

## Ödev-1

**Konu:** "Müşteri (Customer) Yönetimi"

**Senaryo:** E-ticaret sisteminin olmazsa olmazı müşterilerdir. Ürünlerle henüz bir ilişkisi olmasa bile, müşteri verilerini tutacak bir yapıya ihtiyacımız var.

**Görev:**

- Entity:** Data/Models altına Customer sınıfı oluşturun (`Id, FirstName, LastName, Email, City`).
  - Migration:** Customer tablosunu veritabanına ekleyen migration'ı oluşturun ve update edin.
  - Repository:** ICustomerRepository ve CustomerRepository sınıflarını yazın.
  - Unit of Work:** IUnitOfWork içine Customers property'sini ekleyin ve constructor'da bağlayın.
  - DTOs:** CustomerDto (Listeleme için) ve CreateCustomerDto (Ekleme için) sınıflarını oluşturun.
  - Mapping:** AutoMapper profilinde gerekli dönüşümleri tanımlayın.
  - Service & Controller:** CustomerService ve CustomersController'ı yazarak uçtan uca (CRUD) çalışır hale getirin.
- 

## Ödev-2

**Konu:** "Ürün Kuralları (Business Rules)"

**Senaryo:** Şu an veritabanına "Laptop" isminde 10 tane ürün ekleyebiliyoruz. Bu veri kirliliğidir. Ayrıca fiyatı mantıksız derecede yüksek ürünler girilmemeli.

**Görev:**

- ProductService içindeki CreateProductAsync metodunu güncelleyin.
  - Kural 1 (Uniqueness):** Veritabanında aynı Name ile kayıtlı bir ürün varsa, kaydı engelleyin.
  - Kural 2 (Limit):** Bir ürünün fiyatı 1.000.000 TL'den fazla girilirse reddedin.
- 

## Ödev-3

**Konu:** "Veriyi Zenginleştirme (KDV Hesaplama)"

**Senaryo:** Veritabanında fiyatlar KDV'siz (Net) tutuluyor. Ancak API'den veri çeken mobil uygulama (Client), ekranda doğrudan KDV Dahil (%20) fiyatını görmek istiyor.

**Görev:**

- ProductDto sınıfına TaxIncludedPrice (KDV Dahil Fiyat) adında yeni bir property ekleyin.
- Veritabanında bu alan yok. Bu yüzden ProductService içinde veya AutoMapper konfigürasyonunda (`.ForMember` kullanarak) bu alanı hesaplayın.
  - Formül: Price \* 1.20*
- GET /api/products isteği atıldığında, veritabanındaki Price normal gelirken, TaxIncludedPrice alanının %20 fazlası olarak geldiğini kanıtlayın.