

FULL STACK E-TİCARET

- Proje başlangıç: 01 Aralık 2025
- Proje Teslim: 14 - 15 Şubat – Sunum yapılacak
- Teslim: Github Repo Bağlantısı

Katılımcılar:

1. Ahmet Demircan
2. Furkan İşınak
3. Haktan Erbaktı
4. Mustafa Ulu
5. Mümine Muroğlu
6. Serhat Ezen
7. **Tuğba Nur Şimşek**
8. **Ufuk Abravacı**
9. **Ömer Kafkas**
10. İsmail Sümer
11. Çağlar Kızılaslan

Proje Açıklaması ve Amacı

Bu proje, farklı şirketlerin kendi e-ticaret hizmetlerini oluşturabilmelerini sağlayan çok katmanlı bir **E-Ticaret Servis Platformu** geliştirmeyi amaçlamaktadır. Sistem; şirketlerin kayıt olarak kendi ürünlerini yönetebileceği, sipariş takip edebileceği, müşteri bilgilerini görebileceği ve tamamen kendilerine ait API'ler üzerinden bu verileri dış sistemlerde (Angular, iOS, Android vb.) kullanabileceğи bir altyapı sunar.

Proje, gerçek bir kurumsal e-ticaret yazılımının üç katmanda (Admin Panel (Şirket Yönetim) – REST API – Frontend) nasıl geliştirildiğini uygulamalı olarak öğretmeyi amaçlayan kapsamlı bir full-stack geliştirme çalışmasıdır. **Kurumsal mimari, katmanlı yapı, role-based güvenlik, API mimarisi, frontend client entegrasyonu ve modern yazılım geliştirme pratiklerini** tek bir projede deneyimlemesi hedeflenmiştir.

Kazanımlar:

- **Admin panel hizmeti sunan bir MVC uygulaması** geliştirmesi,
- **Modern REST API prensipleri** ile servis üretmesi,
- **JWT ve session tabanlı kimlik doğrulama ve yetkilendirme** kurgulaması,
- **Angular ile API'den gelen verileri işleyen bir storefront** yazması,
- **Global Exception Handling, Validation, Interceptor, Logging** gibi profesyonel geliştirme standartlarını uygulaması,
- **Clean Architecture** prensiplerine uygun bir yapı kurması hedeflenmektedir.

Teknolojiler:

- **Backend MVC:** .NET Core 9 MVC
- **Backend API:** .NET Core 9 REST API
- **Frontend:** Angular 20
- **Auth:** JWT + Role Based + Session(MVC)
- **Database:** Sqlite, MongoDB veya SQL Server seçim geliştiricininindir.

I. MVC (Admin Panel)

MVC tarafı yalnızca **yönetim paneli** olacak ve şirketler bu panel vasıtası ile aşağıdaki hizmetleri kendi girişleri üzerinden yöneteceklerdir.

1. Modüller (Geliştirilecek Özellikler)

- Dashboard
 - **Toplam ürün sayısı**
 - **Toplam Kullanıcı sayısı**
 - Toplam Sipariş Sayısı
 - **Toplam Yorum Sayısı**
- **Ürün Yönetimi**
 - **Ürün listeleme**
 - **Ürün sil / düzenleme**
 - **Ürün resimleri yönetimi (Araştırma)**
 - **Kategoriye, Markaya bağlama,**
- **Kategori Yönetimi**
- **Marka Yönetimi**
- **Sipariş Yönetimi**
- **Müşteri Yönetimi**
- **Yorum Yönetimi**
- **Kargo Ayarları**
- **Slider/Banner Yönetimi (Araştırma)**
- **Kullanıcı ve Rol Yönetimi**
- **Genel Ayarlar**

2. Güvenlik Gereksinimleri

- **MVC uygulaması session ile oturum denetimleri yapacak.**
- **Kullanıcı/rol bazlı sayfa yetkilendirmesi yapılacak.**
- **Admin giriş ekranı MVC içinde olacak.**

3. Zorunlu Yapılması Gerekenler

- **MVC tarafında veri tutma/dummy data yasaktır.**
- **Rich Text editör (örn. Summernote/TinyMCE) kullanılmalıdır. (Araştırma)**
- **MVC tarafında **Exception Filter** veya **Custom Error Handling** zorunludur.**
- **MVC tarafı için **Clean Architecture** benzeri katmanlı yapı kullanılmalıdır.**
- **Tüm formlarda **Model Validation** bulunmalıdır.**

II. REST API

API tüm uygulamanın çekirdeği olacaktır.
Hem MVC hem Angular bu API'yi kullanacaktır.

1. Modüller

(API tarafından aşağıdaki tüm modüller için apiler yazılmalıdır)

- **Products – Sayfalama**
- **Products - Search**
- Categories
- Brands
- Customers
- Orders (status update, refund, cancel, tracking)
- **Reviews**
- Banners
- **Admin Users & Roles**

2. Güvenlik Gereksinimleri

- **JWT Authentication**
- **Role Based Authorization**
- Şirket bazlı API Key desteği

3. Zorunlu Yapılması Gerekenler

- **Global Exception Handling Middleware** zorunludur.
- **Logging** (Serilog veya benzeri) zorunludur.
- **Swagger–UI** mutlaka aktif olmalıdır.
- **DTO** kullanımı zorunludur (entity direkt API'ye açılmayacak).
- **AutoMapper** zorunludur.
- **Validation** zorunludur.

Response formatı standart olmalıdır:

```
{  
  "success": true,  
  "message": "",  
  "data": {}  
}
```

- **Pagination** tüm listeleme endpointlerinde uygulanmalıdır.
- **Search & Filter** desteği en az ürün, müşteri ve siparişlerde olmalıdır.
- **Authorization:** Admin ve Customer rolleri ayrılmalıdır.
- API proje yapısı **katmanlı** olmalıdır (Model + Service + Controller).

III. Angular (Frontend)

Angular tarafı bir **e-ticaret storefront** olacaktır.

1. Sayfalar (Geliştirilecek Modüller)

- **Anasayfa** + slider

- Kategoriye göre ürün listeleme
- Ürün detay
- Sepet yönetimi
- Üye kayıt & giriş
- Profil yönetimi
- Adres yönetimi
- Sipariş oluşturma (checkout)
- Sipariş geçmişi
- Ürün yorumu ekleme
- Hata sayfaları (404 / 500)

2. Güvenlik ve State Yönetimi

- JWT login sistemi
- Role Guard
- Auth Interceptor (token ekleme)

3. Zorunlu Yapılması Gerekenler

- Angular'da **Global Exception Interceptor** olmak zorundadır.
- Tüm API çağrıları **service** katmanı üzerinden yapılmalıdır.
- Lazy-loading module yapısı kullanılmalıdır.
- Angular guard yapısı zorunludur (AuthGuard).
- Custom validator kullanılmalıdır (mail kontrol vb.).
- Tüm sayfalarda loading spinner kullanılmalıdır.
- Pagination ürün ve siparişlerde zorunludur.
- Arama (search) zorunludur.
- Mobil uyumlu responsive tasarım zorunludur.

IV. Proje Teslim Şartları

Proje herkesin kendi GitHub hesaplarında yayınılayacak ve **eksiksiz README** dosyası hazırlayacaktır.

Repo içinde 3 ana klasör olmalıdır

1. Admin Panel
2. Rest Api
3. FrontEnd

README İçeriği (ZORUNLU)

1. Proje açıklaması
2. Kullanılan teknolojiler
3. Proje mimarisi (MVC + API + Angular diyagramı)
4. Kurulum adımları

5. Migration komutu
6. Demo admin ve kullanıcı bilgileri
7. API endpoint listesi
8. Ekran görüntüleri (MVC & Angular)
9. Postman Collection export