

YazLab - Umuttepe Turizm

<http://umuttepeturizm.bbs.tr/>
<https://github.com/kagantemizkan/kocaeli-turizm>

Eraycan Çoban
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
eraycancoban@gmail.com

Kağan Temizkan
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
kagantemizkan@gmail.com

Kağan Temizkan
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
fatiheneskilic9@gmail.com

Özet—Bu raporda*CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract. (Abstract)

Keywords—component, formatting, style, styling, insert (key words)

I. MVC MODELİ

MVC (Model-View-Controller), bir yazılım mimarisidir ve genellikle web uygulamalarının geliştirilmesinde kullanılır. MVC, uygulamayı üç temel bileşen altında düzenler: Model (Veri), View (Görünüm) ve Controller (Denetleyici).

A. Model (Veri)

Model, uygulamanın iş mantığını ve veri temsilini içerir.

Genellikle veritabanı işlemleri, veri manipülasyonu ve işleme kurallarının tanımlanması gibi görevleri üstlenir.

Veriye erişim ve veri manipülasyonu işlemlerini gerçekleştirir.

Veri değişikliklerini izler ve görünüme (View) bilgi sağlar.

Model, genellikle veritabanı tablolarını, ORM sınıflarını veya veri yapılarını temsil eder.

B. View (Görünüm)

View, kullanıcı arayüzünün görsel temsilini oluşturur.

Kullanıcıya sunulan bilgiyi görüntüler ve kullanıcı etkileşimini sağlar.

HTML, CSS ve bazen JavaScript kodlarından oluşur.

Verileri alır ve kullanıcıya uygun bir şekilde görüntüler.

Model tarafından sağlanan verileri temsil eder.

Genellikle birden fazla görünüm (view) olabilir, her biri farklı bir kullanıcı arayüzü sunabilir.

C. Controller (Denetleyici)

Controller, kullanıcı etkileşimlerini işler ve yönlendirir.

Kullanıcının yaptığı istekleri alır, işler ve sonuçları modele ve görünüme iletecek şekilde yönlendirir.

İstemci tarafından gelen istekleri kabul eder, işler ve doğru iş mantığı işlemlerini gerçekleştirmek için modeli kullanır.

Modelden gelen verileri, doğru görünüme iletecek şekilde düzenler ve görünüme iletir.

Web uygulamalarında genellikle URL yönlendirmeleri ve HTTP istekleri üzerinde çalışır.

D. MVC'nin İşleyişi

Kullanıcı bir istekte bulunur.

Bu istek, Controller tarafından alınır.

Controller, işlemi gerçekleştirmek için Model'e gerekli verileri iletir.

Model, gelen verilerle ilgili işlemleri gerçekleştirir.

Model, sonuçları Controller'a iletir.

Controller, sonuçları uygun View'e ileterek kullanıcı arayüzünü oluşturur.

Son olarak, oluşturulan kullanıcı arayüzü (View), kullanıcıya sunulur.

MVC modeli, uygulamanın farklı katmanlarını net bir şekilde ayırarak kodun düzenli ve bakımı kolay olmasını sağlar. Bu nedenle, birçok modern web uygulamasında yaygın olarak kullanılan bir tasarım desendir.

II. PHP

PHP, "Hypertext Preprocessor"ın kısaltmasıdır ve sunucu tarafı bir programlama dili olarak kullanılır. Rasmus Lerdorf tarafından 1994 yılında geliştirilmeye başlanmıştır ve o zamandan beri web geliştirme dünyasında popülerliğini artırmıştır. PHP, dinamik web siteleri ve web uygulamaları oluşturmak için kullanılır.

A. PHP'nin Özellikleri

1) Esneklik

PHP, HTML ile kolayca bir araya gelip dinamik içerik oluşturabileceğiniz esnek bir dil sağlar.

2) Kolay Öğrenme Eğrisi

Diğer dillere göre PHP'nin öğrenme eğrisi oldukça düşüktür, bu da onu yeni başlayanlar için ideal bir seçenek yapar.

3) Geniş Kullanıcı Tabanı

PHP'nin geniş bir kullanıcı tabanı vardır ve bu da sorunlarla karşılaştığınızda çözümler bulmanın kolay olmasını sağlar.

III. COMPOSER

Composer, PHP paketlerinin yönetimi için kullanılan bir bağımlılık yöneticisidir. PHP topluluğunun gelişen ihtiyaçlarına cevap olarak oluşturulan bu araç, dış kaynaklı kütüphanelerin kolayca projeye dahil edilmesini ve güncellenmesini sağlar. Composer, PHP projelerinin bağımlılıklarını yönetmek için gereken tüm işlemleri otomatikleştirir ve hızlı bir şekilde yapılmasını sağlar.

A. Composer'ın Özellikleri:

1) Bağımlılık Yönetimi:

Composer, PHP projelerinizin bağımlılıklarını yönetmek için kullanılır. Projelerinizde kullanmak istediğiniz dış kütüphaneleri kolayca ekleyebilir ve güncelleyebilirsiniz.

2) Otomatik Yükleme ve Güncelleme:

Composer, projenizin bağımlılıklarını otomatik olarak yükler ve günceller. Bu, geliştirme sürecinizi hızlandırır ve yönetimi kolaylaştırır.

3) Proje Bağımlılıklarının Tanımlanması:

composer.json dosyası aracılığıyla projenizin bağımlılıklarını belirleyebilirsiniz. Bu dosya, projenin ihtiyaç duyduğu tüm dış kütüphaneleri ve sürümlerini içerir.

PHP ve Composer, modern web geliştirme süreçlerinde önemli birer araçtır. PHP, sunucu tarafı uygulama geliştirmenin temel taşıdırken, Composer projelerinizde dış kütüphaneleri kolayca yönetmenize olanak tanır. Bu nedenle, her iki aracı da öğrenmek ve kullanmak, başarılı bir PHP geliştirici olmanın önemli bir parçasıdır.

IV. CODEIGNITER

CodeIgniter, PHP tabanlı web uygulamaları geliştirmek için kullanılan bir PHP çatısıdır. CodeIgniter, 2006 yılında EllisLab tarafından geliştirilmeye başlanmıştır ve hafif, esnek yapısıyla dikkat çekmektedir. Birçok geliştirici tarafından tercih edilen bu çatı, basit ve hızlı uygulama geliştirmeyi sağlar.

A. Neden CodeIgniter?

CodeIgniter, PHP tabanlı web uygulamaları geliştirmek isteyen geliştiriciler için birçok avantaj sunar. İşte CodeIgniter'i tercih etmenin bazı nedenleri:

1) Hafif ve Performanslı:

CodeIgniter, hafif bir çatıdır ve minimum sistem gereksinimleri ile çalışır. Bu da uygulamalarınızın hızlı bir şekilde çalışmasını sağlar ve sunucu kaynaklarını daha az kullanır.

2) Kolay Öğrenme Eğrisi

CodeIgniter, PHP geliştiricileri için kolay öğrenilebilir bir çatıdır. Temel PHP bilgisine sahip olan herkes, kısa sürede CodeIgniter ile web uygulamaları geliştirmeyi öğrenebilir.

3) Esnek Mimari

CodeIgniter, MVC (Model-View-Controller) mimarisini temel alır, ancak bu mimariye katı kurallar getirmez. Bu da geliştiricilere esneklik sağlar ve uygulamalarını istedikleri gibi organize etmelerine olanak tanır.

4) Kapsamlı Belgelendirme

CodeIgniter, kapsamlı bir belgelendirme ile birlikte gelir. Bu belgelendirme, geliştiricilerin hızlı bir şekilde başlamasına ve çözümler bulmasına yardımcı olur. Ayrıca, aktif bir topluluğa sahiptir, bu da destek ve yardım almanızı kolaylaştırır.

B. CodeIgniter'in Özellikleri:

1) MVC (Model-View-Controller) Mimari

Uygulamalarınızı düzenli ve modüler bir şekilde organize etmenizi sağlar.

2) Güvenlik

Cross-Site Request Forgery (CSRF) koruması, SQL enjeksiyonu önlemleri ve daha fazlası gibi güvenlik önlemleri sağlar.

3) Form ve Veri Doğrulama Kütüphanesi

Formların kolayca oluşturulmasını ve veri doğrulamasının yapılmasını sağlar.

4) Oturum Yönetimi

Kullanıcı oturumlarını yönetmek için kullanışlı bir kütüphane sunar.

5) Çalışma Ortamı Ayarları

Farklı ortamlar için yapılandırma dosyalarını yönetmeyi kolaylaştırır (örneğin, geliştirme, üretim).

6) Veritabanı Desteği

Çeşitli veritabanı sürücülerini destekler ve veritabanı işlemlerini kolaylaştırır.

CodeIgniter, PHP geliştiricileri için güçlü ve hızlı bir web uygulama geliştirme çatısıdır. Hafif yapısı, kolay öğrenme eğrisi ve esnek mimarisi sayesinde birçok geliştirici tarafından tercih edilmektedir. Hızlı ve güvenli uygulamalar geliştirmek isteyenler için CodeIgniter, ideal bir seçenektir.

V. REACT

React, Facebook tarafından geliştirilen ve modern web uygulamaları oluşturmak için kullanılan bir JavaScript kütüphanesidir. Web geliştirme dünyasında hızla popülerlik kazanan React, esnek yapısı ve performans odaklı yaklaşımıyla dikkat çekmektedir.

A. Neden React?

React, web uygulamaları geliştirmek isteyen birçok geliştirici tarafından tercih edilmektedir. İşte React'ı tercih etmenin bazı nedenleri:

1) Component Tabanlı Yapı

React, bileşen tabanlı bir yapıya sahiptir. Uygulamanızı küçük, yeniden kullanılabilir bileşenlere böler ve her bir bileşenin kendi içinde bağımsız olarak çalışmasını sağlar. Bu, uygulamanızı daha modüler hale getirir ve bakımını kolaylaştırır.

2) Sanal DOM Kullanımı

React, sanal DOM (Document Object Model) kullanarak performansı artırır. Değişiklikleri gerçek DOM'a uygulamak yerine, önce sanal DOM üzerinde değişiklikler yapar ve ardından sadece gereken değişiklikleri gerçek DOM'a yansıtır. Bu, uygulamanın daha hızlı ve verimli çalışmasını sağlar.

3) Tek yönlü veri akışı (One-Way Data Binding)

React, tek yönlü veri akışı ilkesini benimser. Bu, verilerin tek bir yönde (genellikle üstten aşağıya) akmasını sağlar. Verilerin değişimini izlemek ve hata ayıklamak daha kolaydır, bu da kodun daha okunabilir ve sürdürülebilir olmasını sağlar.

4) Geniş Topluluk Desteği

React, geniş bir topluluk tarafından desteklenir. Bu, geliştiricilerin sorunlarını çözmek, öneriler almak ve en iyi uygulamaları paylaşmak için birçok kaynağa erişebilecekleri anlamına gelir. Ayrıca, birçok ücretsiz ve ücretli kaynak mevcuttur.

B. React'ın Özellikleri

React, modern web uygulamaları oluşturmak için bir dizi özellik sunar:

1) JSX Söz Dizimi

JavaScript ile HTML'yi bir araya getirerek bileşenleri oluşturmayı kolaylaştırır.

2) Bileşen Yaşam Döngüsü Yönetimi

Bileşenlerin oluşturulması, güncellenmesi ve yok edilmesi gibi yaşam döngüsü olaylarını yönetir.

3) Hızlı Rendeleme

Sanal DOM kullanarak, uygulamanın hızlı bir şekilde yeniden çizilmesini sağlar.

4) Kuralı ve Tek Yönlü Veri Akışı

Verilerin yönetimini kolaylaştırır ve beklenmedik hataları azaltır.

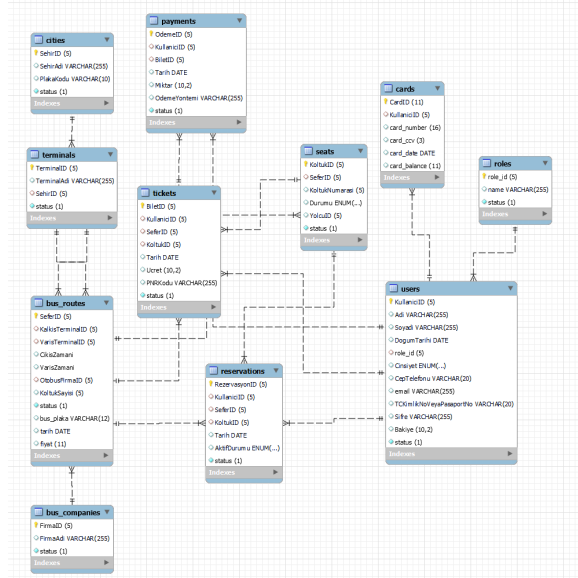
5) Bileşenler Arası İletişim

Bileşenler arasında veri iletişimini ve etkileşimi kolaylaştırır.

React, modern web uygulamaları geliştirmek için güçlü bir araçtır. Esnek yapısı, performans odaklı yaklaşımı ve geniş topluluk desteği sayesinde, birçok geliştirici tarafından

tercih edilmektedir. React kullanarak, güzel, hızlı ve modüler web uygulamaları oluşturabilirsiniz.

VI. MYSQL



Verilen SQL ifadeleri, MySQL veritabanında tablolar oluşturur ve bu tablolar arasında ilişkiler kurar. İlgili dış anahtarlar sayesinde tablolar arasında bağlantılar sağlanmıştır. Tüm tablolar birincil anahtarlar ile ve gerekli dış anahtarlar ile donatılmıştır.

Her tablonun yapısı ve ilişkileri aşağıda özetlenmiştir:

1) bus_companies:

FirmaID birincil anahtardır.
FirmaAdi şirketin adını tutar.

2) cities:

SehirID birincil anahtardır.
SehirAdi şehirlerin adını tutar.
PlakaKodu şehirlerin plaka kodlarını tutar.

3) terminals:

TerminalID birincil anahtardır.
TerminalAdi terminalin adını tutar.
SehirID ilgili şehrin ID'sini tutar. Bu alan cities tablosuna bir dış anahtar olarak bağlanır.

4) bus_routes:

SeferID birincil anahtardır.
KalkisTerminalID ve VarisTerminalID ilgili terminallerin ID'lerini tutar. Her ikisi de terminals tablosuna dış anahtar olarak bağlanır.
CikisZamani ve VarisZamani seferin kalkış ve varış zamanlarını tutar.
OtobusFirmaID ilgili otobüs firmasının ID'sini tutar. Bu alan bus_companies tablosuna bir dış anahtar olarak bağlanır.
KoltukSayisi seferdeki toplam koltuk sayısını tutar.

5) roles:

role_id birincil anahtardır.
name rolün adını tutar.

6) *seats:*

KoltukID birincil anahtardır.

SeferID ilgili seferin ID'sini tutar. Bu alan bus_routes tablosuna bir dış anahtar olarak bağlanır.

KoltukNumarasi koltuğun numarasını tutar.

Durumu koltuğun durumunu tutar: Bos, Rezerve Edilmiş veya SatınAlınmış.

7) *users:*

KullanıcıID birincil anahtardır.

Adi, Soyadi, DogumTarihi, Cinsiyet, CepTelefonu, email, TCKimlikNoVeyaPasaportNo, Sifre, Bakiye kullanıcı bilgilerini tutar.

role_id kullanıcının rolünün ID'sini tutar. Bu alan roles tablosuna bir dış anahtar olarak bağlanır.

8) *tickets:*

BiletID birincil anahtardır.

KullanıcıID, SeferID, KoltukID, Tarih, Ucret, PNRKodu bilet bilgilerini tutar.

KullanıcıID kullanıcının ID'sini tutar. Bu alan users tablosuna bir dış anahtar olarak bağlanır.

SeferID seferin ID'sini tutar. Bu alan bus_routes tablosuna bir dış anahtar olarak bağlanır.

KoltukID koltuğun ID'sini tutar. Bu alan seats tablosuna bir dış anahtar olarak bağlanır.

9) *reservations:*

RezervasyonID birincil anahtardır.

KullanıcıID, SeferID, KoltukID, Tarih, AktifDurumu rezervasyon bilgilerini tutar.

Diğer tablolardan gelen dış anahtar ilişkileriyle bağlantılıdır.

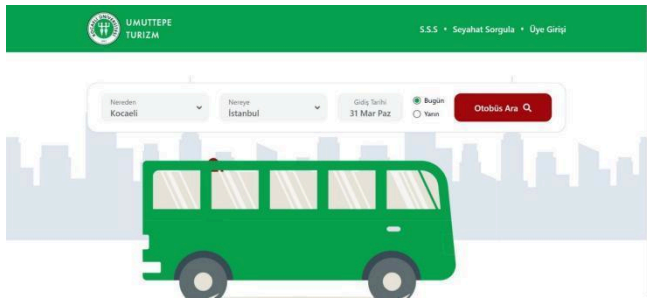
10) *payments:*

OdemeID birincil anahtardır.

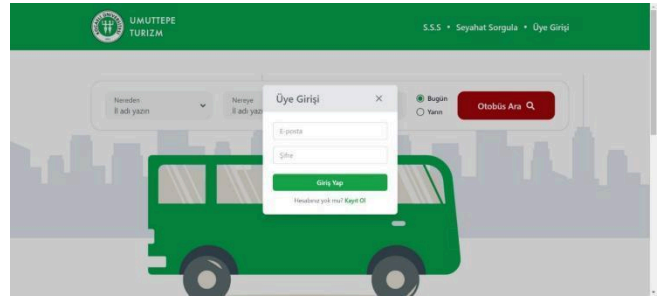
KullanıcıID, BiletID, Tarih, Miktar, OdemeYontemi ödeme bilgilerini tutar.

Diğer tablolardan gelen dış anahtar ilişkileriyle bağlantılıdır.

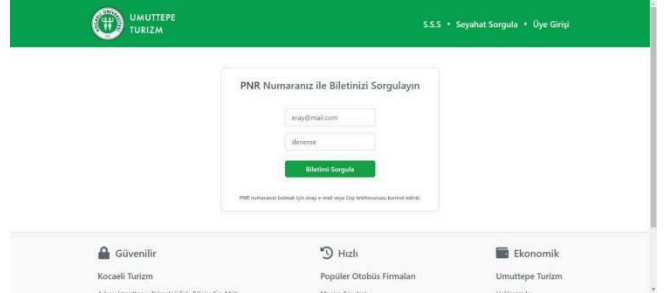
VII. FRONTEND



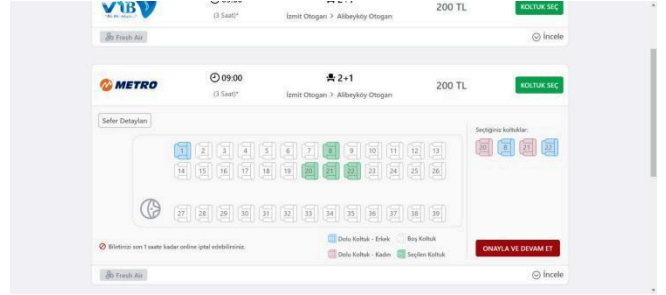
Anasayfa



Üye Girişi



PNR Sorgulama



Koltuk Seçme