

**Bölüm Adı: Elektrik ve Otomasyon**

**Program Adı: Mobil Teknolojileri**

**Ders Adı: Bulut Bilişim**

**Proje Konusu: Personel İşlemleri Uygulaması**

**19MY93004**

**Bilal Mercan**

İçindekiler

[Uygulama Senaryosu 3](#_Toc61825487)

[Uygulamada Kullanılan Componentler 4](#_Toc61825488)

[Props 5](#_Toc61825489)

[State 6](#_Toc61825490)

[Fetch API 7](#_Toc61825491)

[FlatList 8](#_Toc61825492)

[Firebase 9](#_Toc61825493)

[Firebase kurulumu 9](#_Toc61825494)

[Firebase Authentication (Giriş ve Kayıt) 11](#_Toc61825495)

[Firebase CRUD İşlemleri 12](#_Toc61825496)

[Veri Gönderme 12](#_Toc61825497)

[Veri Silme ve Güncelleme 12](#_Toc61825498)

# Uygulama Senaryosu

Uygulama Firebase ile personel bilgilerini veritabanına kaydeder. Personel listesini görebilir ve personel silme işlemlerini yapılması amaçlanmıştır.

Api fetch ile jsonplaceholder sitesindeki kullanıcı bilgilerini ekranda kullanıcı adına dokunarak kullanıcı adı, soyadı, telefon numarasını, web sitesi ve mailini görüntülenmesi amaçlanmıştır.

# Uygulamada Kullanılan Componentler

Prop (Componentler arası veri alışverişini sağlayan yapıdır.)

State (Kotlin kullanılan değiştirilebilir değişken :“var” ile aynı işlemleri yapar.)

Fetch API (Sunucudan veri alınması için kullanılan yapıdır.)

Async Storage (Veri Tabanı işlemleri)

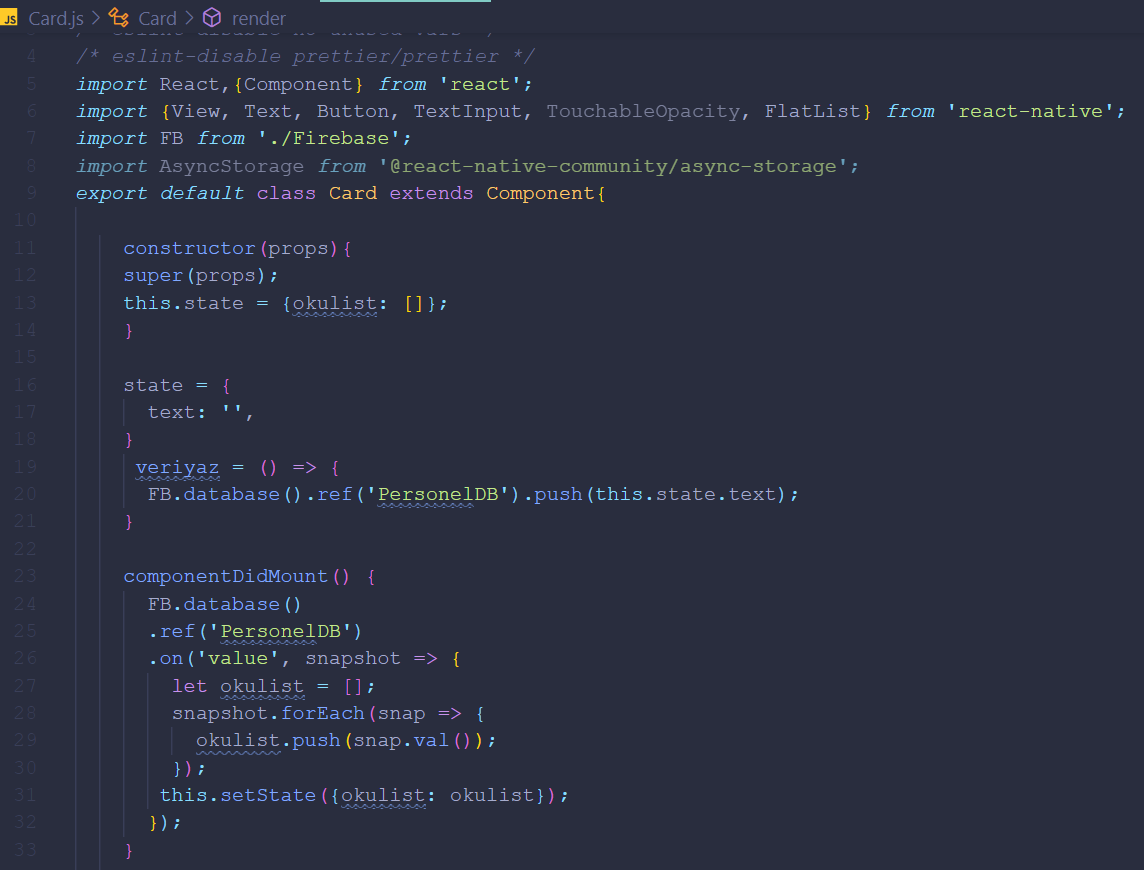
Firebase Authentication (Kimlik doğrulama / giriş)

Firebase Realtime Database (gerçek zamanlı veri tabanı)

React Native tab navigation (ekranda bulunan butona basarak ekranlar arası geçişi sağlar)

# Props

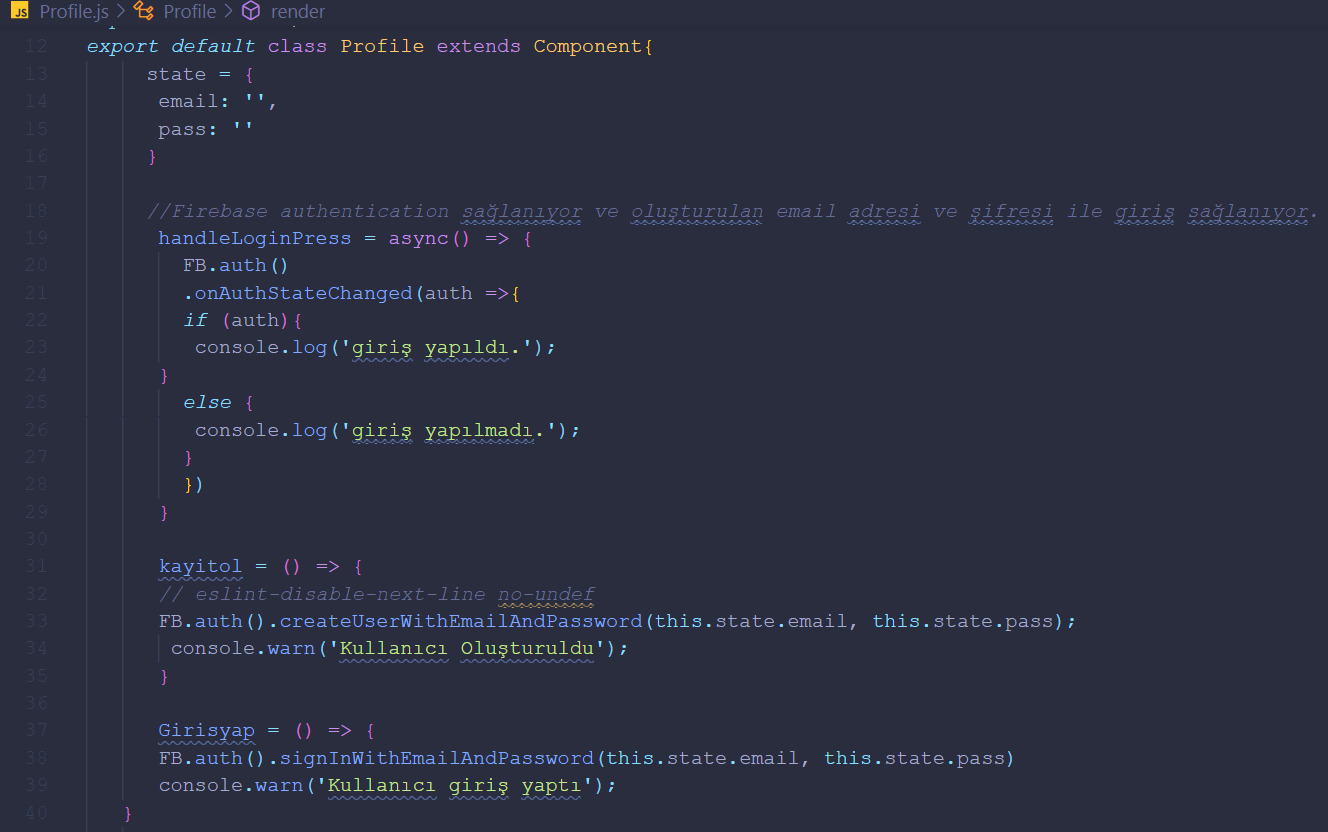
Ekranda görülen kurucu metod ile oluşturulan props ile okulist adında boş bir diziyi tanımlanıyor. Sonrasında componentDidMount içerisinde önceden oluşturulmuş okulist dizisi tekrardan tanımlanıyor. setState ile dizi başka bir componentler arası geçiş sağlanmış oluyor.

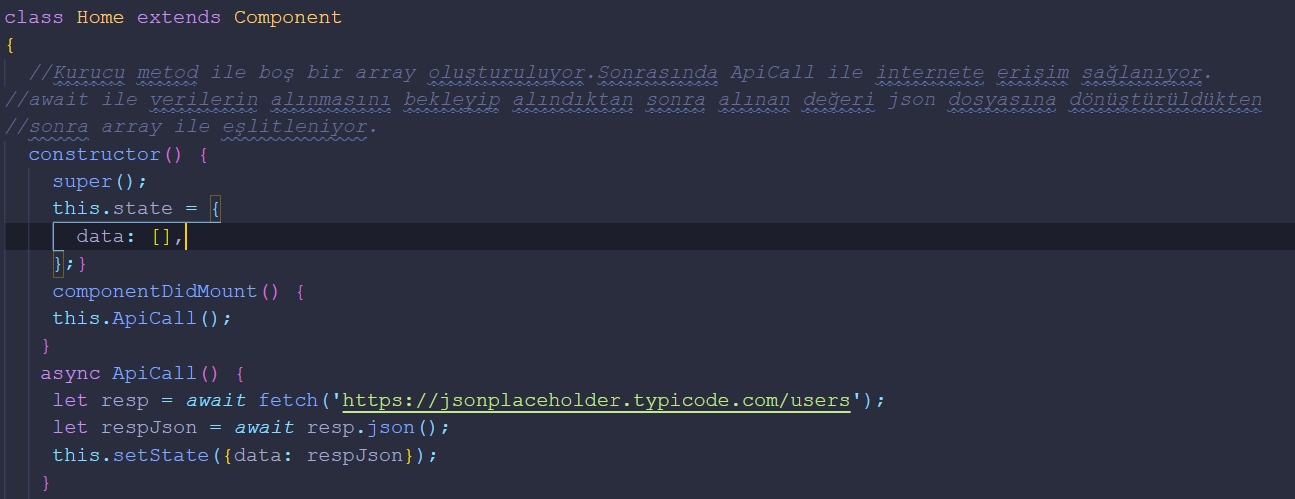


# State

Kotlin, java gibi dillerde de yer alan val, var, let gibi kullanılan değişken ile ayın işlevi görmektedir. Bu uygulamada gönderilen verilerin tutulduğu diziler bu işlevi görüyor.

Profile dosyasında email ve pass adında değişkenler oluşturulmuş ve bu değişkenler ile Firebase’ e gönderilen veriler bu değişkenlerde tutulmaktadır.

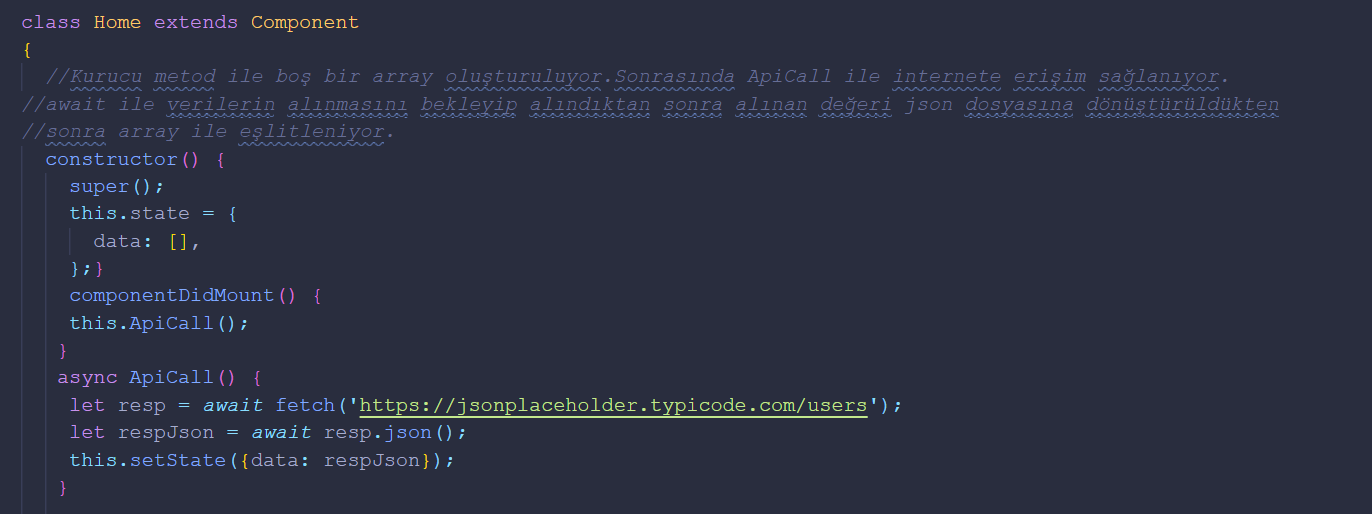


Home sayfasında ise siteden alınan verilerin data adında ki değişkende tutulması sağlanmıştır.

# Fetch API

Uygulamalarda internet erişiminin sağlanması veya internetten veri çekme, yükleme gibi işlemlerde kullanılması gereken özellik.

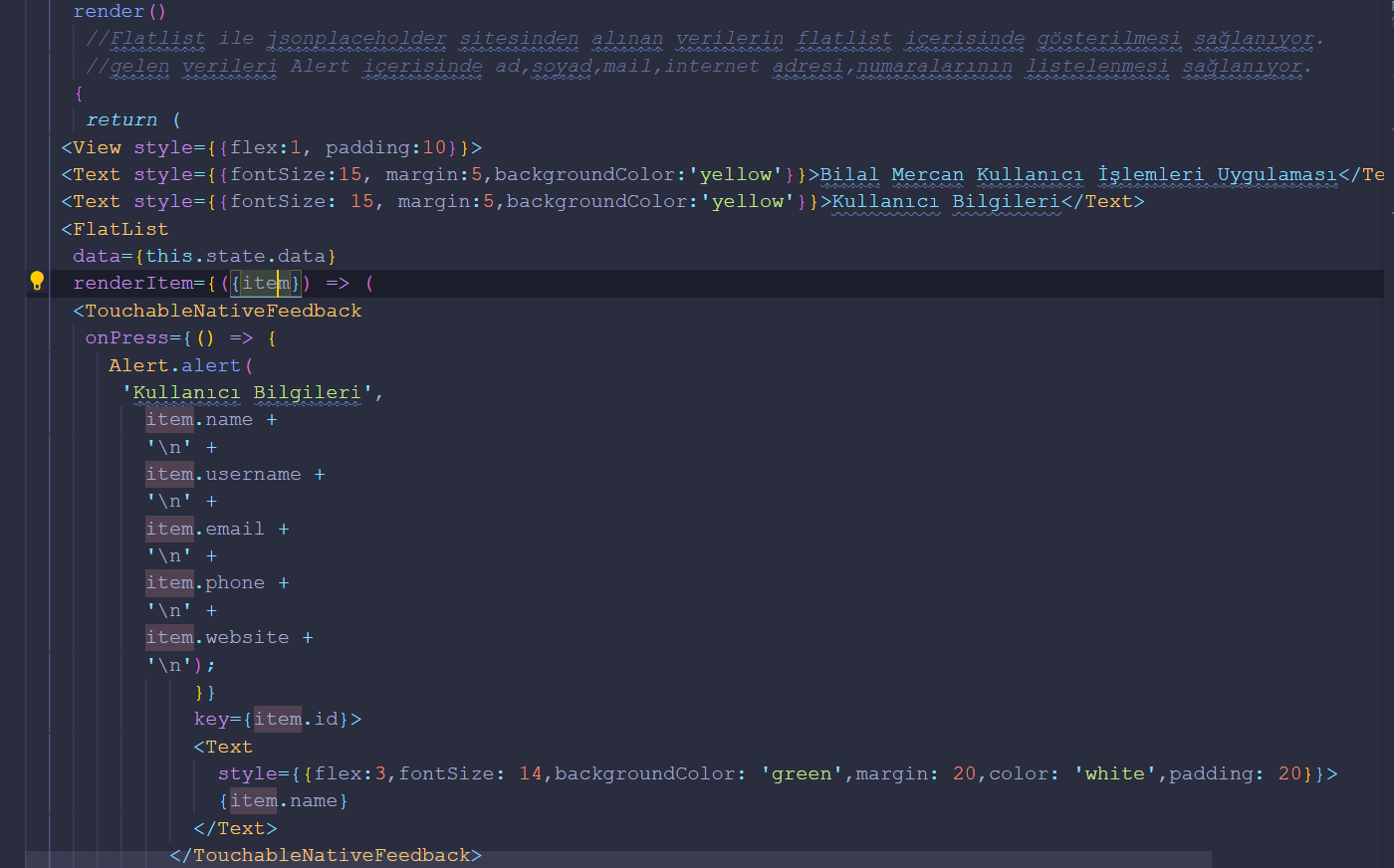
ComponentDidMount ile render metodundan sonra çalışması sağlanır. Yani ilk önce ekran render edilir sonrasında Apicall metodu çalışır.



# FlatList

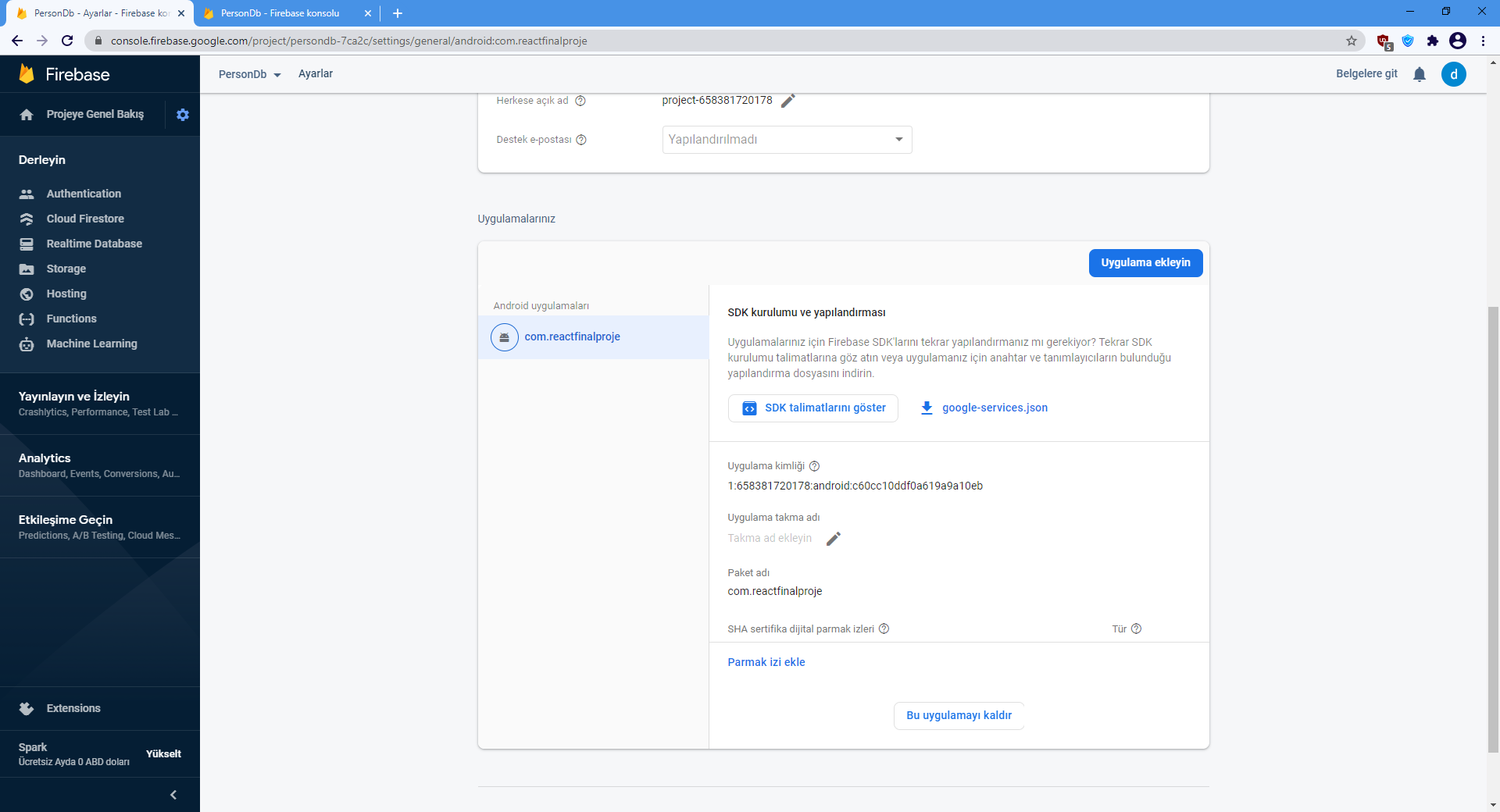
Flatlist ile jsonplaceholder sitesinden alınan verilerin liste görünümünde gözükmesi sağlanır. Alert dialog ile butona yazan butona basıldığında kişi hakkında çeşitli bilgiler verilmektedir.

Daha önceden data dizisinden alınan verileri flatlist içerisinde renderİtem ile tanımlanan item ile çekilen istenilen verilerin alınması sağlanıyor. TouchableNativeFeedback, alert dialog penceresi olarak geliyor ve adı, kullanıcı adı, email, telefon numarası ve web adresini veriyor. Tıklanıldığında verilen bilgilerin doğru olması için anahtar veri olarak id seçiliyor.



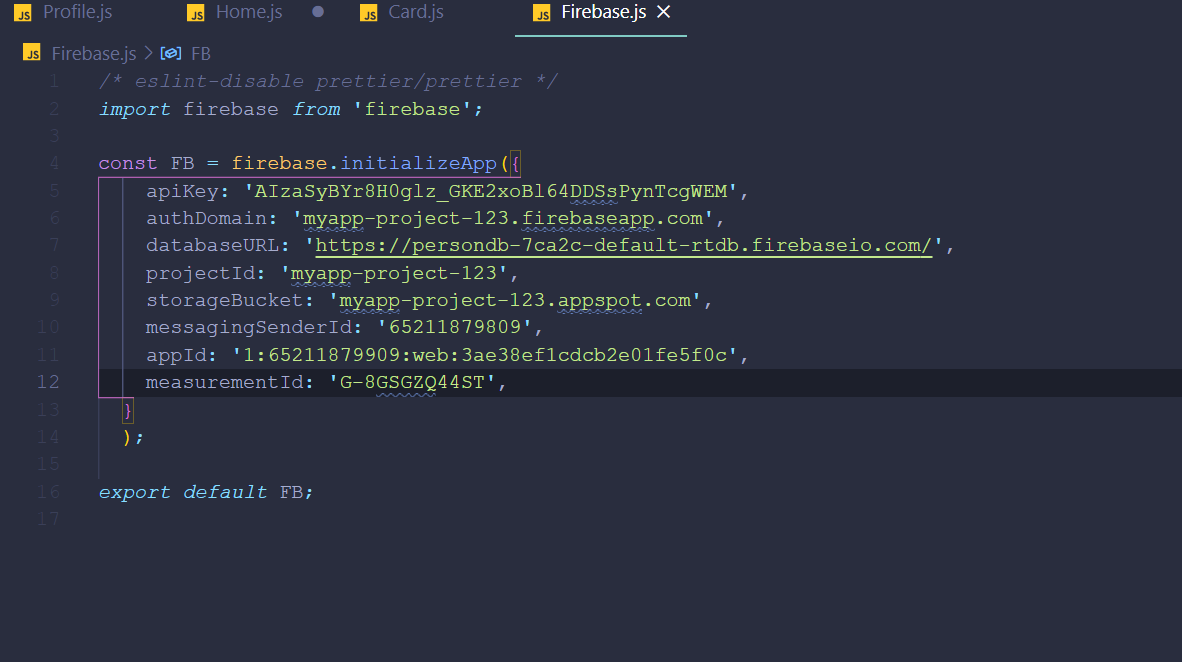
# Firebase

## Firebase kurulumu

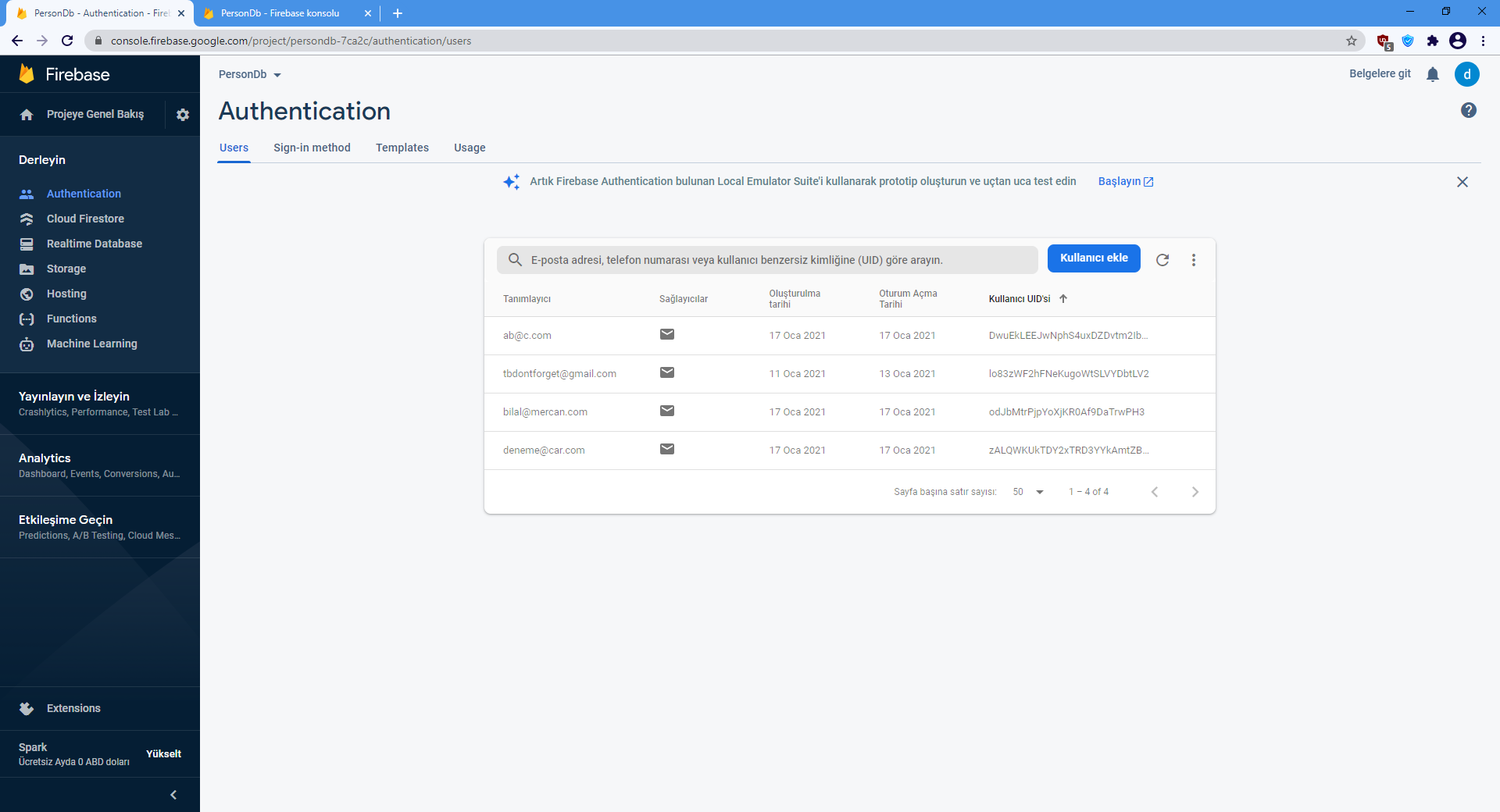
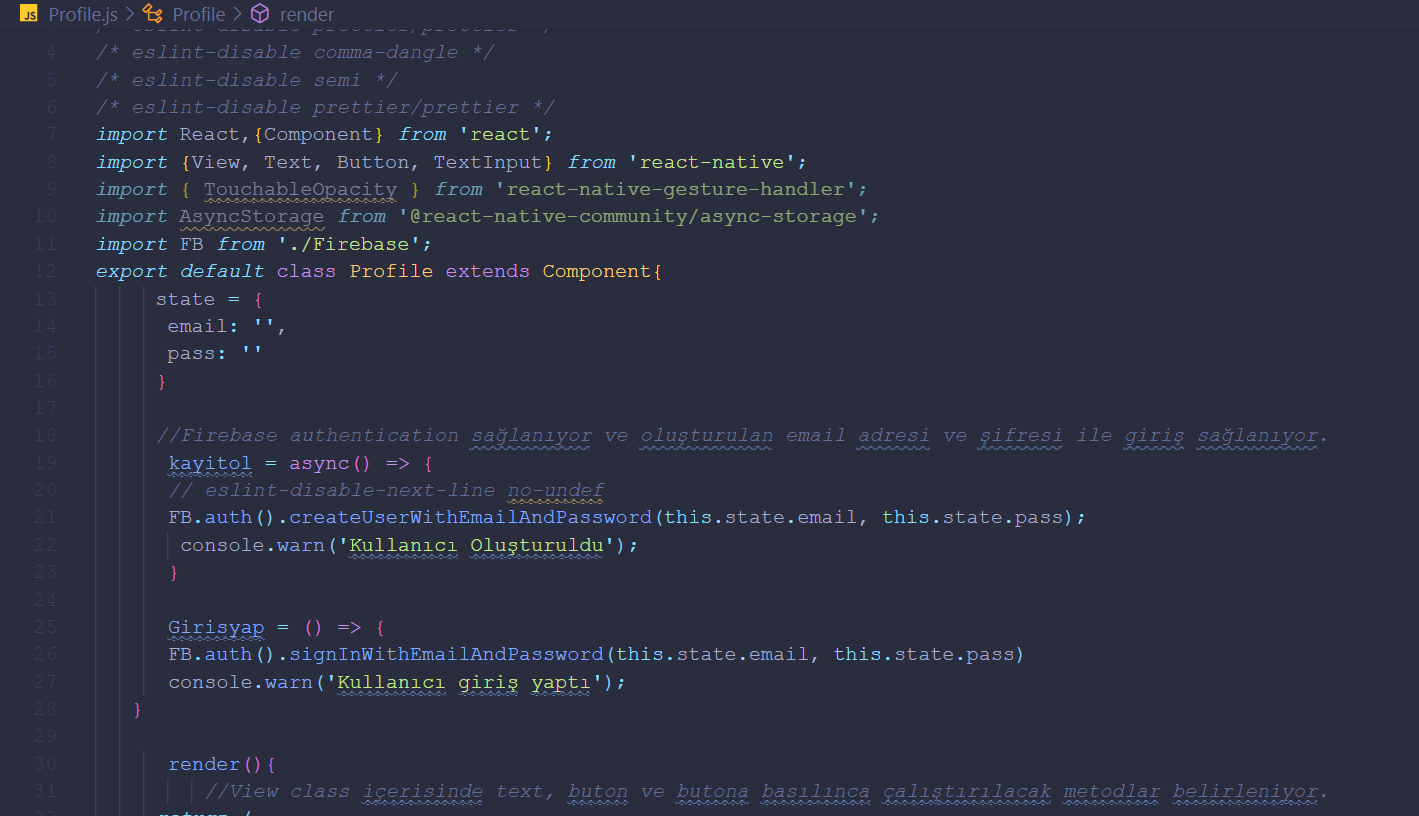


Firebase sitesinden gerekli işlemler yapıldıktan sonra firebase’ i uygulamaya aktarmak gerekiyor.

Firebase adında dosya oluşturuluyor ve firebase kütüphanesi ekleniyor. Sonrasında firebase diğer sayfalarda kullanılacağı için değişkene atanıyor ve initializApp komutu ile başlatılıyor. Firebase sitesinde proje ayarları sekmesinde uygulama için verilen api key, databaseUrl ve proje id’ si düzenleniyor ve dahil ediliyor. Bu bilgilerin yanlış olması halinde firebase işlemleri yapılamaz ve hata verir.

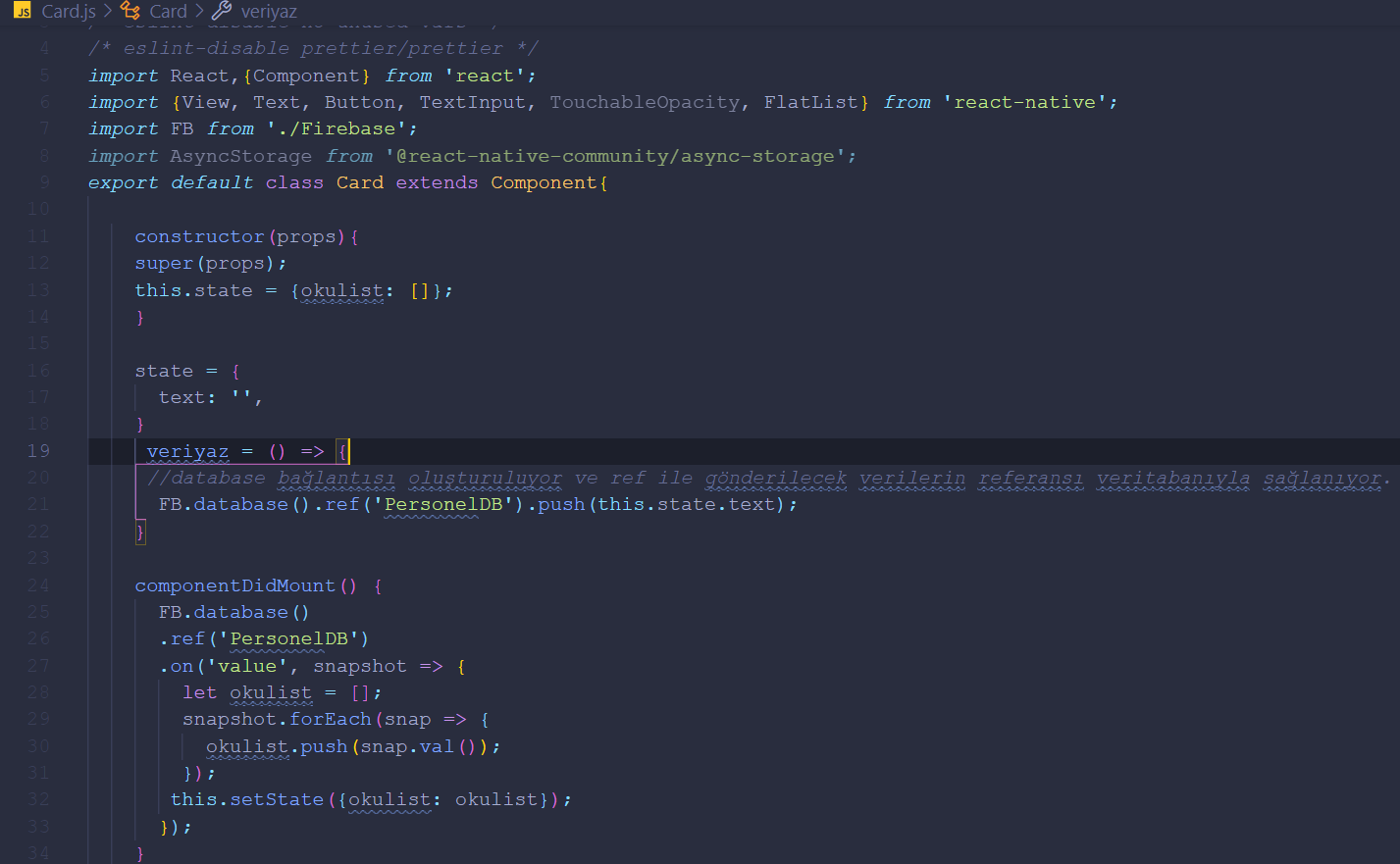


# Firebase Authentication (Giriş ve Kayıt)

Firebase sitesinden authentication bölümünden mail ile girişi aktifleştirdikten sonra giriş yapılabilir. Oluşturulan email ve pass değişkenlerine textinput içerisinden girilen değerler ile yeni kullanıcı oluşturuluyor. Giriş yap butonuna daha önceden kayıt olunan email ve password ile giriş yapılıyor.

# Firebase CRUD İşlemleri

## Veri Gönderme

Text değişkeni ile alınan veriler database’ e push ile gönderiliyor. Gönderilen verileri çağırmak için database içindeki değerleri okulist e aktarılıyor sonrasında setState ile gelen veriyi karşılıyor böylece gönderilen veri gönderildikten sonra görülebiliyor.

## Veri Silme ve Güncelleme

Database’ e giriş yaptıktan sonra tine referans adı ile .remove() komutu ile bütün verilerin silinmesi sağlanır. Güncellemek için girilen veri değiştirildikten sonra veri güncelle butonuna basılarak güncellenmesi sağlanır.

