

1. SecurePayAPI

1.1. Projeye nasıl başladım?

Projeye başlamadan önce proje içerisinde kullanılacak olan tablolar belirlendi. Relational Database kullanılacağı için tablolar arasındaki ilişkileri kod yazılmaya başlanmadan önce belirlendi. Daha sonrasında test caseleri yazıldı. Asıl projeden önce testler hazırlanarak projenin devamında hata olma olasılığı aza indirildi.

1.2. API tasarımı hakkında;

API içerisinde kullanıcılar (customer), ürünler (product) ve kullanıcıların ürünleri satın aldıklarının kaydını tutan payment kısmı bulunmaktadır.

Kullanıcılar email ve password ile kayıt olduktan sonra, kredi kartı bilgilerini girmesi gerekir. Kredi kartı numarası her kullanıcı için unique'tir. Bir kullanıcının kullandığı kredi kartını başka bir kullanıcı kullanamaz.

Payment metodu ile önce ilgili Customer'a ait ID, daha sonra satın alacağı ürünün ID'si doğru bir şekilde girilir. Kullanıcıya kayıtlı olan kredi kartı numarası da doğru girildikten sonra satın alım gerçekleşmiş olur. Aksi halde "Kredi kartı seçili kullanıcıya ait değil." hatası dönecektir.

Customer detay kısmında ID'si alınan kullanıcının email, ad, soyad ve satın alma geçmişi görüntülenir. Payment listesi kısmında ise tüm kullanıcıların ne zaman, hangi ürünü, ne kadar fiyata satın aldıkları listelenir.

1.3. Kullanılan paketler ve teknolojiler;

Proje içerisinde SQL sorgularını kod ile oluşturmak için Entity Framework Core paketi kullanılmıştır.

Kullanıcının API'daki tüm alanları görüntülememesi için view modeller oluşturulmuştur. Maplemeleri ise AutoMapper paketi ile gerçekleştirilmiştir.

Validasyonlar için Fluent Validations paketi kullanılmıştır.

Swagger UI kullanılmıştır.

Loglamalar yapılmış middleware ile dahil edilmiştir.

XUnit ile testleri yazılmıştır.