

ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BLM3522

BULUT BİLİŞİM VE UYGULAMALARI

RAPORU

Proje 1 ve Proje 4

Ömer Asaf DEMİR

20290240

GitHub: AsafDemir/BulutBilisimVeUygulamalariBut

<https://github.com/AsafDemir/BulutBilisimVeUygulamalariBut>

Video:

https://drive.google.com/file/d/17USlsY2_8X56rcg6qGNcldFSVJ9cSVPw/view?usp=sharing

Proje 1: Çift Katmanlı Web Uygulaması (Web API + Frontend)

1. Proje Genel Bakış

Çay Ocağı Yönetim Sistemi, kurumsal ortamlarda çay ve içecek sipariş süreçlerini dijitalleştiren, kullanıcı ve yönetici rolleri ile erişim kontrollü modern bir web uygulamasıdır. Azure bulut altyapısı üzerinde çalışarak yüksek erişilebilirlik, ölçeklenebilirlik ve güvenlik sağlamaktadır.

2. Kullanılan Teknolojiler ve Uygulamalar

Backend Teknolojileri

- .NET Core Web API
- Entity Framework Core (ORM)
- JWT Authentication

Projenin backend tarafı için yüksek performans, düşük kaynak tüketimi ve mikroservis mimarilerine uygun olması nedeniyle .NET Core Web API tercih edildi. RESTful prensiplerine uygun API endpoint'leri oluşturuldu ve HTTP metodları (GET, POST, PUT, DELETE) etkin kullanıldı.

Entity Framework Core ile ORM aracılığıyla veritabanı işlemleri nesne yönelimli olarak yönetildi. Code First yaklaşımı ile veritabanı tasarımı doğrudan C# sınıflarından oluşturuldu, migration kullanımıyla versiyonlama sağlandı ve veri erişim performansı Lazy Loading ve Eager Loading yöntemleriyle optimize edildi.

JWT Authentication kullanılarak güvenli ve stateless kimlik doğrulama gerçekleştirildi. Token'ların içerisinde kullanıcı bilgileri ve roller taşınarak yetkilendirme işlemleri hızlandırıldı.

Frontend Teknolojileri

- Angular 19.0.0
- Angular Material ve CDK
- RxJS
- TypeScript 5.6.2
- Reactive Forms

Frontend kısmında Angular framework'ü kullanıldı. Angular'ın modüler ve komponent tabanlı mimarisi sayesinde sürdürülebilir bir yapı sağlandı. Angular Material kullanılarak kullanıcı arayüzünde modern ve profesyonel bir görünüm oluşturuldu.

RxJS ile reaktif programlama uygulandı, gerçek zamanlı veri akışları yönetildi. Reactive Forms ile kullanıcı girişlerinin anlık doğrulaması ve hata yönetimi gerçekleştirildi.

Veritabanı Teknolojileri

- PostgreSQL

Projenin veri katmanında PostgreSQL kullanıldı. PostgreSQL'in ilişkisel veri yönetimi, sorgu performansı ve JSON desteği tercih sebebi oldu. Entity Framework Core aracılığıyla PostgreSQL üzerinde veritabanı yapıları oluşturuldu ve Azure servisleri ile otomatik yedekleme ve güvenlik yönetimi sağlandı.

Bulut Platformları

- Azure App Services
- Azure Database for PostgreSQL
- Azure Application Insights
- Azure DevOps (CI/CD)

Microsoft Azure platformu, uygulamanın barındırılması için kullanıldı. Azure App Services ile otomatik ölçeklendirme ve yönetim kolaylığı sağlandı. Azure Database for PostgreSQL ile veri güvenliği artırıldı ve performans izleme Azure Application Insights üzerinden gerçekleştirildi. Azure DevOps ile CI/CD süreçleri otomatize edildi, kod kalitesi sürekli kontrol edildi.

3. Sistem Mimari Yapısı

Katmanlı Mimari

- Sunum Katmanı (Angular Components, API Controllers)
- İş Mantığı Katmanı (Services)
- Veri Erişim Katmanı (Entity Framework, Data Layer)

Katmanlı mimari ile kod tabanı modüler hale getirildi. Bu mimari sayesinde bakım ve geliştirme süreçleri basitleştirildi.

RESTful API Tasarımı

RESTful API prensiplerine uygun olarak, açık, anlaşılır ve tutarlı endpoint'ler geliştirildi. HTTP metodlarıyla kullanıcı ve yönetici işlemleri net bir şekilde tanımlandı.

4. Güvenlik ve Kullanıcı Yönetimi

- JWT Authentication
- Rol tabanlı erişim kontrolü
- CORS Yapılandırması

JWT Authentication ile kullanıcı ve yetki yönetimi gerçekleştirildi. Kullanıcı rolleri tanımlanarak, rol bazlı erişim kontrolleri uygulandı ve CORS yapılandırmasıyla frontend güvenliği sağlandı.

5. Sipariş ve Veri Yönetimi

Kullanıcılar için kolaylaştırılmış sipariş arayüzleri oluşturuldu. Reactive Forms sayesinde kullanıcı verileri anlık doğrulanarak hatalı veri girişleri önlendi. RxJS kütüphanesinden BehaviorSubject kullanılarak siparişlerin gerçek zamanlı takibi mümkün hale getirildi.

Yöneticiler için kapsamlı sipariş yönetimi sağlandı. Yönetim panelinde sipariş durumları değiştirilebilir, filtrelenebilir ve iptal edilebilir. Ayrıca detaylı raporlama ve görsel analiz araçları ile yöneticilere kapsamlı veri analizi imkânı sunuldu. Yönetici panelinde içecek stokları, fiyatlandırma ve oda yönetimi için CRUD işlemleri sağlandı.

6. Kullanıcı Arayüzü Tasarımı

- Material Design
- Responsive ve erişilebilir tasarım

Material Design ilkeleri doğrultusunda kullanıcı dostu, responsive ve erişilebilir arayüz geliştirildi.

7. Performans Optimizasyonu

- Lazy loading
- Database Indexing
- Query Optimization

Performans iyileştirmeleri için lazy loading uygulandı, sık sorgulanan verilere indeksleme yapıldı ve sorgular optimize edildi.

8. Test ve Kalite Güvence

- Jasmine ve Karma (birim testleri)
- Protractor (end-to-end testleri)
- ESLint ve Prettier (kod kalitesi)
- Azure DevOps (CI/CD pipeline)

Test ve kalite süreçleri otomatize edilerek kod kalitesi yüksek seviyede tutuldu. CI/CD süreçleri ile sürekli entegrasyon ve dağıtım süreçleri yönetildi.

9. Proje Sonuçları

- Sipariş sürecinde %40 hızlanma
- Hata oranlarında %70 düşüş
- Kullanıcı memnuniyetinde artış

Projenin hedeflerine ulaşılmış olup dijital dönüşüm sağlanarak verimlilik, doğruluk ve kullanıcı memnuniyeti artırıldı.

10. Gelecek Geliştirmeler

- Mobil uygulama
- Gerçek zamanlı bildirimler
- Yapay zeka önerileri
- Otomasyon ve IoT entegrasyonu

Mobil uygulama, gerçek zamanlı bildirimler, yapay zeka destekli öneriler ve IoT entegrasyonu ile projenin kapsamı ve işlevselliği genişletilmesi planlanmaktadır.

Proje 4: Otomatik Ölçeklendirme ve Yük Yönetimi

Otomatik Ölçeklendirme ve Yük Yönetimi

Çay Ocağı Yönetim Sistemi, Microsoft Azure platformu üzerinde barındırılmakta olup, yüksek erişilebilirlik ve performans hedefleri doğrultusunda **otomatik ölçeklendirme** destekli bir yapıya geçirilmiştir. Uygulama, Azure App Services hizmetinde **Standard S1** planında çalışacak şekilde yapılandırılmış ve Azure portal üzerinden özel autoscale kuralları tanımlanmıştır.

Bu yapılandırma kapsamında:

- Uygulamanın CPU kullanım oranı %70'in üzerine çıktığında, Azure tarafından otomatik olarak yeni bir instance başlatılmakta,
- CPU kullanımı %30'un altına düştüğünde ise fazla instance'lar otomatik olarak kapatılmaktadır.

Bu sayede sistem dinamik olarak **yatay ölçeklenebilir** hale gelmiştir. Azure Load Balancer aracılığıyla yük dengelemesi sağlanmış, ani trafik artışlarında sistem performans kaybı yaşamadan hizmet verebilir duruma getirilmiştir.

Bununla birlikte:

- Azure Application Insights ile sistem performansı ve hata izleme gerçekleştirilmektedir.
- Azure Database for PostgreSQL ile ilişkisel veri yönetimi ve otomatik yedekleme süreçleri güvenli biçimde sürdürülmektedir.
- Azure DevOps entegrasyonu sayesinde, CI/CD süreçleriyle sürekli teslimat ve güncelleme operasyonları sağlanmıştır.

Bu çalışmalar neticesinde uygulama, sadece fonksiyonel olarak değil, **altyapı düzeyinde de modern, güvenli ve sürdürülebilir bir yapıya kavuşmuştur**. Projenin bulut mimarisi sayesinde, ileriye dönük ölçekleme, entegrasyon ve bakım süreçleri kolaylaştırılmıştır.