PROJE GENEL BİLGİLERİ:

* .Net 5.0 ile Rest Api projesi olarak geliştirildi (.Net Core 3.1 ile geliştirilecekti ancak .Net 5.0 kişisel merakımdan dolayı .Net 5.0 sürümünü denemek istedim),
* Çok katmanlı mimari kullanıldı,
* Swagger entegrasyonu gerçekleştirildi,
* Trafiğin takip edilebilmesi adına ILogger ile önemli görülen işlemlerin loglama işlemleri gerçekleştirildi,
* Jason Web Tokens kullanılarak kullanıcı yetkilendirmesi kuruldu,
* ConnectionString, Password gibi önemli bilgiler Crypto’lanarak kullanılmıştır,
* Local çalışırken ConnetionString şifresini Azure Key Vault üzerinde tutuyordum ancak test eden kişinin de bu konfigürasyonla bağlanabilmesi için bilgisayarına bir sertifika indirmesi gerekli olacaktı. Bu nedenle harici bir çalışma yapmamanız adına Azure Key Vault comment’lenerek devre dışı bırakılmıştır (Program.Cs),
* SOLID prensipleri temel alınmıştır,
* Uygulamayı Container olarak deploy edebilmek içini mage oluşturan gerekli DockerFile proje içinde mevcuttur,

ORTAMLAR:

* Development ve Testing olmak üzere 2 ortam kullanıldı,
* Integration Test projesi için Testing ortamı ve Testing Veritabanı kullanılacak şekilde ayarlandı,
* Her iki ortam için de Swagger kurulumu gerçekleştirildi,

VERİ TABANI:

* Geliştirilen çalışma mantığından dolayı NoSQL yerine relational bir veritabanı kullanıllması daha uygun görüldü, bu nedenle MSSQL üzerine kuruldu,
* EF Code First ile oluşturulan veritabanı üzerindeki güncelleme gereksinimleri Migration’lar kullanılarak geliştirildi,
* Test ve Production olmak üzere 2 farklı veritabanı kuruldu,
* ORM olarak EF kullanıldı,

DESIGN PATTERN:

* Unit Of Work,
* Repository,
* Dependency Injection,
* CQRS Design Pattern kullanma isteğim vardı ancak çok hakim olmadığım bir konu olması ve hızlı yetiştirme gereksinimimden dolayı bu pattern’ı uygulayamadım,

TEST:

* TDD programlama tekniği uygulandı (başta planladığım senaryolar değişim gösterdiği için TDD yönteminde kısmi değişiklikler gerçekleştirilmek zorunda kaldı)
* TDD için proje geliştirme çalışmalarına Unit Test’lerin geliştirilmesi ile başlandı,
* Unit Test’leri için MS Test Framework kullanıldı,
* Proje katmanları geliştirildikten sonra Integration Test geliştirmesi yapıldı,
* Integration Test için xUnit Framework kullanıldı,

ÇALIŞMA MANTIĞI:

* CreateToken, GetBasket ve AddToBasket olmak üzere toplam 3 tane Rest Api metodu bulunmaktadır,
* AccessController içinde CreateToken ile başlangıçta bir Token oluşturulması gerekmektedir, aksi durumda BasketController metotlarına ulaşılamayacaktır,
* Oluşturulan başarılı Token bilgisi ile BaseController üzerinde kullanıcı yetkilendirmesi yapılarak GetBasket ve AddToBasket metotlarına erişim sağlanabilecektir,
* GetBasket;
  + Kullanıcının Aktif bir sepeti varsa burada sepet bilgileri döndürülür,
  + Aktif mantığı, satın ama işleminden sonra Basket’in pasife çekilerek daha sonraki Sepet oluşturmalarında sepetteki ürünlerin tekrar getirilmemesi ve satış kaydının tutulması adına uygulanmıştır,
  + Eklenen ilk ürün için otomatik olarak Basket oluşturulmaktadır ancak bu metot aracılığı ile aktif bir basket yok ise oluşturma metodu tetiklenmemektedir,
  + TRY dışında EUR ve USD para birimli ürünler de bulunmaktadır. Başarılı GetBasket dönüşünde Merkez Bankası’nın 1 önceki gün kapanış kurları çekilerek Sepet fiyatı hesaplanmaktadır,
  + Sepet satın alma işlemine kadar döviz kurunda yaşanacak değişimden dolayı bu kurlar kaydedilmeyecek, sepet satın alma işleminden sonra iptal-iade işlemlerinde ödenecek tutarın kayıt altına alınmış olması için BasketProduct tablosuna satış kuru kaydedilecektir,
* AddToBasket;
  + Çiçek Sepeti alışveriş yapısı düşünülerek stoklara Şehir parametresi de bağlanmıştır. Kullanıcı bulunduğu şehirde stok bulunmamasından dolayı sitede aktif görünen ürünlerin sepete eklenememe şansı bulunmaktadır,
  + Min ve Max Order Quantity bilgileri tutulmaktadır, talep edilen ürün adedinin uygunluğu bu kısıtlamalara göre dönüş yapılmaktadır,
  + Basket’te aynı üründen varsa yeni bir kayıt açmak yerine mevcut BasketProduct kaydının adet bilgisi güncellenmektedir,
  + Başarılı stok sorgusundan sonra ürün kaydedilirken aktif bir Basket yok ise bu aşamada oluşturulmaktadır,
  + Ürünün sepete eklenmesi ile aynı bağlantı üzerinde stok azaltma işlemi de gerçekleştirilmektedir,
* Ürün sepete eklenirken sepette başka şehir ürünü var mı diye kontrol etmeyi gözden kaçırmışım, daha fazla bekletmemek adına bu hata ile beraber göndermek durumunda kalındı,
* Test senaryoları için eklenen veritabanı kayıtları aşağıdaki tablodadır;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ProductId | CityId | Quantity | Min. Or. Q. | Max. O. Q. | CurrencyId |
| 1 | 1 | 10000 | 1 | 100 | 1 (TRY) |
| 1 | 2-3-4-5 | 10000 | 1 | 100 | 1 (TRY) |
| 2 | 1-2-3-4-5 | 10000 | 5 | 10 | 1 (TRY) |
| 3 | 1-2-3-4-5 | 10000 | 10 | 50 | 2 (USD) |
| 4 | 1-2-3-4-5 |  |  |  | 3 (EUR) |
| 5 | 1-2-3-4-5 | 500 | 1 | 10 | 1 (TRY) |
| 6 | - | - | - | - | 1 (TRY) |

|  |  |  |
| --- | --- | --- |
| Kullanıcı Adı | Şifre | Sepet Durumu |
| SeedUser | password | Yok |
| SeedUser1 | password1 | Yok |
| SeedUser2 | password2 | Yok |
| SeedUser3 | password3 | Yok |
| SeedUser4 | password4 | Yok |
| SeedUser5 | password5 | Yok |