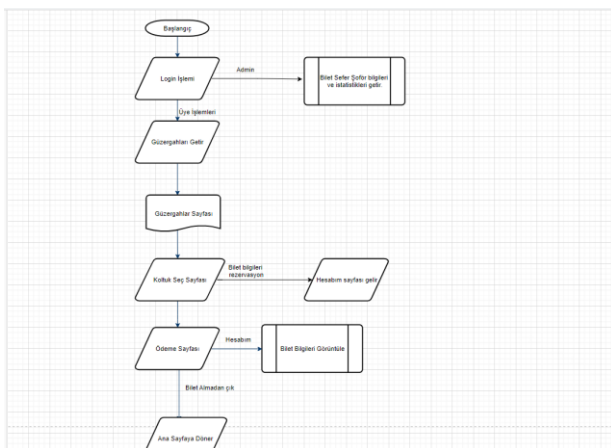
Abstract— With this project we wanted to create a web application that users can use to travel in a nice and safe way. The main purpose of our application is not only users to buy travel tickets, but also for the application to actively manage the travel process in an effective relationship with the users. For this we used CodeIgniter Framework based on PHP programming language, HTML, CSS Bootstrap

Özet— Bu proje ile kullanıcıların güzel ve güvenli şekilde seyahat edebilmeleri için kullanabilecekleri bir web uygulaması yapmak istedik uygulamamızın asıl amacı kullanıcı bireylerin sadece seyahat bileti alması değil aynı zamanda uygulamanın kullanıcı ile efektif bir ilişki içinde olarak aktif şekilde seyahat etme sürecini yönetmesini sağladık. Bunun için de PHP üzerine inşaa edilmiş CodeIgniter Frameworkü ile HTML, CSS ve Bootstrap kullandık.

Keywords — *seyehat, CodeIgniter, View, Controller, Bootstrap, MVC*

I. GİRİŞ

Seyahat etmek ve bunu ekonomik, güvenli ve organize şekilde yapmak temel olarak hepimizin sorunu. Hem bilet almaya vakit bulmak hem de bu işlemi uygun zamanda ve ekonomik olarak gerçekleştirmek için bir web uygulaması yapmak istedik. Bu uygulama ile sadece bilet almakla kalmayan üye aynı zamanda seçtiği konforlu yer ile konforlu bir yolculuk imkanı da elde ediyor.



Şekil-1

II. BACKEND

A. CodeIgniter

Bu kısım projemiz için büyük önem teşkil etmektedir çünkü projede backend mimarinin en önemli yapı taşıdır. PHP tabanlı bir Framework olan CodeIgniter, yapısı içinde MVC mimarisi üzerine inşa edilmiş bir Framework'tür. Projemizin backend yapısı bu Framework kullanılarak geliştirildi. Buradaki MVC mimarisinde, model yapısı veritabanı işlemlerinin yapıldığı karam, view yapısı kullanıcıya sunulan arayüz kısmı, controller yapısı ise model ve view yapıları arasında HTTP isteklerini alıp, bunlara uygun şekilde view yapısını çağırıp client'in tarayıcısına bu yanıtı döndürmektedir.

B. Veritabani

Projemiz, veri tabanı olarak MySQL teknolojisini kullanmaktadır. MySQL veritabanında kullanacağınız tabloları ise tasarlamak için Mysql Workbench yazılımını kullandık. Bu amaçla projemizin isterleri doğrultusunda; aktif bilet bilgileri, açığa alınan bilet bilgileri, yolcular, kart bilgileri, rezervasyonla, ödeme verilerini tutabilmek üzere tablolar oluşturulmuştur.

C. View

View yapısı sayesinde client'in tarayıcısına bir çıktı verebilmekteyiz. Bundan dolayı, kullanıcının ekranına gönderilecek tüm içerikleri bu klasörde tuttuk. Bir controller içinde bir view dosyasını yüklemek için fonksiyonları kullandık. Projemizin sayfalarını Html uzantılı dosyaları PHP uzantılı hale getirdiğimizde bu klasörde tuttuk.

[illegible]

Şekil-2

D. Controller

Kullanıcı isteklerini yönlendirme işini projemizde Controller yapısı ile sağladık. Arayüzümüz için hazırladığımız View yapıları ile sorguları yaptığımız Model yapısı arasında mantıksal ilişki kurmamızı sağladı. HTTP isteklerini yönlendirerek arayüz için hazırladığımız view yapılarını yönetmemizi sağladı.

```
class Home extends BaseController
{
    public function index()
    {
    }

    public function form()
    {
        return view('form');
    }

    public function güzergahlar()
    {
        return view('guzergahlar');
    }

    public function admin()
    {
        return view('admin');
    }

    public function seferkontrol()
    {
        return view('seferkontrol');
    }

    public function biletkontrol()
    {
        return view('biletkontrol');
    }

    public function kayıtlar()
    {
        return view('kayitlar');
    }

    public function login()
    {
        return view('login');
    }
}
```

Şekil-4

E. Routes

Routes yapısı sayesinde client uygulamada farklı sayfalara erişebilmesini sağladık. Burada, HTTP isteklerini yönetmek için \$routes nesnesini kullanarak client'ın tasarladığımız sayfalara ulaşabilmesini sağladık. Ayrıca burada sitemizin de haritasını çıkarmış olduk.

```

$routes->get('/', 'Home::index');
$routes->get('deneme', 'Home::deneme');
$routes->get('guzergahlar', 'Home::guzergahlar');
$routes->get('form', 'Home::form');
$routes->get('admin', 'Home::admin');
$routes->get('giris-yap', 'Home::login');
$routes->get('seferkontrol', 'Home::seferkontrol');
$routes->get('biletkontrol', 'Home::biletkontrol');
$routes->get('kayitlar', 'Home::kayitlar');
$routes->get('login', 'Home::login');
$routes->get('adminlogin', 'Home::adminlogin');
$routes->get('login/forgetpassword', 'Home::forgetpassword');
$routes->get('admin/allbus', 'Home::allbus');
$routes->get('admin/bustetail', 'Home::bustetail');
$routes->get('admin/soforliste', 'Home::soforliste');
$routes->get('guzergahlar/odeme', 'Home::odeme');
$routes->get('admin/adminseferler', 'Home::adminseferler');
$routes->get('koltuksec', 'Home::koltuksec');
$routes->get('hesabim', 'Home::hesabim');
$routes->post('giris', 'Home::giris');
$routes->get('odeme', 'Home::odeme');
```

Şekil-3

F. Ana Sayfa

Projemizin backend kısmını hayata geçirmeye gerçek manada başlamadan önce CodeIgniter kendi sitesinden composer yapısı ile birlikte indirdikten sonra C diski içinde bulunan htdocs klasörünün içine terminalden composer yardımıyla proje oluşturduk.

Ad	Uygulama tarihi	İş	Statüs
anonymos	16.02.2024 17:09	Deneme	Klasik
apache	16.02.2024 17:09	Deneme	Klasik
cp-bin	16.02.2024 17:10	Deneme	Klasik
cshtml	16.02.2024 17:10	Deneme	Klasik
FluFluFTP	16.02.2024 17:10	Deneme	Klasik
Index	23.02.2024 16:38	Deneme	Klasik

Şekil-4

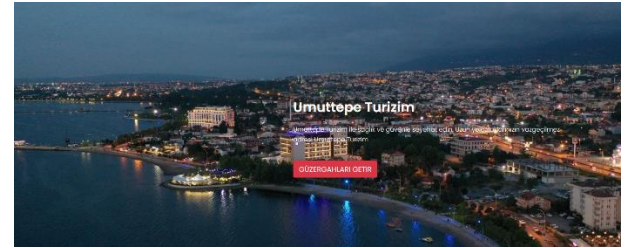
Ana sayfamızı dinamik hale getirmeden önce Projemiz için aldığımız template yapısı içinde bulunan sayfayı tasarımsal olarak değiştiriyoruz.. Öncelikle dönmesini istediğimiz view dosyasını html formattan PHP uzantılı formata dönüştürdük. Backend için yaptığımız bu düzenlemeden sonra uygun controller yapısı için fonksiyon yazdık ve routes.php routeri üzerinden url olarak dönmesini ve istediğimiz sayfaya yönlendirme yapabilmesi için uygun router yapısını yazdık.

```
namespace App\Controllers;

use activeLoginModel;

class Home extends BaseController
{
    public function index()
    {
        return view('deneme');
    }
}
```

Şekil-5



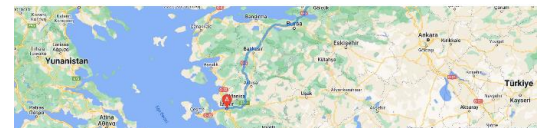
Şekil-6

```
7 | */
8 $routes->get('/', 'Home::index');
```

Şekil-7

G. Güzergahlar

Bu sayfada kullanıcının gidebileceği istikametleri Google Maps'in bize sağladığı API'yi kullanarak göstermek istedik. Bu sayfada kullanıcı seçini yaptıktan sonraki kısımda Google Maps üzerinden aradaki mesafeyi görebiliyor. Burada gerekli seçimi yaptıktan sonra bilet alabilmek için koltuk seçimi kısmına geçiyoruz. Aynı şekilde Controller view ve router mantığını kullanarak öncelikli olarak sayfa tasarım yapısını PHP yapısına dönüştürdük içine API yapısını entegre ettik. Ardından da sefer istikametlerini ekledik.



Kaynak Noktası:	Hedef Noktası:
<input type="text" value="İzmir otogar"/>	<input type="text" value="Kocaeli otogar"/>
<input type="button" value="Güzergahı Göster"/>	<input type="button" value="İstikamet Göster"/>

Şekil-8

Şekil-9

Aynı zamanda sayfa yüklendiğinde uygulama veri tabanından gidilebilecek güzergahları çekerek bunları liste halinde ekrana bastırıyor. Buradan sonra butona tıkladığımızda bizi sefer kontrol sayfasına yönlendiriyor. Burada tarihe göre sefer listesi içinde seçim yaptıktan sonra kullanıcıyı koltuk seçimi aşamasına geçiyoruz.

H. Sefer Kontrol Sayfası

Bu sayfamızda seçtiğimiz sefer için saat bilgileri geliyor. Ayrıca manuel olarak da seçebilme seçeneği sunduk. Bu sayfada aynı zamanda firmamızda düzenlediğimiz seferler ile ilgili mola yeri bilgisi de yer almaktadır. Bu sayfamızdan sonra ise kullanıcıyı sefer için koltuk seçimine yönlendirme aşamasına geçtik.

Şekil-10

İ. Koltuk Seçme Sayfası

Bu sayfada kullanıcımız otobüs seferindeki boş ve dolu koltukları görebilip buradan seçim yapabilmektedir. Kadın kullanıcılarımızın satın almış olduğu koltuklar pembe, erkek kullanıcılarımızın satın almış olduğu koltuklar ise mavi renkte belirtilmektedir.

Şekil-11

Kullanıcı, biletini satın almadan önce cinsiyetini ekranda çıkan bir “popup” ile seçerek bu şekilde koltukların rengini belirtebilmektedir.

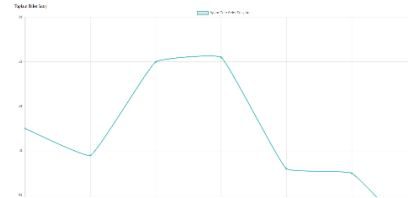
Şekil-12

ÖDEME SAYFASI

Bu sayfada kullanıcı üyemizden aldığı bilet için kart bilgilerini istiyoruz. Daha sonrasında ödemesini onaylayan üyemizi bilet kontrol sayfasına yönlendiriyoruz. Bu sayfada PNR kodunu giren üyemiz bileti için gerekli QR bilgisine ulaşabiliyor. Daha sonrasında ise ana sayfaya yönlenebiliyor

ADMIN SAYFASI

Bu sayfa, uygulamanın admin yetkisine sahip üyesinin sitede çeşitli yönetim işlemlerini yapabilmesine uygun olacak şekilde tasarlandı. Sistem ile alakalı genel bilgileri ve otobüs seferleri ile ilgili detayları görüntüleyebilir. Admin sayfası login durumuna geçerken geçerli kullanıcı adı ve şifre girilmediği takdirde erişim sağlanamayacaktır. Bunun yanı sıra aynı zamanda login paneli ve admin bilgilerini koruyabilmek için çeşitli önlemler aldık. Bunların başında karakterlerin htmlspecialchars methoduna dahil ederek kötü niyetli yazılımcıların sistemimize sızmasını önlemek. Bu sayfada aynı zamanda düzenlediğimiz seferler ile alakalı çeşitli bilgilere ulaşabiliyoruz. Bunları grafikler ile de görselleştirebiliyoruz.



Şekil-13

```

<div class="container">
  <script>
    document.addEventListener("DOMContentLoaded", function() {
      document.getElementById("giris-yap-butonu").addEventListener("click", function() {
        var username = document.getElementById("exampleFormControlInput1").value;
        var password = document.getElementById("exampleInputPassword1").value;
        if(username == "admin" && password == "umuttepe") {
          window.location.href = "<?php echo base_url('/admin')?>";
        } else {
          alert("Kullanıcı adı veya şifre yanlış!");
        }
      });
    });
  </script>
</div>

```

Şekil-14

KULLANICI SAYFASI VE GÜVENLİK ÖNLEMLERİMİZ

Kullanıcı sayfamızda üye olan kullanıcımız bilgilerini girdikten sonra hesabım sayfasına yönlendirilmesi

yapılmaktadır. Burada bilet kontrolü kısmından bilet bilgilerini görebilmektedir. Sayfamızda güvenliğini sağlamak için bilgileri get methodundan güvenli bir yapıya sahip olan post methodunu tercih ettik. Bununla birlikte admin sayfasına girişte aldığımız güvenlik önemli fonksiyon yapısı sayesinde sitenin XSS ve CSRF saldırılarına uğrama olasılığını azaltmış ve kötü niyetli hackerların hedefi olup açık hedef haline gelmesinden korunmuş oldu.

```
<?php
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    $adminuser = test_giris($_POST["username"]);
    $adminpassword = test_giris($_POST["password"]);
}

function test_giris($data)
{
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>
```

Şekil-20

Trim fonksiyonu ile uygulama ismindeki boşluklar kaldırıldı ve xss sebepli saldırıların önüne geçildi. Stripslashes fonksiyonu ile url kısmına slash işaretleri olan kısımlardan temizlendi. Htmlspecialchars fonksiyonu ile de farklı karakter yapısı ile sayfayı hackleme denemeleri yapılmasının önüne geçtik.

Bilet Bilgilerim		
Açık Bilet ID	Aktif Bilet	Pasif Bilet
2	İstanbul-İzmir	Kocaeli-Zonguldak
4	Zonguldak-İstanbul	Kocaeli-İstanbul

Şekil-15

Şekil-16

KAYIT OLMA SAYFASI

Bu kısımda kullanıcı sayfasına kayıt olarak aktif kullanıcı hesabı oluşturmak istedik. Kullanıcımız gerekli bilgileri girdikten sonra hesabı oluşmuş oluyor ve hesap bilgilerine girerek aktif ve pasif bilet bilgilerini görebiliyor.

KULLANDIĞIMIZ TEMPLATE

Bizim kullandığımız template yapısında bir ana sayfa ve ekstra bağımlılıkları olan bir template yapısıydı biz bu yapıyı tasarımı olarak kendi ihtiyaçlarımız doğrultusunda

Şekil-17

Models

Models klasöründe veritabanı ve oturumla ilgili işlemler yapılmaktadır. Bu kısımda yukarıda bahsettiğimiz login olma işlemleri yapılmaktadır. CodeIgniter'in Models yapısı ile aktif kullanıcı oluşturma sayfası yapmak istedik. Models yapısı aynı zamanda Codeigniter yapısı içinde veritabanı işlemleri ve oturum yönetimi yapılmasını sağlar. Biz de oluşturduğumuz modelde aktif session oluşturarak dinamikliği sağlamak istedik.

```
public function giris()
{
    $email = $_POST['email'];
    $password = $_POST['password'];
    $this->load->model('active_login_model');
    $user = $this->load->model('active_login_model');
    $session = session();
    $session->put('user', $user);
    if ($user) {
        return view('home');
    }
    else {
        return view('login');
    }
}

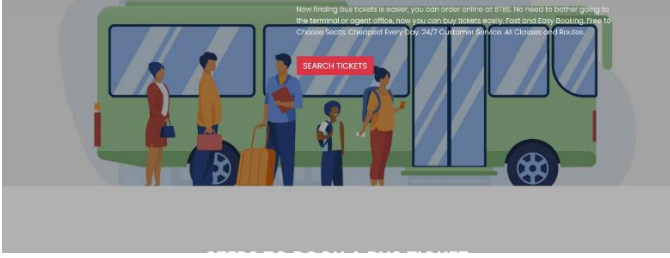
public function guncelle()
{
    $this->load->model('active_login_model');
    $user = $_POST['user'];
    $id = $_POST['id'];
    $password = $_POST['password'];
    $email = $_POST['email'];
    $session = session();
    $user = $session->get('user');
    $id = $user['id'];
    $this->load->model('active_login_model');
    $session = session();
    $user = $session->get('user');
    $password = $user['password'];
    $email = $user['email'];
    $session->put('user', $user);
    return view('home');
}
```

Şekil-16

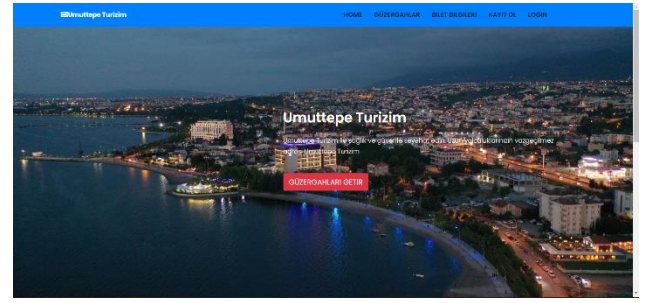
Kodumuzun bu kısmında kullanıcı güncelleme ve giriş işlemini oturum üzerinden yapmak istedik. Form verilerini aldık gerekli işlemleri yaptıktan sonra uygun view yapılarının dönmelerini sağladık. Oluşturduğumuz active_login_model üzerinden e-posta ve şifre sınıfından bir örnek oluşturduk. Bunun sonucunda session yapısından kullanıcı bilgilerini user anahtarı ile aldık ve kullanıcının ID'si belirlenir. Güncelle fonksiyonu ile kullanıcı bilgileri güncellenir. Son olarak hesabım view döner.

özelletirdik. İlave sayfalar koyduk ve template yapısının kendi içindeki bağımlılıkları ortadan kaldırdık. Daha sonrasında ise projemizin isterlerine göre template yapımızda düzenlemeye gittik. Aşağıda ise template yapımızın son hali bulunmaktadı

Şekil-17



Şekil-18



Şekil -19

KAYNAKÇA

[HTTPS://WWW.YOUTUBE.COM/WATCH?V=FNMBtI19foQ&LIST=PLoAXRfd1G8WI2Qg5Ly2JHZYU5z3uy7Qmv&AB_CHANNEL=SAVA%C5%9FDERSIM%C3%87ELIK](https://www.youtube.com/watch?v=FNMBtI19foQ&list=PLoAXRfd1G8WI2Qg5Ly2JHZYU5z3uy7Qmv&ab_channel=Sava%C5%9FDersim%C3%87elik)

[HTTPS://WWW.TUTORIALSPPOINT.COM/CODEIGNITER/INDEX.HTM](https://www.tutorialspoint.com/codeigniter/index.htm)

[HTTPS://WWW.YOUTUBE.COM/WATCH?V=kJoeCNIRLkQ&LIST=PL_F2F0OYAJ4_xGYQLWGSD9GI1UGNTTHkC&AB_CHANNEL=KABLOSUZKEDI](https://www.youtube.com/watch?v=kJoeCNIRLkQ&list=PL_F2F0OYAJ4_xGYQLWGSD9GI1UGNTTHkC&ab_channel=KABLOSUZKEDI)

PROJE LINKLERİ

GITHUB LINKİ: [HTTPS://GITHUB.COM/TYFNACICI/BUS-TICKET](https://github.com/tyfnacici/BUS-TICKET)

Canlı Link: <http://umutpeturizm.site/>