

Otobüs Bilet Satış Sistemi

Kadir Aydoğdu

Bilişim Sistemleri Mühendisliği
Teknoloji Fakültesi
221307085@kocaeli.edu.tr

Özet

Bu doküman, CodeIgniter frameworkü kullanılarak dinamik olarak geliştirilen, geliştirilmesinde PHP dili kullanılan otobüs bilet satış sitesini detaylı bir şekilde açıklamak ve nasıl tasarlandığını irdelemek için hazırlanmış bir rapordur.

Abstract— This document is a report prepared to explain in detail the bus ticket sales site, which was developed dynamically using the CodeIgniter framework and used PHP language in its development, and to examine how it was designed.

1. Giriş

Bu raporda; projemizde kullandığımız hazır responsive template, projenin hedeflerinin belirlenmesi, oluşturulacak projenin responsive bir yapıya sahip olması, siteyi oluşturan genel sayfaların hiyerarşik yapısını gösterilmesi, kısa açıklamalarla bilgiler verilmesi, framework detaylarından bahsedilmesi, siteyi oluşturan genel sayfaların düzeninin ve işleyişinin gösterilmesi, temel amaç olarak belirlenmiştir.

Sırasıyla proje sistemi, kullanılacak framework bileşenleri, veri tabanı tasarımı, site düzeni ve çalışma mantığı anlatılacaktır.

2. Site Gereksinimleri (Senaryo)

Otobüs bilet satış sitesinde tek bir firma olan umuttepe turizm, online olarak bilet satışı gerçekleştirecektir. 5 adet ilde çalışan bu firma her 8 saatte bir bulunan tüm illerden diğer illere sefer çıkaracaktır. Tek yön ve gidiş-dönüş olarak çalışmaktadırlar. Seferler listelenecek, etkileşimli koltuk arayüzü açıldığında boş koltuklardan koltuk seçilebilecektir. Yeşil koltuklar boş, kırmızı dolu koltukları simgelemektedir. Koltuğunu seçen kullanıcı devam ederek bilgilerini göreceği ve ödeme yapacağı sayfaya yönlendirecektir. Bu sayfada ödeme sayfasına geçince stripe tarafından sağlanan bir API tarafından ödemesi gerçekleşecektir. Siteye geri döndüğümüzde kullanıcının bilet sorgulaması için de bir kısım olacaktır. PNR numarasından sorgulanan biletin bilgileri listelenmelidir. Aynı zamanda tüm kullanıcıların üye

olması gerekmektedir. Üye işlemi olmadan bilet alımı gerçekleşemez. Siteye üye olan kullanıcı kendi panelinde aldığı biletleri görüntülemelidir. Burada biletlerini açığa alabilir ve QR kod oluşturabilir. Bu sitede aynı zamanda admin paneli olması gerekmektedir. Admin giriş yaptığında gelen iletişim mesajlarını görebilmeli, tüm seferleri listeleyebilmeli, otomatik eklenen seferlere manuel olarak da sefer eklemesi ve silme işlemlerini yapabilmesi gereklidir.

3. Framework Entegrasyonu

Kullanılması gereken framework CodeIgniter'dır. Bu framework dinamik PHP sayfalarının hazırlanmasında MVC yapısını kullanarak projeyi böl-parçala-yönet mantığına göre dizayn etmekte ve alışla gelmiş süreci hızlandırmaktadır. Framework resmi sitesinden edinildikten sonra gerekli ayarlamalar yapılması, projeye uyarlanması gerekmektedir.

Bu projede Codeigniter 3 sürümü kullanılacaktır.

3.1. Config Dosyalarının Yapılandırılması

Config klasörü içerisinde bulunan config, autoload ve database dosyalarını düzenlememiz gereklidir. Diğer dosyalarla fazla işimiz olmayacaktır. Config dosyasına girip sayfamızda kullanacağımız base url, language ayarlarımızı tanımlıyoruz. Ardından autoload dosyasına girip, açılışta yüklenmesini istediğimiz kütüphaneleri atıyoruz. Son olarak da database dosyasına girip kullandığımız sql server bilgilerini doğruluyoruz. Bu işlemler bittikten sonra siteye oluşturulmaya başlanabilir. Temel ayarlar bu kısımda bu kadar anlatılacaktır.

3.2. Template ve MVC Yapısı

İnternet üzerinden hazır responsive yapıda bootstrap temelli bir şablon edinmemiz gereklidir. Bu şablonu edindikten sonra MVC yapısına uygun hale getirmemiz gereklidir. MVC yapısı model-views-controller olarak bilinir. Yani model klasöründe veri tabanı bilgilerimiz, controllerda genel backend yönlendirmelerimiz ve

views klasöründe de site sayfalarının bulunması demektir. Dinamik bir site tasarlayacağımız için sitemizin header, navbar ve footerı ayrı ayrı parçalama işlemi yapmamız gerekmektedir. Views içerisine include adlı bir klasör oluşturup bölünmüş ayrı ayrı header ve footer sayfalarını ekliyoruz. Proje ana dizine çıkıp sayfalarda kullandığımız javascript, css ve bootstrap dosyalarını da assets isimli klasöre frontend ve backend olmak üzere iki klasör oluşturarak frontend klasörüne kopyalıyoruz. Views içerisine tekrar gelip admin için bir klasör oluşturuyoruz. Burada da kullanacağımız admin paneli için aynı işlemleri tekrar edip assets klasöründeki backend klasörüne yardımcı dosyalarımızı kopyalıyoruz. Site tasarımı ve MVC yapısını da bu şekilde halletmiş olduk. Şimdi site sayfalarını açıklayabiliriz.

4. Site İçeriği

Sitemizde anasayfa, bilet ara, bilet sorgula, giriş ve kayıt ol sayfaları olacaktır. Kullanıcı giriş yaptığında giriş yerine hesabım; kayıt ol yerine ise çıkış yap butonu olarak görünecektir. Bu kısımda sayfaların tek tek içeriklerinden, yapı ve işlevlerinden, kullanıldıysa veri tabanından bahsedilecek ve detaylı açıklamalarla anlatılacaktır.

5. Anasayfa

Anasayfada bizleri bir wallpaper karşılayacaktır. Burada bir buton bulunmaktadır. Bu buton bilet ara sayfasına yönlendirecektir. Alt tarafa doğru indikçe hizmet politikası ve mobil uygulaması sayfaları gösterilecektir. Uygulama temsili görsel olarak oluşturulduğu için butonda herhangi bir indirme linki yoktur. Sitenin estetik görünümü açısından popüler lokasyon görselleri bulunmaktadır. Bu sayfadan sonra müşteri hizmetleri için bir form bulunmaktadır. Şikayet ve öneri formu admin paneline girildiğinde mesajları göster kısmında listelenecektir.

6. Bilet Ara

Bu sayfada bizleri bir form karşılayacaktır. Bu formda seyahat bilgilerimizi girip istediğimiz tarihte ve yön belirterek bizlere seçtiğimiz kriterlerde seferleri listelemektedir. Bu sayfada “umt_seferler” tablosundan verileri çekerek, form verileri karşılaştırılacaktır. Bu filtrelemeye uygun seferler sonraki sayfada listelenecektir.

7. Seferler

Bu sayfada istenilen iki şehir arasındaki rotayı çizen bir Google Maps API kullanılmıştır. Alınan veriler rota çiziminde kullanıldı ve sayfa başında ilk bu api gösterildi. Sonrasında ise Seçilen rotaya uygun seferler tablosu listelendi. Kullanıcı istediği koltuğu seçip

devam edebilir veya geriye dönüp farklı bir şehir seçebilir. Bu kısımda koltuk seçilince session değişkeninde oturum bilgileri kontrol edilmektedir. Eğer oturum açılmışsa işleme devam etmekte aksi durumda giriş yap sayfasına yönlendirmektedir.

8. Ödeme Yap

Giriş yapıldıktan sonra koltuk seçip devam ettiğimizde bizi artık tüm seçimlerimizi gösteren bir tablo karşılamakta ve özel durum varsa seçmemizi istemektedir. Özel durumu olan kullanıcılar indirimden yararlanabileceklerdir. Ödeme yap sayfasına devam ettiğimizde bizden kart bilgileri istemektedir. Burada Stripe API entegrasyonu kullanıldığı için Stripe sayfasından test kartlarından birini yazıp işleme devam ediyoruz. Onayla butonuna tıkladıktan sonra “umt_biletler” tablomuza üyenin aldığı bilet kaydedilmektedir. Üye ilgili seferin ilgili koltuğunu aldığı için o koltuk artık bilet iptal edilene kadar dolu gözükecektir.

9. Bilet Sorgula

Bilet sorgula sayfasında basit bir text form vardır. Buraya bilet PNR kodunu girince “umt_biletler” tablosundan ilgili PNR sorgulanıp bulunursa ekrana tablo biçiminde listelemektedir. Bulunamazsa sayfada hata mesajı vermektedir. PNR kodu şöyle oluşmaktadır: “Kalkış il plaka + öğleden önce, sonra durumu + işlem yapılan saat + peron no + otobüs plakası” olarak belirlenmiştir.

10. Hesabım

Bu sayfa oturum değişkenleri kontrolünde girilebilir. Aksi takdirde erişilemez. Girildiğinde oturum değişkenlerinden alınan ad verisine göre kullanıcıya ismiyle karşılar. Kullanıcının id bilgisini kullanarak veri tabanını sorgular ve aldığı tüm biletleri ekrana tablo şeklinde listeler. Açığa al seçeneğini seçerse bileti iptal edilir ve tutarı bakiye hesabına yansır. Bakiye hesabı bileti alma işleminde öncelikli olarak kullanılır ve hesaptan biletin fiyatı düşülür. Biletlerim tablosunda yine QR kod oluşturarak biletine mobil ortamdan da QR kod okutarak erişebilir. Oluşturulan QR tablo yanına eklenir ve indirilebilirdir.

11. Admin Paneli

Admin paneline giriş yapmak için sağ üstte admin giriş linkine tıklayıp login sayfamıza erişebiliriz. Bu sayfanın giriş bilgileri herhangi bir tabloda tutulmamaktadır. Çünkü tek bir firma olduğu için gerek görülmemiştir. Hazır kullanıcı adı ve şifre ile giriş yapılmaktadır. “admin” ve “1234” bilgileri ile admin paneline giriş yapabiliriz. Bizi sidebar ve içerik kısmından oluşan sade bir tasarım karşılayacaktır. Burada direkt sefer ekleme sayfası gelmektedir. Burada otomatik sefer ekleyen olay

komutu dışında da manuel olarak da sefer eklenebilir. Seferleri görüntüle ekranından tüm kayıtlı seferler listelenir, arama yapılabilir ve ilgili sefer silinebilir. Bu sayfalara session oturum değişkeni kontrolü yapıldığı için herhangi bir şekilde url den erişilemez. En alt menüde ise şikayet/öneri formundan gelen mesajlar listelenmektedir. Son olarak da sağ üstte bir logout butonumuz bulunmaktadır. Buraya tıklayarak admin panelden çıkış yapabiliriz.

12. Veri Tabanı Modeli ve İlişkiler

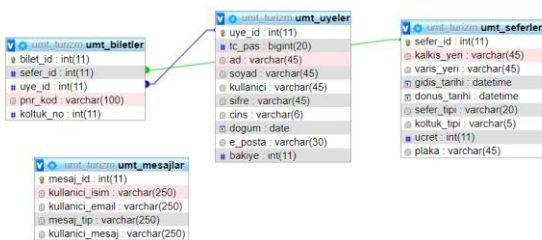
Veri tabanı ismi “umt_turizm” olarak belirlendi. 4 adet tablo bulunmakta 4 adet güncel çalışan event bulunmaktadır. Bu eventlar alt başlıklarda açıklanacaktır. Öncelikle tabloları açıklamaya başlayalım.

“umt_uyeler” tablosu üyenin kayıt olurken ki tüm bilgilerini ve ekstradan üyenin bakiye bilgisini de kayıt etmektedir. Her üye primary bir id’ye sahiptir.

“umt_seferler” tablosu cronjob olarak çalışan event’ların oluşturduğu tüm seferleri içeren tablodur. Her seferin benzersiz bir anahtarı vardır. Her seferin kalkış, varış, tarih, ücret ve ilgili seferin otobüsünün plaka bilgileri kayıt edilmektedir.

“umt_mesajlar” tablosu anasayfada bulunan müşteri hizmetleri formundan gelen talep ve önerileri kaydetmektedir. Her mesajın benzersiz bir anahtarı vardır. Gelen formdaki tüm verileri tutan sütunları vardır.

“umt_biletler” tablosu en önemli tablolardan birisidir. Kullanıcıların aldığı biletlerin kaydını tutan tablodur. Her biletin benzersiz bir anahtarı vardır. Bilet alınan seferin anahtarı, bileti alan üyenin anahtarı yabancı key olarak tutulmaktadır. Bu yüzden bu tablo ilişkili bir tablodur. Aynı zamanda bilet için oluşturulan PNR kodu da bu tabloya eklenmektedir. Kullanıcının seçtiği koltuk numarası da bu tablo kayıtlarında tutulmaktadır. İlgili seferin ve üyenin id si alınarak seçtiği koltuk bilgisini tuttuğu için gerçekten önemli ve hassas bir tablodur. Bu tablolar arasındaki ilişkiler aşağıdaki gibidir.



Şekil 1 – İlişkiler

13. Events

Son bölüm olarak seferlerin nasıl oluşturulduğu hangi eventın ne iş yaptığı bu bölümde anlatılacaktır. Seferler 8 saatte bir eklenmesi gerekli olduğundan gerekli kod verilmiştir. Bu kod her ilden ile sefer düzenleyecek şekilde kombinasyonlar yapıyor ta ki ilin kendisiyle eşleşene kadar. Plaka kodu için illerin gerekli plaka verileri verilir; kalkış ve varış ili plaka kodları arasına BSM25 olarak gelecek şekilde belirlenmiştir. Random sayıya göre dönüş tarihini null atamaktadır. Bu bize ilgili seferin gidiş veya gidiş-dönüş olacağını belirleyecektir. Aynı şekilde random olarak otobüsün koltuk tipi belirlenecektir. Bu işlemleri yapan event sefer ekle() eventıdır.

Tabloya sefer verilerini girdikten sonra dönüş tarihini null atanların sefer tipine “Gidiş”, Null olmayanlara ise “Gidiş-Dönüş” atanacak şekilde update komutu ile başlayan event sefer güncelle() ve sefer guncelle2 eventlarıdır. İki ayrı event olmaları hata bakımından daha sağlıklıdır bu yüzden iki ayrı update sorgusu olarak yazıldı. Son olarak da tarihi geçen, güncel olmayan sefer kayıtlarını silmesi için sefeleri sil() event’ı eklendi.

Özet olarak 8 saatte bir çalışan bu olay kümesi, öncelikle seferleri ekliyor sonra dönüş tarihine göre sefer tipini belirliyor ve son olarak da günü geçmiş seferleri tablodan silerek güncelliyor.

Not: Kullanılan sunucu sürekli çalışmadığı için events sürekli güncel kalamıyor bilgisayar kapanınca events scheduler de mecburen kapanıyor. Bu yüzden de seferler tablosunda güncel veriler olamayabilir. Ancak kullanılan yöntem ve kod parçacıkları sürekli çalışan bir sunucuda sorunsuz kullanılabilir.

14. Kaynakça

- [1] <https://codeigniter.com/>
- [2] Google AI, (2023) Bard, Google
- [3] OpenAI Resmi Web Sitesi, (2023), GPT-3.5
- [4] <https://github.com/>
- [5] <https://webimgo.com/blog/codeigniter-3-projesine-paytr-entegrasyonu-yapma>
- [6] <https://okantastan.com/codeigniter-iyzico-entegrasyonu/>
- [7] <https://github.com/alirizaoztetik/ci-iyzico>
- [8] <https://www.youtube.com/watch?v=IjuoJ91DJXw&list=PL14iwH7T84H9ZlhBKhRr-9eFu3GKmCOGe>
- [9] <https://www.youtube.com/watch?v=T4JUpyjo1oE&list=PL9VOSsas3g3ZGJZ90Xq9siGh3C22Xaa1M>
- [10] W3Schools
- [11] Wikipedia