<“Ömer Faruk Akyapak” >

**BACKEND :**

Backend, kullanıcıların görmediği arka yüz (Server-side) geliştiren kişidir. Yani sistemin mimarisini oluşturan, veri tabanı yönetimini planlayan, sunucu ayarlamalarından sorumlu, sistemin maksimum seviyede verimli ve hızlı çalışmasını sağlayan kişidir.

Backend programcıları, sunucu tarafında uygulamanın iş mantığını, veri tabanını ve diğer önemli işlevleri geliştirir. Kullanıcı arayüzü ile doğrudan etkileşime girmezler, ancak uygulamanın temel işlevselliğini ve veri yönetimini sağlarlar.

Backend programcılarının sorumlulukları şunlar olabilir:

* Veritabanı Tasarımı ve Yönetimi: Verileri depolamak ve yönetmek için veri tabanı yapısını tasarlar ve optimize eder.
* İş Mantığı Geliştirme: Kullanıcının yaptığı eylemlere dayalı olarak uygulamanın nasıl çalışacağını ve nasıl tepki vereceğini belirleyen iş mantığı kodlarını yazma.
* API Geliştirme: Frontend ile veri alışverişi yapabilmek için API'ları (Application Programming Interface) tasarlar ve geliştirir.
* Güvenlik ve Kimlik Doğrulama: Kullanıcı kimlik doğrulama, yetkilendirme ve veri güvenliği önlemlerini uygular.
* Sunucu Yönetimi: Uygulamanın sunucularının konfigürasyonunu, güvenliğini ve performansını yönetir.
* Hata Yönetimi ve Güncellemeler: Uygulamanın hatalarını takip eder, hata düzeltmeleri ve güncellemeleri yapar.
* Veri İşleme ve Analizi: Toplanan verileri analiz ederek anlamlı bilgiler üretir.

**Backend Mimarisi**

Solution “project”

-Bussiness : Yazılım iş kurallarının bulunduğu katmandır. Proje içerisinde gerçekleştirilecek işlem, fonksiyonların bulunduğu kaymandır.

-Core : Proje bazlı değişmeyen, her projeye uyarlanabilir tool, extension gibi nesnelerin ve DataAccess ve Entities katmanlarına bağlı sabit olan framework ve concreteleri barındıran katmandır.

-DataAccess : Projede bussiness ve entity katmanı arasında veri akışını sağlayan katmandır.

-Entities : Projede kullanılacak nesnelerin tutulduğu katmandır.

-WebAPI : Projenin veri tabanı üzerinde işlemleri gerçekleştiren katmandır. Bu işlemler sonucunda oluşan state değişikliklerinin görüntülenmesi ve crud işlemlerinin yapılabildiği katmandır.

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Monolitik Mimari

Bu öğrendiklerimizden sonra artık bir yazılım üretmeye başlayabiliriz. Monolitik mimari bir yazılımın tüm bileşenlerinin tek bir büyük ve bütünleşik bir yapıda olduğu tasarım yaklaşımını ifade eder. Bu noktada küçük start-up projeleri, basit ve ölçeklenmesi gerekmeyen projeler, parçalanmaya ihtiyaç duyulmayacak kadar az işlev barındıran projeler için motolitik mimari idealdir.

Uygulamamızın kullanıcı(client) tarafından gelen istekleri yönlendirdiği kısım API

İşlemlerin ve methodların gerçekleştirildiği Business Layer

Oluşturacağımız nesnelerin özelliklerini yazdığımız alan(field) kısımları olan “entity”ler ve bu entitylerin veritabanı ile ilişkilerinin olduğu “repository”lerin barındığı Data Layer

Son olarak da veritabanımız olan Database bizim katmanlarımız oluyor.