

FİNANSAL BT ÖDEVİ

MİMARİ TASARIM DÖKÜMANI

Uygulamamıza karar verirken back-end kısımda en mantıklısı Firebase kullanmak gözüktüyordu. Sadece minimalist duvar kağıtları için bir API bulamadık. Bu yüzden en mantıklısı duvar kağıtlarını kendimiz bulmamız ve mobil uygulamamızda göstermekti. Bizde duvar kağıtlarını bulduktan sonra Firebase'e yüklemeye karar verdik. Firebase'in bulut yapısı hem ödevde istediğiniz maddelere uygundu hem de mobil uygulamalar için çok uygun bir yapıydı. Firebase'de proje oluşturduk ve Storage kısmına resimleri yükledikten sonra Cloud Firestore kullanarak, duvar kağıtlarımıza gerekli id ve kategori bilgilerini girdik.

Firestore'de uygulama oluşturduktan sonra Google'un kendisinin otomatik olarak sunduğu ve indirmemiz gereken Google-service.json dosyası ise yapılan mobil uygulamaya Firestore'in özelliklerinin yüklenmesini oldukça kolaylaştırıyor ve uzun uğraşlardan kurtarıyordu. Biz uygulamamızı Flutter üzerinde geliştirdik. Flutter kullandığı Dart dili ve Firestore ile çalışmamızı oldukça kolaylaştıran kütüphaneleri ile bizim tercihimiz oldu. Ayrıca Flutter'ın arayüzde sağladığı esneklikte bu tercihimizi güçlendiren başka bir nedendi. Bu sayede aslında sadece duvar kağıtları olmasını düşündüğümüz uygulamada farklı kategoriler ve favoriler gibi kısımlar ekleme imkanımız oldu. Hatta Flutter kütüphaneleri sayesinde Firestore dışında bir başka bulut platformu daha da kullanmış olduk. Uygulamamızda Favoriler kısmı olsa güzel olur diye düşündük ama Firestore'de kurduğumuz veritabanında favoriler için bir anahtar değer kullanmamıştık. Hepsini tekrardan işlemek oldukça uzun vakit alacaktı. Bizde Flutter'ın Hive kütüphanesi kullanarak favoriler kısmını farklı bir NoSQL veritabanında saklamayı başardık. Firestore API'leri kullanmak içinse cloud_firestore ve firebase_core kütüphanelerini kullandık. Duvar kağıtlarını istediğimiz ölçülerde gösterebilmek ve telefonun arka planında ayarlamaya sağlayabilmek içinse wallpaper_manager ve photo_view gibi kütüphaneler kullandık. Son olarak ise tüm resimlerin tek bir sayfada gösterilmesi konusunda her tıkladığında resimler tekrardan yüklenmesin ve o uzun bekleme süreleri sıklıkla yaşanmasın diye cache hafıza üzerinde çalışmamızı sağlayan cache_network_image ve flutter_cache_manager kullandık.