# E-Ticaret Projesi Dokümantasyonu

## Kurulum

## Bilgisayarınızda .NET 6.0 SDK olduğundan emin olun. Zipten çıkarın ve SLN’ye çift tıklayın. Daha sonra ETicaretUI’ya gelip sağ tıklayın. Açılan pencerede “set as Startup project’e tıklayın.

## metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

## Daha sonra Solution Explorer(Çözüm gezgini)’dan Data ->Context -> ETicaretContext.cs sınıfına gelip "Server=SQLSERVERADINIZ;Database=veritabanıadı;Trusted\_Connection=True;"); kısmını kendi server ve db adınıza göre değiştirin.

## Daha sonra Tools -> Nuget Package Manager -> Package Manger Console’a basarak çıkan terminale sırasıyla Add-migration initial, update-database yazıp çalıştırın

## metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

## Mimari Yapı ve Katmanlar

Proje, klasik katmanlı mimari yaklaşımını benimsemektedir ve aşağıdaki katmanlardan oluşur:

### 1. Sunum Katmanı (ETicaretUI)

MVC pattern üzerine kurulu sunum katmanı, kullanıcı arayüzünü ve controller'ları içerir.

### 2. Veri Erişim Katmanı (Dal - Data Access Layer)

Veri erişimi için repository pattern uygulanmıştır. İlgili interfaceler ve concrete implementasyonları bu katmanda yer alır. İlgili class’a göre gerekli metotlar eklenilebilir.

### 3. İş Mantığı Katmanı (Business)

İş kuralları ve servis uygulamalarını içerir. Generic Repository pattern'in çatısı bu katmanda yer alır.

### 4. Veri Katmanı (Data)

Entity modelleri, DbContext , Identity ve ViewModels sınıfları bu katmanda yer alır.

## Veritabanı Yapısı ve Entity Modelleri

Projede kullanılan temel entity'ler ve bunların özellikleri aşağıda detaylandırılmıştır. Bu entity'ler, veritabanında karşılık gelen tablolara karşılık gelmekte ve birbirleriyle ilişki içinde bulunmaktadır.

**Ürün (Product) Entity**

**Alanlar:**

* **ProductId**: Ürünün benzersiz kimliği
* **Name**: Ürün adı
* **Description**: Ürün açıklaması (opsiyonel)
* **Image**: Ürün görseli dosya yolu
* **Stock**: Stok miktarı
* **Price**: Ürün fiyatı
* **IsHome**: Anasayfada gösterilip gösterilmeyeceği
* **IsApproved**: Ürünün satışa onaylı olup olmadığı
* **IsActive**: Ürünün aktif/pasif durumu (soft delete için)
* **CategoryId**: Bağlı olduğu kategori ID'si

**İlişkiler:**

* Bir ürün sadece bir kategoriye aittir (Category entity'si ile one-to-many ilişki)
* Bir ürün birden fazla sepet öğesinde bulunabilir (CartItem entity'si ile one-to-many ilişki)
* Bir ürün birden fazla sipariş satırında bulunabilir (OrderLine entity'si ile one-to-many ilişki)

**Kategori (Category) Entity**

**Alanlar:**

* **Id**: Kategori benzersiz kimliği
* **CategoryName**: Kategori adı
* **Description**: Kategori açıklaması (opsiyonel)
* **IsActive**: Kategorinin aktif/pasif durumu (soft delete için)

**İlişkiler:**

* Bir kategori birden çok ürüne sahip olabilir (Product entity'si ile one-to-many ilişki)

**Sepet (Cart) Entity**

**Alanlar:**

* **Id**: Sepet benzersiz kimliği
* **UserId**: Kullanıcı ID'si
* **CreatedDate**: Sepet oluşturma tarihi

**İlişkiler:**

* Her sepet bir kullanıcıya aittir (AppUser entity'si ile many-to-one ilişki)
* Bir sepet birden çok sepet öğesi içerebilir (CartItem entity'si ile one-to-many ilişki)

**Sepet Öğesi (CartItem) Entity**

**Alanlar:**

* **Id**: Sepet öğesi benzersiz kimliği
* **CartId**: Bağlı olduğu sepet ID'si
* **ProductId**: Ürün ID'si
* **Quantity**: Miktar
* **DateAdded**: Eklenme tarihi

**İlişkiler:**

* Her sepet öğesi bir sepete aittir (Cart entity'si ile many-to-one ilişki)
* Her sepet öğesi bir ürüne aittir (Product entity'si ile many-to-one ilişki)

**Sipariş (Order) Entity**

**Alanlar:**

* **Id**: Sipariş benzersiz kimliği
* **UserId**: Kullanıcı ID'si
* **OrderNumber**: Sipariş numarası
* **OrderDate**: Sipariş tarihi
* **ShipAddress**: Teslimat adresi
* **ShipCity**: Teslimat şehri
* **IsCompleted**: Siparişin tamamlanma durumu

**İlişkiler:**

* Her sipariş bir kullanıcıya aittir (AppUser entity'si ile many-to-one ilişki)
* Bir sipariş birden çok sipariş satırı içerebilir (OrderLine entity'si ile one-to-many ilişki)

**Sipariş Satırı (OrderLine) Entity**

**Alanlar:**

* **Id**: Sipariş satırı benzersiz kimliği
* **OrderId**: Bağlı olduğu sipariş ID'si
* **ProductId**: Ürün ID'si
* **Quantity**: Miktar
* **Price**: Birim fiyat

**İlişkiler:**

* Her sipariş satırı bir siparişe aittir (Order entity'si ile many-to-one ilişki)
* Her sipariş satırı bir ürüne aittir (Product entity'si ile many-to-one ilişki)

**Adres (Address) Entity**

**Alanlar:**

* **Id**: Adres benzersiz kimliği
* **Title**: Adres başlığı (Ev, İş vb.)
* **FullAddress**: Tam adres metni
* **City**: Şehir
* **UserId**: Kullanıcı ID'si
* **IsDefault**: Varsayılan adres olup olmadığı
* **IsActive**: Adres aktif/pasif durumu (soft delete için)

**İlişkiler:**

* Her adres bir kullanıcıya aittir (AppUser entity'si ile many-to-one ilişki)

**AppUser (Kullanıcı) Entity**

**Alanlar:**

* **Id**: Kullanıcı benzersiz kimliği (Identity'den miras alınır)
* **UserName**: Kullanıcı adı (Identity'den miras alınır)
* **Email**: E-posta adresi (Identity'den miras alınır)
* **PasswordHash**: Şifre hash'i (Identity'den miras alınır)
* **FirstName**: Kullanıcı adı
* **LastName**: Kullanıcı soyadı

**İlişkiler:**

* Bir kullanıcı birden çok adrese sahip olabilir (Address entity'si ile one-to-many ilişki)
* Bir kullanıcı birden çok siparişe sahip olabilir (Order entity'si ile one-to-many ilişki)
* Bir kullanıcı bir sepete sahip olabilir (Cart entity'si ile one-to-one ilişki)

**AppRole (Rol) Entity**

**Alanlar:**

* **Id**: Rol benzersiz kimliği (Identity'den miras alınır)
* **Name**: Rol adı (Identity'den miras alınır)

**İlişkiler:**

* Bir rol birden çok kullanıcıya atanabilir (Identity yapısı tarafından yönetilir)

## Repository Pattern Uygulaması

Projede Generic Repository pattern uygulanmıştır. Bu, Entity Framework işlemlerini soyutlayarak, kod tekrarını azaltan ve bakımı kolaylaştıran bir yaklaşımdır. Genel CRUD işlemleri burada yapılır.

### Generic Repository Interface ve Implementasyonu

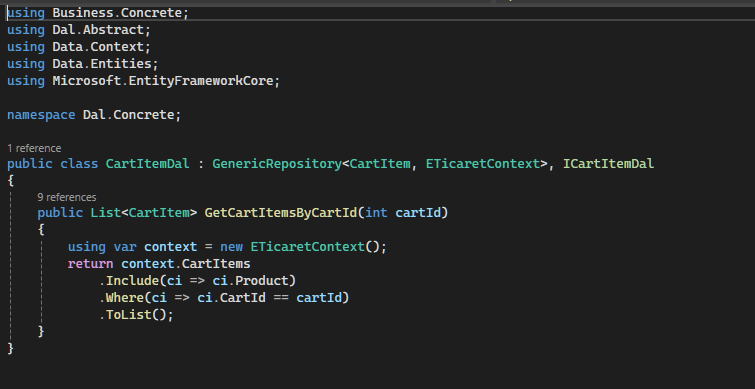
### metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

### Generic Repository Metotları ve İşlevleri

* **GetAll(Expression<Func<Tentity, bool>> filter = null)**:
  + Filtre parametresi opsiyonel olarak kabul ederek, tüm entity kayıtlarını liste halinde döndürür. Entity'de IsActive özelliği varsa, varsayılan olarak aktif kayıtları getirir ve ek filtreleme yapılabilir.
* **Get(int id)**:
  + Verilen ID'ye sahip tek bir entity kaydını döndürür. Entity'de IsActive özelliği varsa ve false ise null döner.
* **Get(Expression<Func<Tentity, bool>> filter)**:
  + Belirtilen ifadeye göre filtrelenmiş tek bir entity kaydını döndürür. IsActive özelliği varsa, önce aktif kayıtlar arasında arama yapar.
* **Add(Tentity tentity)**:
  + Yeni bir entity kaydını veritabanına ekler ve değişiklikleri kaydeder.
* **Update(Tentity tentity)**:
  + Mevcut bir entity kaydını veritabanında günceller ve değişiklikleri kaydeder.
* **Delete(Tentity tentity)**:
  + Belirtilen entity nesnesini soft-delete (IsActive = false) yaklaşımıyla pasif hale getirir. IsActive özelliği yoksa tamamen siler.
* **Delete(int id)**:
  + Belirtilen ID'ye sahip entity kaydını bulup soft-delete yaklaşımıyla pasif hale getirir. IsActive özelliği yoksa tamamen siler.

### Bu metotlar, tüm entity'ler için ortak CRUD (Create, Read, Update, Delete) işlemlerini sağlayarak, kod tekrarını önler ve veri erişim katmanının tutarlı bir şekilde uygulanmasını sağlar. Ayrıca, soft-delete(IsActive ile) yaklaşımı sayesinde verilerin fiziksel olarak silinmesi yerine pasif hale getirilmesi mümkün olur

### Entity-Specific Repository'ler

Her entity için özel repository sınıfları, GenericRepository'den türetilir ve gerekirse özel metodlar eklenebilir. Özel metotlarından biri aşağıda verilmiştir.   


Bu metot, belirli bir sepete ait tüm öğeleri, ilişkili ürün bilgileriyle birlikte (Include ile) getirir. Bu sayede bir sepete ait tüm ürünleri, ürün detayları ile birlikte listelemek mümkün olur. Kullanıcıya özel sepeti listeleme için kullanılmıştır.

metin, ekran görüntüsü, yazılım, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.  
Ekstra metot gerekmediğinde yukardaki gibi görünmektedir. Bu yaklaşım:

* Kod tekrarını önler
* Yazılımın bakımını kolaylaştırır
* Bağımlılıkları azaltır
* Gelecekte ihtiyaç olduğunda özel metotların eklenmesine esneklik sağlar
* Repository pattern'in tutarlı bir şekilde uygulanmasını garanti eder

## Soft Delete Mekanizması

Projede, kayıtları fiziksel olarak silmek yerine "Soft Delete" yaklaşımı kullanılmıştır. Bu yaklaşımda, kayıtların `IsActive` özelliği `false` olarak işaretlenir ve bu kayıtlar sorgulamalarda varsayılan olarak getirilmez.

metin, ekran görüntüsü, yazılım içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Bu yaklaşım, GenericRepository içinde dinamik olarak uygulanmıştır. Repository, ilgili entity'nin `IsActive` özelliği olup olmadığını kontrol eder ve buna göre silme işlemini gerçekleştirir.

## Identity Yönetimi

## AppUser sınıfı:metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

## AppUser sınıfı, standart Identity özelliklerine ek olarak isim, soyisim ve adres bilgilerini saklar.

## *Controller Sınıfları*

**1. HomeController**

**Sorumluluklar**: Ana sayfa ve ürün listeleme işlevleri

* **Index**: Ana sayfayı gösterir, öne çıkan ürünleri listeler
* **List**: Tüm ürünleri kategori, fiyat aralığı ve arama terimine göre filtreleyerek listeler
* **Details**: Belirli bir ürünün detay sayfasını gösterir
* **Search**: Ürün arama işlemini gerçekleştirir
* **Privacy**: Gizlilik politikası sayfasını gösterir
* **Error**: Hata sayfasını gösterir

**2. AccountController**

**Sorumluluklar**: Kullanıcı hesabı yönetimi, kimlik doğrulama işlemleri

* **Login**: Kullanıcı girişi yapar, anonim sepeti kullanıcı sepeti ile birleştirir
* **Register**: Yeni kullanıcı kaydı oluşturur
* **Logout**: Kullanıcı çıkışı yapar
* **Index**: Kullanıcı profil sayfasını gösterir
* **EditProfile**: Kullanıcı profilini düzenleme işlemlerini yönetir
* **ChangePassword**: Şifre değiştirme işlemlerini yönetir
* **GetOrders**: Kullanıcının siparişlerini listeler
* **AccessDenied**: Yetkisiz erişimlerde gösterilecek sayfayı yönetir

**3. CartController**

**Sorumluluklar**: Alışveriş sepeti ve sipariş işlemleri

* **Index**: Sepet içeriğini görüntüler
* **Buy**: Ürünü sepete ekler
* **Delete**: Sepetten ürün siler
* **RemoveAll**: Sepetteki tüm aynı ürünleri siler
* **Checkout**: Sipariş tamamlama sürecini yönetir
* **OrderCompleted**: Sipariş tamamlandı sayfasını gösterir
* **SaveOrder**: Siparişi veritabanına kaydeder

**4. ProductController**

**Sorumluluklar**: Ürün yönetimi (Admin)

* **Index**: Tüm ürünleri listeler
* **Create**: Yeni ürün ekleme formunu gösterir ve ekler
* **Edit**: Mevcut ürünü düzenleme formunu gösterir ve günceller
* **Delete**: Ürün silme işlemini gerçekleştirir
* **Details**: Ürün detaylarını gösterir

**5. CategoryController**

**Sorumluluklar**: Kategori yönetimi (Admin)

* **Index**: Tüm kategorileri listeler
* **Create**: Yeni kategori ekleme formunu gösterir ve ekler
* **Edit**: Mevcut kategoriyi düzenleme formunu gösterir ve günceller
* **Delete**: Kategori silme işlemini gerçekleştirir
* **Details**: Kategori detaylarını gösterir

**6. OrderController**

**Sorumluluklar**: Sipariş yönetimi (Admin)

* **Index**: Tüm siparişleri listeler
* **Details**: Sipariş detaylarını görüntüler
* **Edit**: Sipariş durumunu güncelleme işlemlerini yönetir
* **Delete**: Sipariş silme işlemini gerçekleştirir

**7. AddressController**

**Sorumluluklar**: Kullanıcı adres yönetimi

* **Index**: Kullanıcının adreslerini listeler
* **Create**: Yeni adres ekleme formunu gösterir ve ekler
* **Edit**: Mevcut adresi düzenleme formunu gösterir ve günceller
* **Delete**: Adres silme işlemini gerçekleştirir
* **SetDefault**: Bir adresi varsayılan olarak ayarlar

**8. UserController**

**Sorumluluklar**: Kullanıcı yönetimi (Admin)

* **Index**: Tüm kullanıcıları listeler
* **Create**: Yeni kullanıcı ekleme formunu gösterir ve ekler
* **Edit**: Mevcut kullanıcıyı düzenleme formunu gösterir ve günceller
* **Delete**: Kullanıcı silme işlemini gerçekleştirir
* **Details**: Kullanıcı detaylarını gösterir
* **UserRoles**: Kullanıcı rollerini yönetir

**9. RolesController**

**Sorumluluklar**: Rol yönetimi (Admin)

* **Index**: Tüm rolleri listeler
* **Create**: Yeni rol ekleme formunu gösterir ve ekler
* **Edit**: Mevcut rolü düzenleme formunu gösterir ve günceller
* **Delete**: Rol silme işlemini gerçekleştirir
* **RoleUsers**: Belirli bir role sahip kullanıcıları görüntüler

## *Projenin UI Kısmı*

## *Ana Sayfa*

## Kullanıcı giriş yaptıktan sonra gördüğü kısım. isHome ile Öne çıkan ürünler belirlenilir. metin, ekran görüntüsü, web sitesi, yazılım içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

## *Alışveriş sayfası*

## metin, ekran görüntüsü, web sitesi, yazılım içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.Kullanıcı bu sayfada ürünlere bakabilir. Kategoriye göre listeleyebilir. Fiyat aralığını filtreleyebilir. İsim ve fiyata göre artan ve azalan olarak sıralayabilir. Sayfa başına 6 ürün listelenir. Diğer ürünler için alttan diğer sayfalara geçebilir. Ürün arayabilir.

## metin, yazı tipi, sayı, numara, ekran görüntüsü içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.Giriş ve Kayıt Sayfaları

## Kullanıcı Sepeti

## metin, yazılım, web sayfası, bilgisayar simgesi içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.Kullanıcı yalnızca giriş yaptıktan sonra sepete ürün ekleyebilir. Kaldır butonu ile eklediği ürünü silebilir. +,- tuşlarıyla da ürünün alış miktarını belirleyebilir. Kullanıcı ürünün stok sayısından fazla sayıda sipariş veremez.

## metin, sayı, numara, yazılım, ekran görüntüsü içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.Siparişi tamamla butonuna bastığında alttaki gibi bir ekran gelir. Burada eğer varsayılan adres seçiliyse tüm adres bilgileri otomatik dolu olarak gelir.

## metin, sayı, numara, yazı tipi, yazılım içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

## *Kullanıcı eğer kayıtlı adreslerimden birini kullan butonunu işaretlerse bu şekilde kayıtlı adreslerini seçebilir.*

## metin, yazılım, web sayfası, web sitesi içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.*Profil Sayfası*

## Kullanıcı profil sayfasında verdiği siparişleri, onaylanıp onaylanmadığını görebilir. Son siparişler kullanıcının verdiği son 2 siparişi gösterir. Onayı admin verir. Veya isterse iptal edebilir.

## Admin Paneli

## Admin ürünleri listeleyip yeni ürün ekleyebilir. Pasife alabilir

## metin, ekran görüntüsü, sayı, numara, yazılım içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

## metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

## Düzenleye bastığında ana sayfada gösterebilir, aktif veya pasifliğini (soft delete), satışa uygunluğunu değiştirebilir. Ürünün stok ve fiyatını güncelleyebilir. Resmini değiştirebilir.

## Admin Kullanıcıların rollerini değiştirebilir. İsim ,soyisim, Numara gibi özelliklerini değiştirebilir. Evet gerçek dünya uygulamalarında olmaması gereken bir özellik. Belki ilerde düzenleyebilirim.

## metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.Admin yeni roller ekleyip silebilir. Eğer silinecek rolde kullanıcı bulunuyorsa uyarı verir.

## Admin siparişleri listeleyebilir. Ve ürünleri onaylayıp iptal edebilir.

## metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

## Kategori silme işlemi içinde hiçbir ürün yokasa gerçekleştirilebilir. Eğer ürün olan bir kategori silinmeye çalışırsa aşağıdaki gibi bir hata verilir.

## metin, ekran görüntüsü, yazı tipi içeren bir resim Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

## *Güvenlik Önlemleri*

### *CSRF Koruması*

ASP.NET Core'un varsayılan CSRF koruması uygulanmıştır. Form'larda `@Html.AntiForgeryToken ()` kullanılarak CSRF tokenlar eklenir ve controller'larda `[ValidateAntiForgeryToken]` attribute'u ile doğrulama yapılır.

Formun Üstü

### *Kimlik Doğrulama ve Yetkilendirme*

ASP.NET Core Identity kullanılarak kimlik doğrulama ve yetkilendirme işlemleri yapılmaktadır. Rol tabanlı yetkilendirme ile kullanıcıların erişim hakları kontrol edilir.   
[Authorize(Roles = "Admin")] ile kullanıcılardan rolu admin olanın erişebilmesi sağlanır.  
[Authorize]ise kullanıcı girişiyle sayfaya erişilebildiğini ifade eder.

### *Şifre Politikaları*

Identity'nin şifre politikaları aşağıdaki gibi özelleştirilmiştir. Her bir kodun yanında yorum satırı olarak ne yaptığı yazmaktadır.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.