

Bus Ticket Application

With PHP, Codeigniter4

Yunus Emre Gül
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
211307009@kocaeli.edu.tr

Yunus Emre Gölbaşı
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
211307043@kocaeli.edu.tr

Serhat Akçadağ
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
211307032@kocaeli.edu.tr

Abstract— In this article, we present the development process of a modern web-based application for bus ticket sales using CodeIgniter 4 framework. The application serves the needs of Bus Ticket Application, an intercity bus company, allowing users to book tickets conveniently. We delve into the purpose, technologies used, methodology, and stages of development to provide a comprehensive overview of the project.

Keywords—Codeigniter4, PHP, Bus Ticket, MYSQL

I. GİRİŞ

Bilet satış uygulamasının ihtiyaçları doğrultusunda, şirketin müşterilerine daha etkili ve kullanıcı dostu bir hizmet sunabilmesi için modern bir web tabanlı otobüs bilet satış uygulaması geliştirilmesi gerekmektedir. Bu proje, müşterilerin otobüs bileti satın alma sürecini kolaylaştırmayı, rotaları belirlemeyi, rezervasyon yapmayı ve ödemelerini gerçekleştirmeyi amaçlamaktadır. Bu makalede, proje amacı, beklentileri ve hedefleri detaylı bir şekilde incelenecek ve geliştirme sürecinin nasıl ilerlediği açıklanacaktır.

II. KULLANILAN TEKNOLOJİLER

A. CodeIgniter4

CodeIgniter 4, PHP tabanlı bir web uygulama geliştirme çatısıdır. Hızlı, hafif ve esnek bir yapıya sahiptir. MVC (Model-View-Controller) mimarisini destekler ve geliştiricilere web uygulamaları oluşturmak için güçlü bir altyapı sağlar. CodeIgniter 4, güvenlik, hata yönetimi, oturum yönetimi gibi önemli özellikleriyle tanınır.



B. MySQL

MySQL, açık kaynaklı bir ilişkisel veritabanı yönetim sistemidir. Uygulama tarafından kullanıcı bilgileri, bilet rezervasyonları, otobüs sefer bilgileri gibi verilerin depolanması ve yönetilmesi için MySQL kullanılmıştır. Güvenilirlik, performans ve geniş bir kullanıcı tabanı nedeniyle tercih edilen bir veritabanı çözümüdür.



C. Javascript

JavaScript, web uygulamalarında etkileşimli özellikler eklemek için kullanılan bir programlama dilidir. Uygulama içinde kullanıcı arayüzü etkileşimlerinin yönetilmesi ve QR kod oluşturmak için Qrious isminde JavaScript tabanlı bir kütüphane kullanılmıştır.



III. METODOLOJİ

Projemizin geliştirilme süreci, Codeigniter4 kullanarak istekleri karşılayacak şekilde bir tasarım yapmaktır. Bu tasarım, projenin başlangıcından itibaren dikkatlice planlanan ve adım adım ilerleyen bir süreci kapsamaktadır.

Bus Ticket Booking uygulamasının geliştirme süreci, kullanıcı ihtiyaçlarının belirlenmesi ve projenin temel hedeflerinin belirlenmesiyle başlamıştır. Bu aşamada, Umutepe Turizm'in gereksinimleri ve beklentileri analiz edilmiş ve bir zaman çizelgesi oluşturma uygulamasının ana özellikleri belirlenmiştir. Projenin başlangıcında, CodeIgniter 4'ün sağladığı modüler yapısı ve MySQL'in veritabanı entegrasyonu, uygulamanın temel mimarisini şekillendirmek için kullanılmıştır.

Uygulamanın geliştirme aşamasında, hem arka planda (backend) hem de ön planda (frontend) birçok önemli adım atılmıştır. Frontend tarafında, CodeIgniter 4'ün sağladığı kolaylık ve esneklikle kullanıcı arayüzü tasarlanmış ve geliştirilmiştir. Kullanıcı dostu bir arayüz, bilet satın alma işlemlerini kullanıcılara anlaşılır ve etkili bir şekilde iletmek amacıyla özenle planlanmıştır. Kullanıcı etkileşimleri, uygulamanın başarılı bir şekilde tamamlanabilmesi için büyük önem taşımaktadır.

Backend geliştirme aşamasında, MySQL kullanılarak kullanıcılara ait bilet ve rezervasyon bilgileri güvenli ve etkili bir şekilde depolanmıştır. Veritabanı entegrasyonu, uygulamanın veri yönetimi ihtiyaçlarını karşılayarak sistemin sağlam bir temel üzerine inşa edilmesini sağlamıştır.

Genel olarak, Bus Ticket Booking uygulamasının tasarımı, kullanıcıların ihtiyaçlarına odaklanan, etkileşimli ve kullanıcı dostu bir deneyim sunmayı hedeflemiştir. Başlangıçtan bitişe kadar dikkatlice planlanan bu süreçte, her adım projenin ana hedeflerine uygun olarak ilerlemiştir. Uygulamanın başarılı bir şekilde tamamlanması, CodeIgniter 4 ve MySQL gibi güçlü teknolojilerin etkili bir şekilde kullanılmasına dayanmaktadır.

IV. PROJENİN YAPIM AŞAMALARI

Bu bölümde projenin yapım aşamaları alt başlıklar halinde detaylandırılmıştır.

A. Genel Tasarım

Proje, genel tasarım aşamasında başlamıştır. Kullanıcı ihtiyaçları ve projenin temel hedefleri belirlenerek, otobüs bilet uygulamasının genel tasarımı şekillendirilmiştir. Bu aşamada, projenin ana özellikleri, kullanılacak teknolojilerin gereksinimleri, CodeIgniter4 araştırılmış ve istelere göre oluşturulacak modellerin genel şablonu planlanmıştır.

B. Codeigniter4 Proje Şablonu Oluşturulması

CodeIgniter, PHP dilinde yazılmış açık kaynaklı bir framework'tür. Hafif ve esnek yapısıyla bilinir, bu da uygulamaların hızlı bir şekilde geliştirilmesini sağlar. CodeIgniter, MVC (Model-View-Controller) mimarisini destekler ve bu sayede uygulamaların modüler bir şekilde oluşturulmasını sağlar.

Proje şablonunun oluşturulması aşamasında, CodeIgniter 4 kullanılarak temel proje yapısı hazırlanmıştır. CodeIgniter, modüler yapısı ve kolay kullanımıyla bilinir ve web uygulamalarının hızlı bir şekilde geliştirilmesini sağlar. Composer kurulumu yapıldıktan sonra, “composer create-project codeigniter4/appstarter projectName” komutuyla CodeIgniter projesi oluşturabilir ve “php spark serve” ile oluşturduğunuz uygulamayı tarayıcıda görebilirsiniz.

C. Veri Tabanı Entegrasyonu

Veri tabanı entegrasyonu aşamasında, MySQL veri tabanı kullanılarak veri tabanı yönetimi sağlanmıştır. Bu aşamada, CodeIgniter 4 ile MySQL'in entegrasyonu oldukça önemlidir çünkü bu, uygulamanın veri tabanı işlemlerinin düzgün bir şekilde gerçekleştirilmesini sağlar.

Entegrasyonu başarılı bir şekilde yapmak için öncelikle CodeIgniter4 proje dizininde bulunan ‘app/Config/Database.php’ dosyasında bulunan ayarları kendi MySQL veri tabanınıza göre ayarlamalısınız.

D. Modellerin Oluşturulması

Bu aşamada projenin istelerine göre modeller oluşturulup, MySQL'de bu modeller geçilmiştir.

Bu modeller, CodeIgniter 4'ün Model sınıfından türetilir ve gerekli veri tabanı işlemlerini gerçekleştirir. Model dosyalarını app/Models klasörü altında oluşturabilirsiniz. (Figure1). Modelleri oluşturduktan sonra ise “php spark migrate:create tableName” komutuyla migration dosyası oluşturup sonrasında ise “php spark migrate” komutuyla DB'de tablolar oluşturulmuş olur. Tablolar oluştuktan sonra

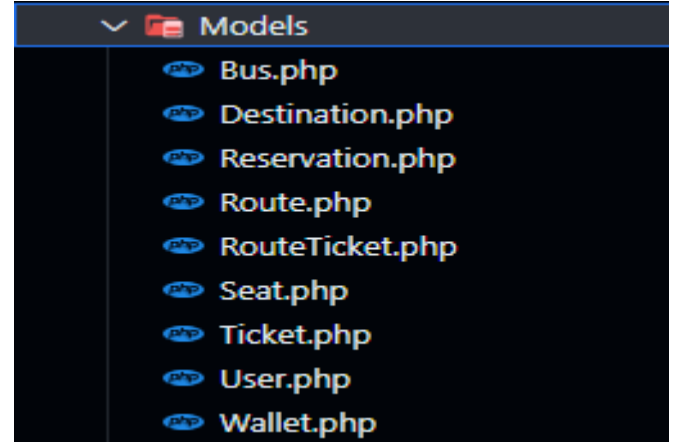


Figure 1

ise modellere has metodlar eklendi. Örnek olarak generatePNR metodu sadece Ticket modeline özgün olan bir metod olup kullanıcının bilet aldıktan sonra PNR numarası oluşturması için kullanılmıştır.(Figure2)

```
public function generatePnr($ticket_id)
{
    $plates = ['antalya' => '07', 'izmir' => '35', 'ankara' => '06', 'istanbul' => '34'];
    $ticketRoutes = new RouteTicket();
    $route_id = $ticketRoutes->where('ticket_id', $ticket_id)->first()['route_id'];
    $routeModel = new Route();
    $destination = $routeModel->getDestination($route_id);
    $bus = $routeModel->getBus($route_id);
    $dateTime = new DateTime();
    $ampm = $dateTime->format('a') == 'pm' ? 'ÖS' : 'ÖÖ';
    $formattedDateTime = $dateTime->format('dmYHis');
    $cityCode = $plates[$destination['starting_destination']];
    $plate = str_replace(' ', '', $bus['plate']);
    $pnrCode = $cityCode . $ampm . $formattedDateTime . 'D' . $plate;
    $ticketToUpdate = $this->find($ticket_id);
    $ticketToUpdate['pnr_number'] = $pnrCode;
    $this->save($ticketToUpdate);
    return $pnrCode;
}
```

Figure 2

E. Controller Oluşturulması

Controller oluşturulması aşamasında, uygulamanın iş mantığının yönetildiği ve gelen isteklerin işlendiği bölümdür. Bu adımda, her bir web sayfasının veya işlevselliğin

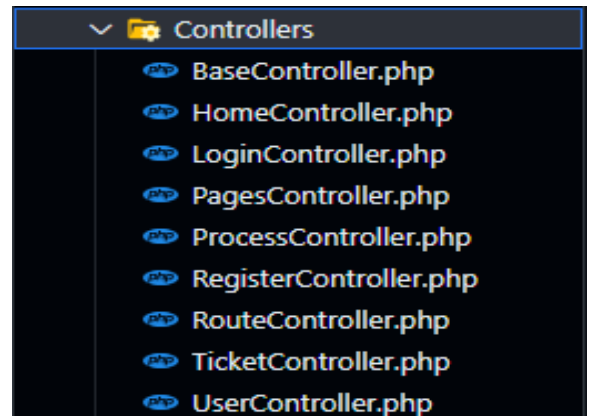


Figure 3

kontrolünü sağlayacak olan controller sınıfları oluşturulmuştur. Controller'lar, gelen istekleri alır, gerekli veri işlemlerini yapar ve sonuçları kullanıcı arayüzüne döndürür. Controller'lar(Figure3) ve Ticket Controller(Figure4).

```
<?php

namespace App\Controllers;

use App\Models\Route;

class TicketController extends BaseController
{
    public function searchTickets()
    {
        $starting_destination = $this->request->getGet('starting_destination');
        $ending_destination = $this->request->getGet('ending_destination');
        $going_date = $this->request->getGet('going_date');
        $returning_date = $this->request->getGet('returning_date');

        $routeModel = new Route();
        $tickets = $routeModel->searchTickets($starting_destination, $ending_destination, $going_date, $returning_date);

        return view('ticket_search_results', [
            'starting_destination' => $starting_destination,
            'ending_destination' => $ending_destination,
            'going_date' => $going_date,
            'returning_date' => $returning_date,
            'tickets' => $tickets,
        ]);
    }
}
```

Figure 4

F. Frontend Kısımının Oluşturulması

Frontend oluşturulması aşamasında, kullanıcı arayüzü tasarlanmış ve geliştirilmiştir. Bu aşama, kullanıcıların uygulamayı kullanırken karşılaştıkları görsel ve etkileşimli bileşenlerin oluşturulmasını içermektedir. Kullanıcı arayüzü, kullanıcıların uygulamayı etkili bir şekilde kullanmasını sağlayacak şekilde tasarlanmış ve geliştirilmiştir. Payment.php isimli view. (Figure 5)

The screenshot shows a web application interface. At the top, there's a navigation bar with 'Umuttepe Turizm' and 'Routes' on the left, and 'Log In' and 'Sign Up' on the right. Below the navigation bar, there's a 'Continue shopping' link. The main content area is divided into two sections. On the left, there's a 'Shopping cart' section with the text 'You have 2 items in your cart'. It lists two items: 'İstanbul - Ankara' for 2024-01-01, 34 XYZ 34, One Way, priced at ₺300, and 'Kocaeli - Ankara' for 2024-01-02, 34 XYZ 41, Round Trip, priced at ₺500. On the right, there's a 'Card details' section. It contains a 'Cardholder's Name' field, a 'Card Number' field with the value '1234 5678 9012 3457', and an 'Expiration' field with 'MM/YY' and 'Cvv' labels. Below these fields, there's a 'Subtotal' of ₺800, a 'Discount (%15)' of ₺120, and a 'Total Price' of ₺680. A '₺680 Checkout' button is at the bottom of the card details section.

Figure 5

Views'ler, kullanıcı arayüzünün HTML ve Bootstrap şablonu kullanılarak oluşturulduğu dosyalardır. Bu dosyalar,

kullanıcıların uygulamayı görüntülediği ve etkileşime girdiği sayfaları içermektedir. Her bir sayfa için ayrı bir view dosyası oluşturulmuş ve ilgili öğeler bu dosyalar içerisinde yer almıştır. (Figure6)

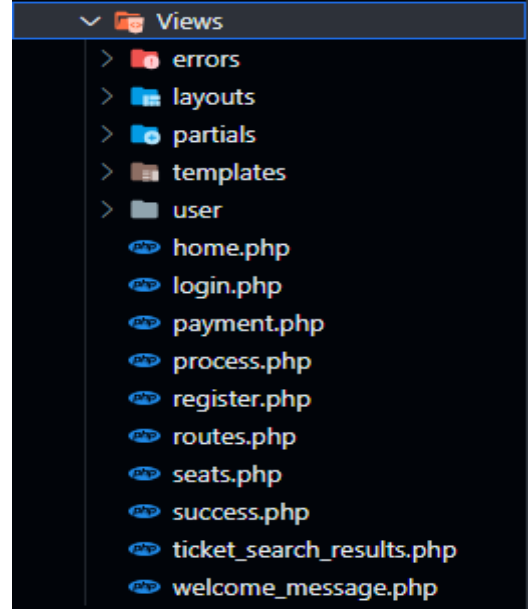


Figure 6

Örneğin, bilet arama sayfası için bir view dosyası oluşturulmuş ve bu dosyada bilet arama formu ve sonuçları görüntülenmiştir. Benzer şekilde, bilet satın alma sayfası için de ayrı bir view dosyası oluşturulmuş ve bu dosyada kalkış yeri, varış yeri, gidiş tarihi ve dönüş tarihi seçen kullanıcıya uygun biletlerin gösterilmesi amacı gerçekleştirilmiştir (Figure7)

The screenshot shows a search form for bus tickets. It has four input fields: 'Kalkış Yeri' (Departure Location) with a dropdown menu showing 'Antalya', 'Varış Yeri' (Arrival Location) with a dropdown menu showing 'Ankara', 'Gidiş Tarihi' (Departure Date) with a date picker showing '09.08.2024', and 'Dönüş Tarihi' (Return Date) with a date picker showing '09.08.2024'. There are two buttons at the top: 'Tek Yön' (One Way) and 'Gidiş Dönüş' (Round Trip). A 'Search' button is at the bottom right.

Figure 7

V. KAYNAKÇA

Otobüs bileti alımı, otobüs bileti rezervasyonu, koltuk seçimi, PNR numarası oluşturma, bilet arama, mail gönderme, qr kod oluşturma ve ödeme gibi işlevleri gerçekleştiren bu proje oluşturulurken Codeigniter4 Tutorial'dan yararlanılmış ve özgün olarak geliştirilmiştir.

[1] [Codeigniter4 Tutorial](#)

Kaynak kodlarını içeren Github reposu için [tıklayınız](#)