****

**GÖMÜLÜ SİSTEM ÇÖZÜMLEMELERİ**

**Dr. Öğr. Üyesi Fırat AYDEMİR**

**Mobil Film-Dizi Tanıtma Uygulaması**

**2022-2023 Güz Dönemi**

**16/11/2022**

**İsmail KEYVAN – 201913171030**

**Muhammed Emir GÖZCÜ – 201913171053**

**Giriş**

Yaptığımız araştırmalar ve örnek incelemelerinden sonra film ve dizilerin bilgilerini bize getirecek olan API hakkında daha fazla bilgi topladık. Uygulamamız içerisinde film ve dizilerin genel bilgileri yanında afiş resimleri ve (eğer mevcut ise) fragman videolarını koymayı da planladık. Oluşturacağımız detay sayfasına film ve dizilerin genel puanları yanında genel bir açıklama kısmı da eklemek istiyoruz. Ayrıca uygulamamıza kullanıcı giriş sistemi de eklemek istiyoruz. Bu kısımda Firebase Authentication ve Firebase Realtime Database kullanmayı planlıyoruz. Kullanıcı girişi sistemini ekledikten sonra her kullanıcı beğendiği diziyi beğenebilecek ve daha sonra izle listesine ekleyebilecek. Bu bilgiler veri tabanında tutulacağı için kullanıcı uygulamaya tekrar giriş yaptığında bu listeye erişebilecek.

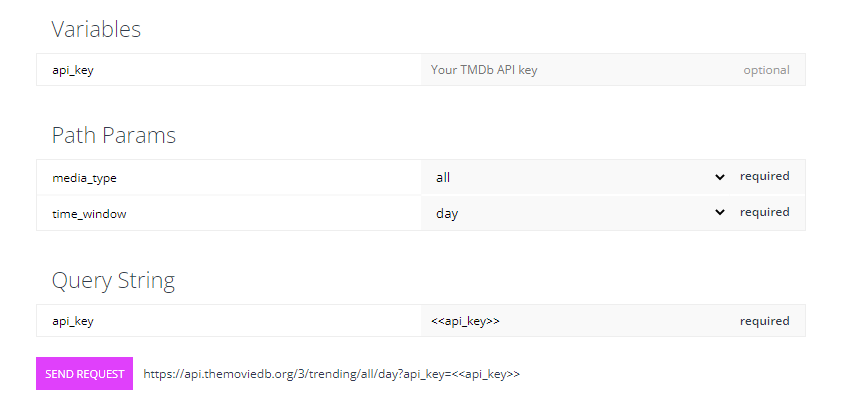
**Yapılan Çalışmalar**

Uygulamamızda gerekli film ve dizi bilgilerinin kullanıcıya gösterilebilmesi için bu bilgileri bize sağlayacak bir kaynak gereklidir. Bu kaynağı kendimiz oluşturabileceğimiz gibi bize bu bilgileri sağlayan daha önceden hazırlanmış hazır kaynaklardan da edinebiliriz. Biz The Movie Database (TMDB) web sitesinin sağlamış olduğu API ile film ve dizi içeriklerine ulaşmayı planlıyoruz.

**TMDB API:** İstekleri yapabilmek için bir API anahtarı gerektiren ve herkesin kullanımına açık bir API. Kullanıcıları ile film, TV şovu veya oyuncu görüntülerini ve/veya verilerini kullanmak için 3. Taraf verilerini paylaşır.

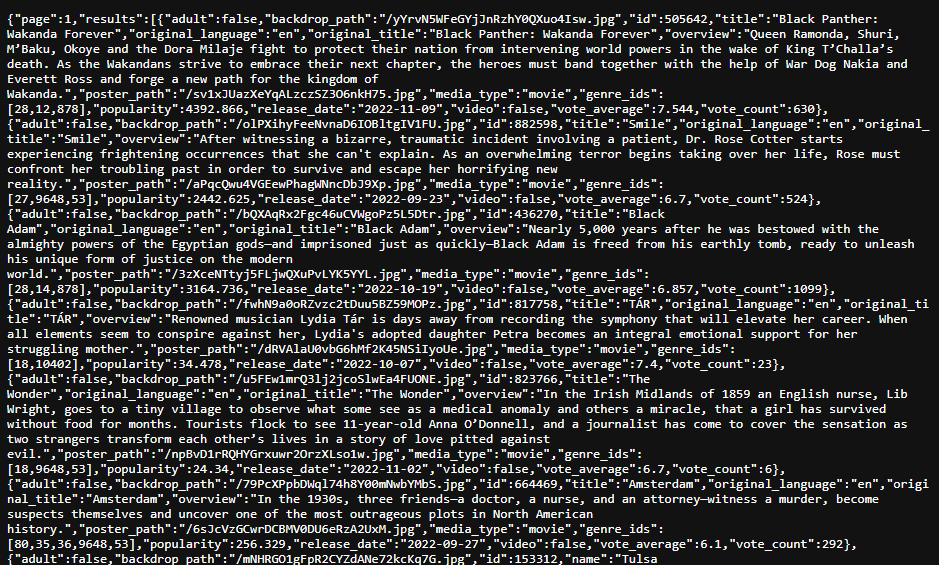
API’ın bize sağladığı hizmetlerden faydalanmak için bir API isteği atmamız ve cevabında dönen bilgileri kullanmamız gerekmektedir. API isteği, bir geliştirici bir URL'ye uç nokta eklediğinde ve sunucuya bir çağrı yaptığında gerçekleşir. Bir API uç noktası, bir API ile başka bir sistem arasındaki etkileşimin temas noktalarını ifade eder. Bir uç nokta, bir API'ın ihtiyaç duyduğu kaynaklara eriştiği konumu sağlar. API'lerin bir istek göndermek ve belirli bir kaynağın nerede bulunduğunu belirtmek için kullandığı iletişim kanalı, uç nokta olarak bilinir. Kaynaklara tam olarak nereden erişilebileceğini tanımlamada bütünleyici bir rol oynar ve onunla etkileşime giren herhangi bir yazılımın düzgün çalışmasını garanti eder.

API anahtarını aldıktan sonra atacağımız isteğe göre bize bir JSON dosyası dönecektir. Örnek olarak trendler için bir istek atmamız gerekirse aşağıdaki gibi bir istek atmamız gerekecektir. TMDP API dokümantasyonlarını inceleyerek bize atmış olduğumuz istek ile dönecek olan cevap hakkında bilgiler vermiş olduğu bilgiler ile daha anlaşılabilir bir hale getirmek mümkündür.



Bu istek bizden 3 adet parametre istemektedir. İlk olarak media\_type parametresi all, media, tv veya person parametrelerini alabilir. Biz all parametresini kullanarak tüm türlerden trend içerikleri uygulamamıza ekleyebiliriz. Sonraki parametre olan time\_window ise day veya week parametrelerini alıyor. Günlük veya haftalık trend içerikleri incelemek için bu parametre ile oynanabilir. Son olarak ise almış olduğumuz API anahtarını istek URL’imize ekleyerek bir istekte bulunduğumuzda bize ayarladığımız parametrelere göre gerekli veriler bir JSON dosyası formatında bize dönecektir.

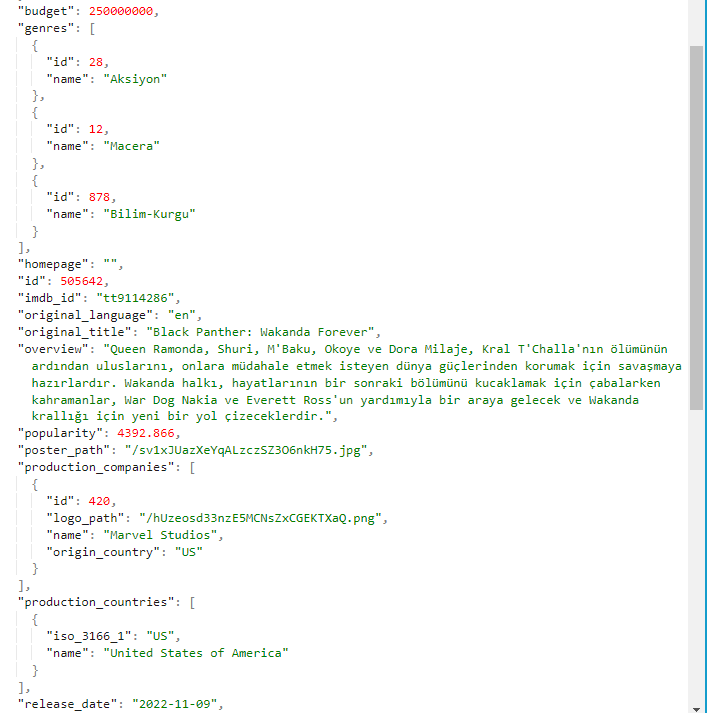
Örnek olarak trend içerikler için dönen JSON dosyası şu şekilde olacaktır:



Bu JSON dosyasını okumak zor olacaktır. Bu yüzden jsonbeutifier web sitesini kullanarak biraz da okunaklı bir hale getirirsek:

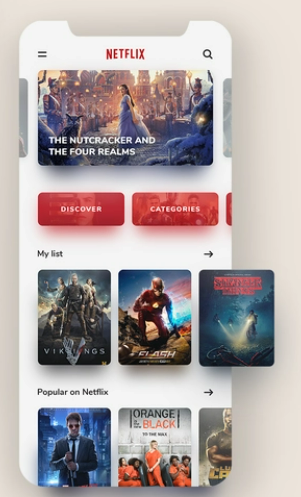


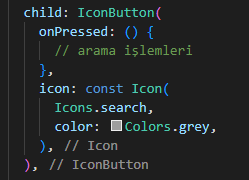
Bu cevap ile dönen JSON dosyası içerisinde trendlerdeki içerikler en trend olandan başlayarak sıralanmış bir şekilde gelir. Genel olarak dönen bilgilere bakarsak içeriğin afiş resminin, başlığının, orijinal dilinin, kısa bir açıklamasının, medya tipinin, tür ID’lerinin ve diğer farklı bilgilerin dönmüş olduğunu görürüz. Her içeriğin kendisine has bir ID’si bulunmaktadır. Bu ID ile ilgili içerik için daha detaylı bilgileri erişmemizi sağlayacak bir istekte bulunabiliriz. Üsteki dönmüş olan Black Panther: Wakanda Forever filmi için filmin ID’sini kullanarak başka bir istek atarsak:



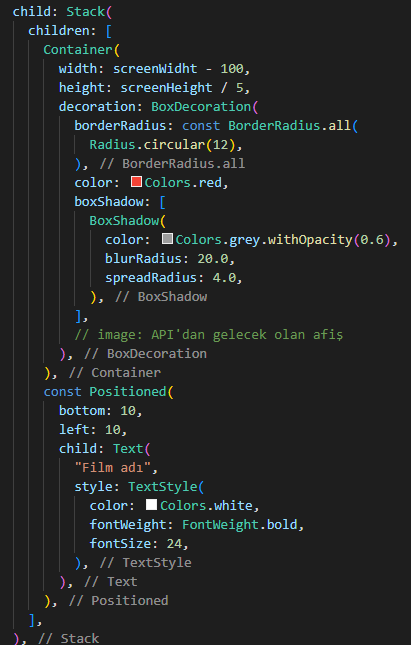
Cevabını bize döner. İstek URL’imizin son kısmına ek olarak “&language=tr-TR4” anahtar kelimesini ekleyerek dönen cevabın hangi dilde olacağını da kendimiz seçebiliriz.

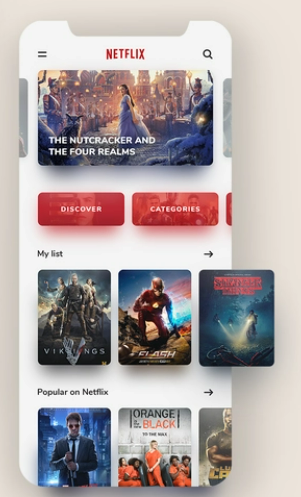
**Arayüz Tasarımı:** Genel olarak film veya dizi uygulamalarına baktığımızda çoğunlukla koyu temalarda hazırlanmış ve karışık bir arayüzü olan uygulamalar görüyoruz. Yapacağımız arayüz tasarımın göze hitap eden ve anlaşılması kolay bir arayüz olmasını planlıyoruz. Yaptığımız araştırmalardan sonra kullanacağımız arayüz tasarımında esinleneceğimiz tasarımı kararlaştırdık. Ayrıca genel olarak kullanacağımız widget’ları da belirlemeye çalıştık. Yapmayı planladığımız ana sayfa tasarımı ve kullanılacak bazı widgtelar:

Kullanıcının genel içerikler arasından istediği içeriğin ismini yazarak tüm içerikler

Arasından arama yapmasını sağlayan bir buton.

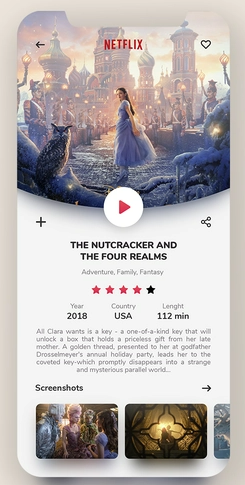
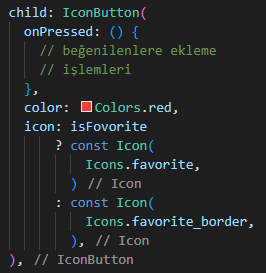
Kendi logomuzun geleceği kısım. Diğer ikonlar arasına ortalanmış olması daha güzel bir görüntü ortaya çıkarıyor.



Popüler içeriklerin geleceği slider yapısının bir elemanı. Yeni ve popüler içerikler bu kısımda gösterilecek ve sağa veya sola kaydırılarak diğer içeriğe geçiş yapılabilecek.

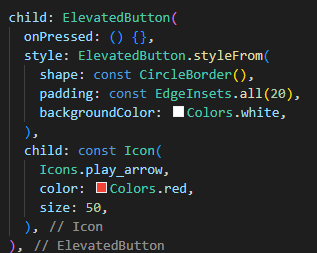
Üstteki widget ile benzer bir yapıya sahip kategoriler kartları. Aynı şekilde kaydırılarak kategoriler arasında geçiş yapılabilecek ve tıklanınca ilgili kategoride içeriklerin listelendiği sayfaya yönlendirme yapılacak.

Daha sonra izle listesine eklenen içeriklerin gösterileceği kısım. Popüler içeriklerin gösterileceği kaydırılabilir kısım. Tıklanılan içeriğe göre ilgili içerik için hazırlanmış detay sayfasına yönlendirme yapılacak.



İçeriğin favorilere eklenmesini ve favorilerden çıkarılmasını sağlayacak olan buton. Eğer içerik zaten favorilere eklendiyse tamamen kırmızı bir renk alacak.

İçerik hakkındaki detay bilgileri gösterecek olan kısım. API’dan gelen bilgileri bu kısımdaki yerlerine ekleyip kullanıcıya içerik hakkında genel bir bilgi verilecek.

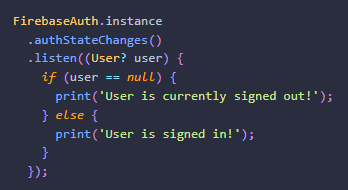


Eğer içerik için bir fragman var ise bu fragmanın oynatılmasını sağlayacak oynatma butonu. API’dan gelen veriler içerisinde bazı içeriklerin fragman videoları bulunmuyor. Bu durumda bu buton gösterilmeyecek.

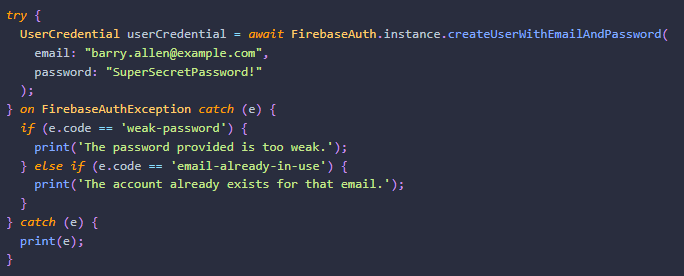
**Firebase Authentication:** Kullanıcıların uygulama içerisinde oturum açabilmesini ve kendilerine özel olarak oluşturulmuş dinamik bir kullanıcı deneyimi yaşaması için gerekli olan sistemi Firebase Authentication sağlar. Firebase Authentication, uygulamalarda kullanıcıların kimliğini doğrulamak için kullanımı kolay SDK'lar ve hazır kullanıcı arabirimi kitaplıkları sağlar. Şifre, telefon numarası, Google, Facebook ve Twitter gibi sağlayıcıları kullanarak kimlik doğrulaması yapmamızı sağlar.

Uygulamamızda FirebaseUI'ı bir kimlik doğrulama çözümü olarak kullanmak ve daha fazla oturum açma yöntemini entegre etmek için Firebase Authentication SDK'yı kullanmalıyız. Bu sayede uygulamamızda kullanıcıların oturumu açabiliriz.

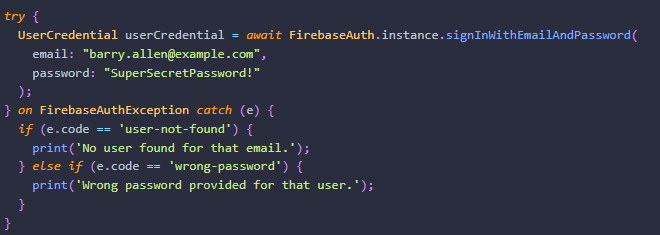
Uygulamamızda Firebase Authentication kullanmak için öncelikler firebase\_auth paketini uygulamamıza eklemek gerekiyor. Bu işlemden sonra aşağıdaki kod parçaları ile gerekli kontroller ve işlemler yapılabiliyor.

Kullanıcının kayıt durumunu incelemek için yandaki kod satırları kullanılır. Eğer kullanıcı uygulamada oturum açmamış ise oturum açma sayfasına yönlendirme yapılır.

E-posta ve şifre ile kullanıcı kaydı yapmak için kullanılacak kod parçası:



E-posta ve şifre ile kullanıcı girişi yapmak için kullanılacak kod parçası:

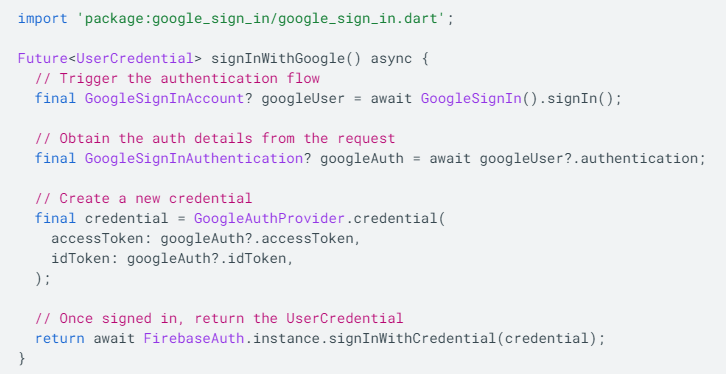


Kullanıcı e-posta doğrulaması yapmak için kullanıcıya bir e-posta gönderilmesi gerekiyor. Kullanıcı bu e-posta içerindeki URL’e tıklayarak hesap doğrulamasını gerçekleştirmiş oluyor. Kullanıcı e-posta doğrulaması yapmak için gerekli olan kod parçası:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Google ile kullanıcı girişi yapmak için kullanılacak olan kod parçası:



**Sonuç**: API kullanarak belirli URL dizilimleri ile uygulamamızda kullanacağımız olan film ve dizi içeriklerine ulaşabiliyoruz. Bu içeriklere ulaşmamız için kullanacağımız bir API anahtarı aldık ve birkaç istek atarak dönen cevapları inceledik. Cevaplar içerisinde uygulama içerisinde kullanacağımız film afişi, fragmanı ve diğer bilgilere ulaşabiliyoruz. Uygulama içerisinde bu bilgileri ilgili widget’lar içerisinde kullanacağız. Uygulama için farklı arayüz tasarımlarını inceledik ve karmaşıklıktan uzak olan bir tasarımın kullanıcılara daha verimli bir deneyim sağlayacağını gördük. Genel olarak takip edeceğimiz ve kendi fikirlerimiz çerçevesinde düzenleyeceğimiz bir tasarım kararlaştırdık. Bu tasarımın parçalarını oluşturmak için kullanacağımız widget’ların bazılarını hazırladık. API üzerinden gelecek olan verileri bu widget’lar içerisinde kullanıcıya göstereceğiz. Ayrıca kullanıcı giriş sistemi için kullanılacak alan Firebase Authentication sistemi kullanımını araştırdık. Kullanıcı kaydolma şekline farklı seçenekler kullanılabiliyor. Kullanıcı girişi durumu canlı olarak takip edilebiliyor ve bu durumlara ilgili sayfalara yönlendirme yapılabiliyor.

**Yapılacak Olan Çalışmala**r**:** Bir sonraki rapor döneminde yapmak istediğimiz uygulamamızın kodlamasına başlamak ve ana arayüz tasarımlarını oluşturmak. Kullanıcı kayıt ve giriş sistemini oluşturmak ve bunun için gerekli arayüzü hazırlamak. Kullanıcı kayıt sisteminin sorunsuz olarak çalıştığından emin olduktan sonra arayüz tasarımına başlamak istiyoruz.

**Kaynakça**

[1]<https://www.themoviedb.org/documentation/api>

[2] <https://jsonbeautifier.org/>

[3] <https://dribbble.com/shots/5026483-Netflix-Mobile-App-Redesign>

[4] <https://developers.themoviedb.org/3/movies/get-movie-details>

[5] <https://firebase.google.com/docs/auth/android/google-signin>

[6] <https://firebase.flutter.dev/docs/auth/usage/>

[7] <https://firebase.google.com/docs/auth/flutter/federated-auth#google>