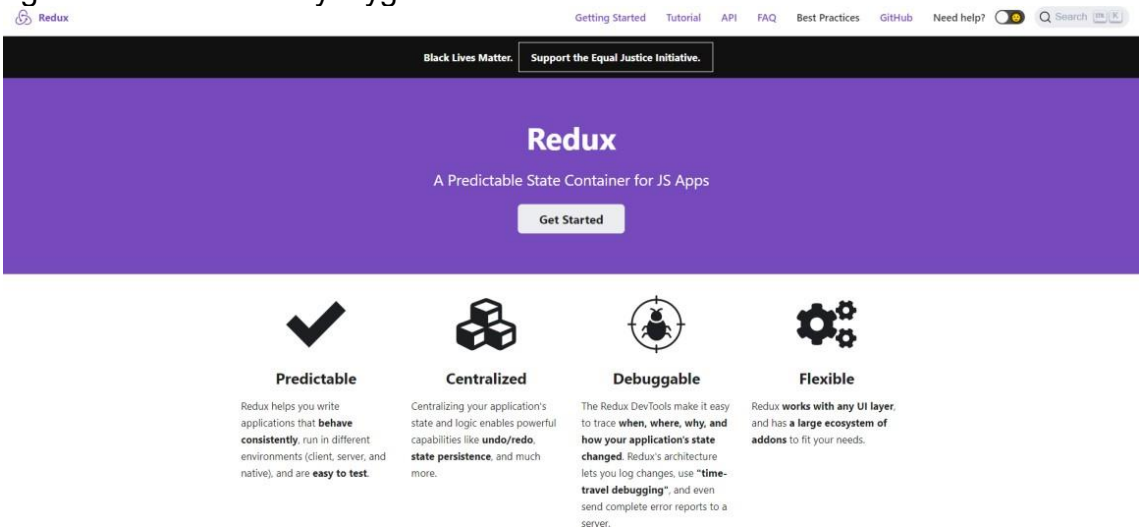
Burak Kalınkaya 171602011

Redux

Redux, React.js'de durum yönetimi için kullanılır. Varsayılan durum yönetimi, yüksek oranda ölçeklenebilir uygulamalar için yeterince iyi değildir. Redux'u daha basit hale getirmek için kullanabilirsiniz. Redux ve React-Redux, her ikisi de farklı kütüphanelerdir. Redux herhangi bir Javascript uygulamasıyla kullanılabilir. React Redux kütüphanesi, redux üzerinden bir bağlanmadır. Bu kütüphane hem React.js hem de React Native ile kullanılabilir. React Redux bir npm paketi olarak mevcuttur ve mevcut bir projeye kolayca entegre edilebilir. Redux mevcut bir projeye eklenebilir, ancak bunu proje geliştirmenin başlangıcında eklemek en iyi uygulamadır.

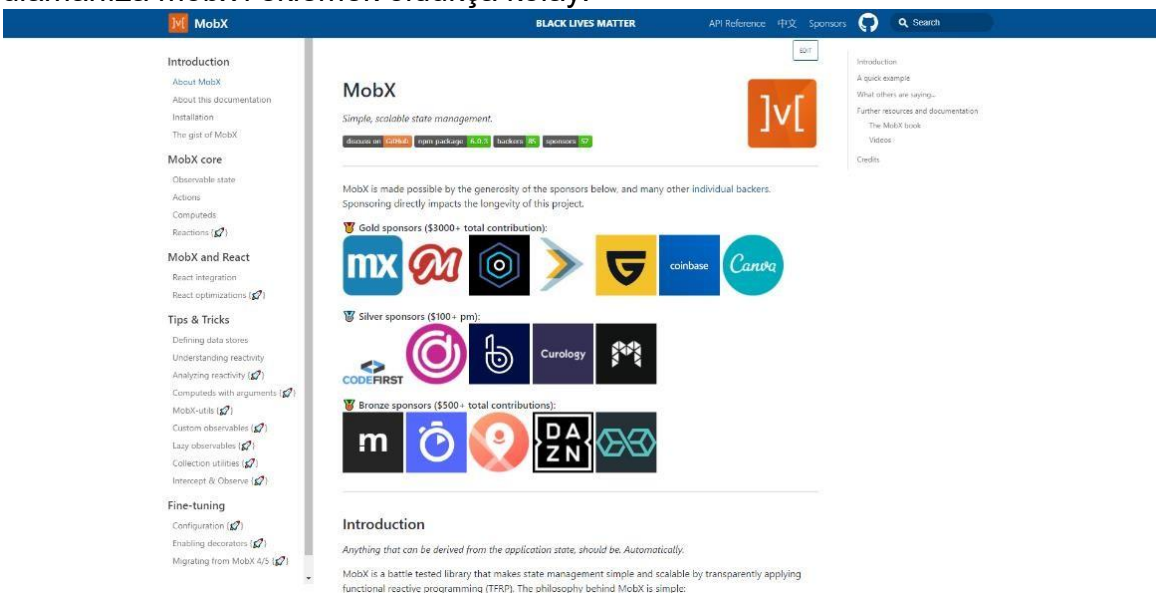


MobX

Mobx oluşturmak istediğiniz uygulamalar üzerinde veri yönetimini kolaylaştıran bir kütüphane. SPA(Single Page Application) oluşturma konusunda karşılaşılan mühim sorunlardan biri de veri(state) yönetimi ve uygulamanın yaşam döngüsü boyunca barındırdığı verilerin geçirdiği değişiklikleri tespit etmek.

Dolayısıyla, birçok "component" barındıran bir uygulamanızın çeşitli aşamalarında söz konusu verileri değiştirmek, tekrardan kullanılabilir hale getirmek çok önemli. Çünkü yapınız ve uğraştığınız veri tipi büyüdükçe, iş içinden çıkılmaz bir hale gelebiliyor.

Mobx'in burada avantajlı olduğu konular, özellikle güçlü bir kütüphane olan Redux'a göre, "daha basit ve kullanışlı olması", "hafif bir çözüm olarak" tanımlanabilmesi diyebiliriz. Özellikle varolan
uygulamanıza mobx'i eklemek oldukça kolay.

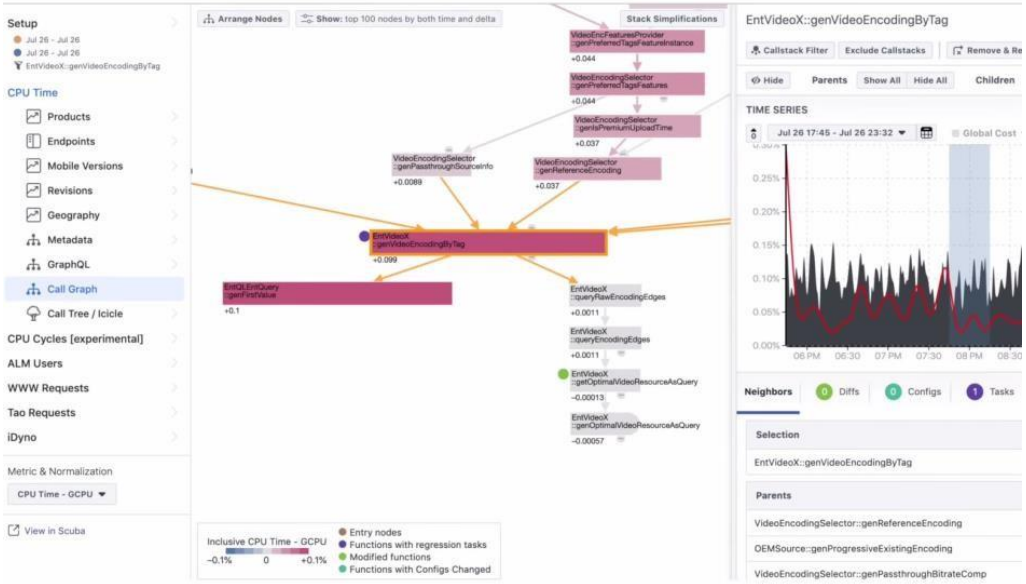


Burak Kalınkaya 171602011

Recoil

Ortaya çıkışı

David McCabe, Douglas Armstrong ve Christian Santos tarafından üretildi. Dave'in aktardığı bilgiye göre RecoilJS, Facebook'ta Comparison View adlı veri analizi uygulaması için yapılmıştı.

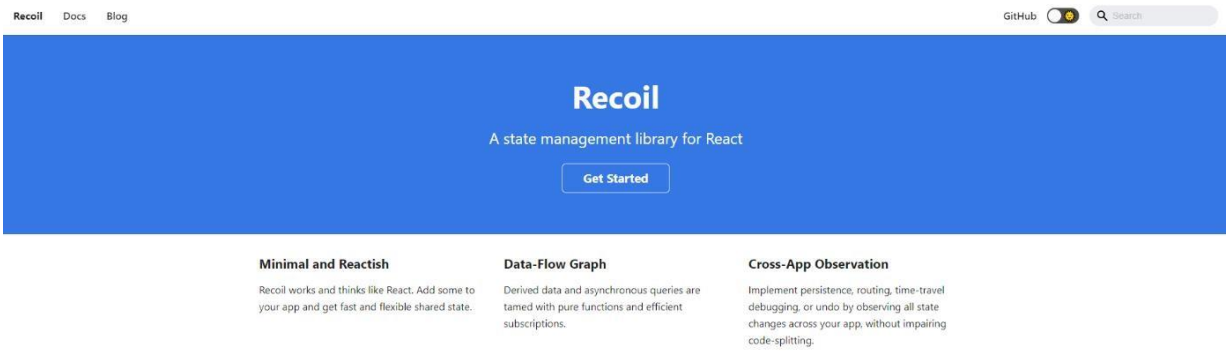


Bu araç istemci ve sunucu tarafındaki performans bazında tıkanıklıkların analiz edilmesi için görsel bir arayüz sunuyordu. Bu uygulamada o kadar fazla grafik ve etkileşimli bileşen vardı ki, Context API,
Redux ve diğer mevcut state kütüphaneleri iş göremez hale geliyordu. Bu kütüphaneler, ya istedikleri kadar esnek değildi, ya aradıkları performansı sunmuyordu, ya da mevcut state kütüphanelerinden birini kullandıklarında kodlar hataya meyilli hale geliyordu.

Bu nedenle state yönetimini uygulama içerisinde farklı bir şekilde gerçekleştirdiler ve yaptıkları
çözümü projeden ayırıp bir kod kütüphanesi (RecoilJS) haline getirdiler. Bu kütüphane, genel olarak karşılaşılan 3 problemin çözümünü üzerinde durmaktadır:

1. State'in esnek bir şekilde paylaşılması
2. Verinin türetilmesi (derived state) ve sorgulanması
3. Uygulama genelinde state'in izlenmesi

Burak Kalınkaya 171602011



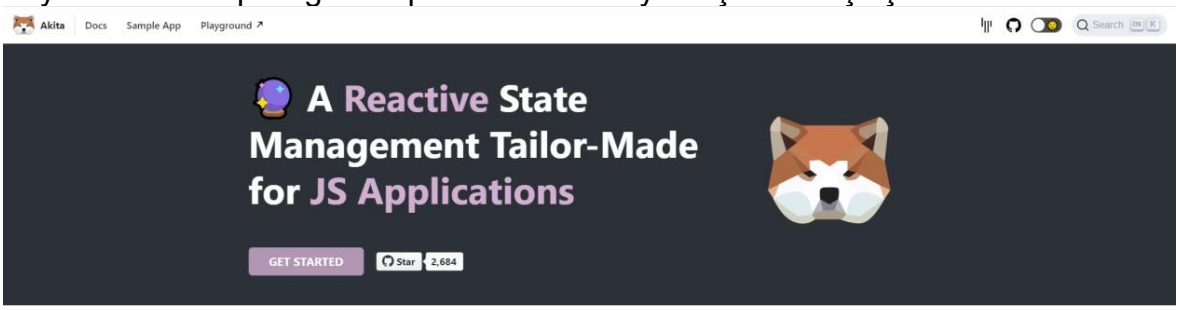
Akita

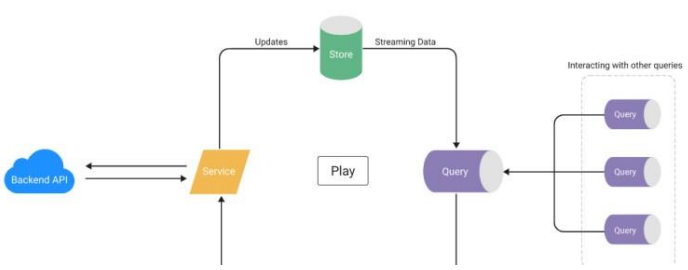
Akita biraz daha az bilinen bir devlet yönetimi kitaplığıdır. MobX'e benzer, ancak MobX gibi işlevsel
bir API yerine daha Nesne Yönelimli (OO) bir API'ye yönelir.

Dolayısıyla Akita, OO yaklaşımıyla kendisini farklılaştırıyor. Özellikle TypeScript ve Dekoratörlerin yoğun kullanımında görülür. Şimdi, TypeScript OOP için ayrılmamış ve çılgınca desteklense de (özellikle bu listedeki tüm kitaplıklar tarafından), Dekoratörler daha çok niş bir özelliktir. Akita
dışında, MobX bu listedeki tek kitaplıktır. Onları resmen destekliyor, ancak odak noktasını v6'da farklı yaklaşımlara kaydırdı.

Yani, tüm Fonksiyonel Programlama (FP) ve kanca çılgınlığı şu anda devam ederken ve Dekoratörlerin
henüz kararlı bir JS özelliği olmadığı gerçeğiyle, Akita sizin cazınız olmayabilir. Belki de bu yüzden, React'ten çok Angular'a (Dekoratörlerin çok daha yaygın olduğu yer) odaklanmış gibi görünüyor.

Ancak genel olarak, Akita ve API'sini seviyorsanız, hiçbir şey sizi React ile kullanmaktan alıkoyamaz.
Sınıf tabanlı bileşenler kullanıyorsanız, harika dokümanlar ve yeterli büyüklükte bir topluluğa sahipseniz özellikle iyi bir şekilde eşleşebilir.

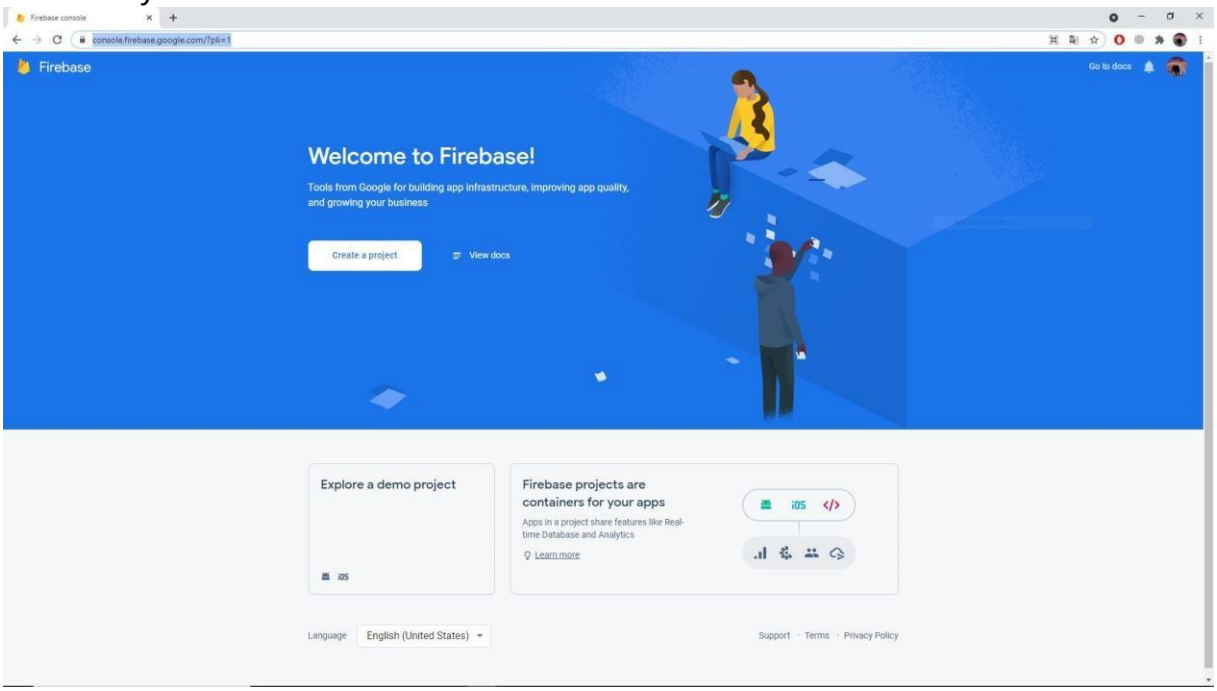




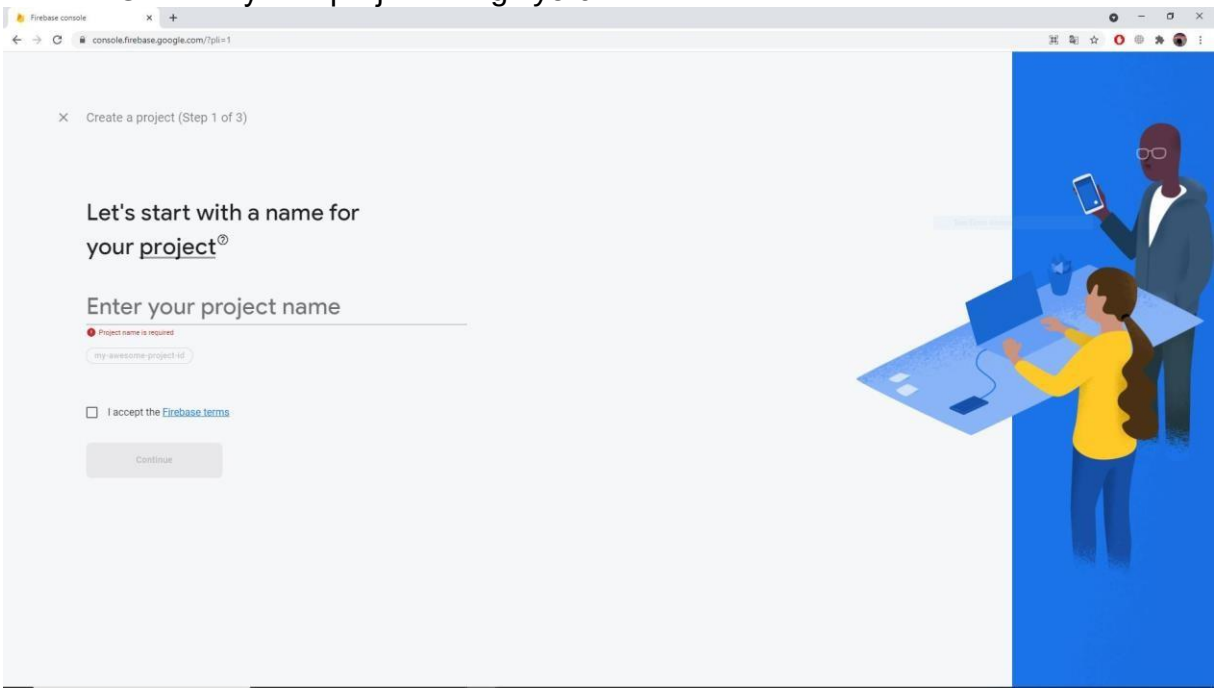
Burak Kalınkaya 171602011

Expo’da Firebase Kurulumu ve Google Analytic Kullanımı

1. https://console.firebase.google.com/?pli=1 sitesine giriyoruz ve “Create Project” diyoruz.

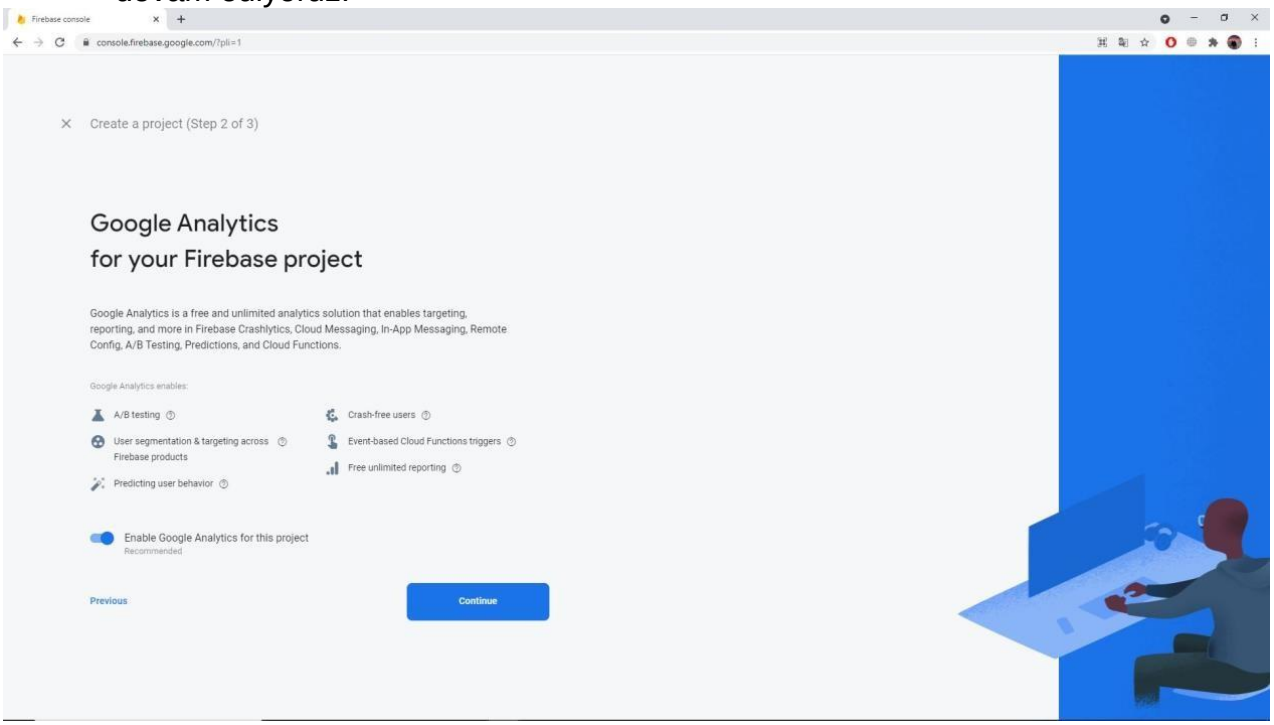


2. Gelen sayfada proje adını giriyoruz.

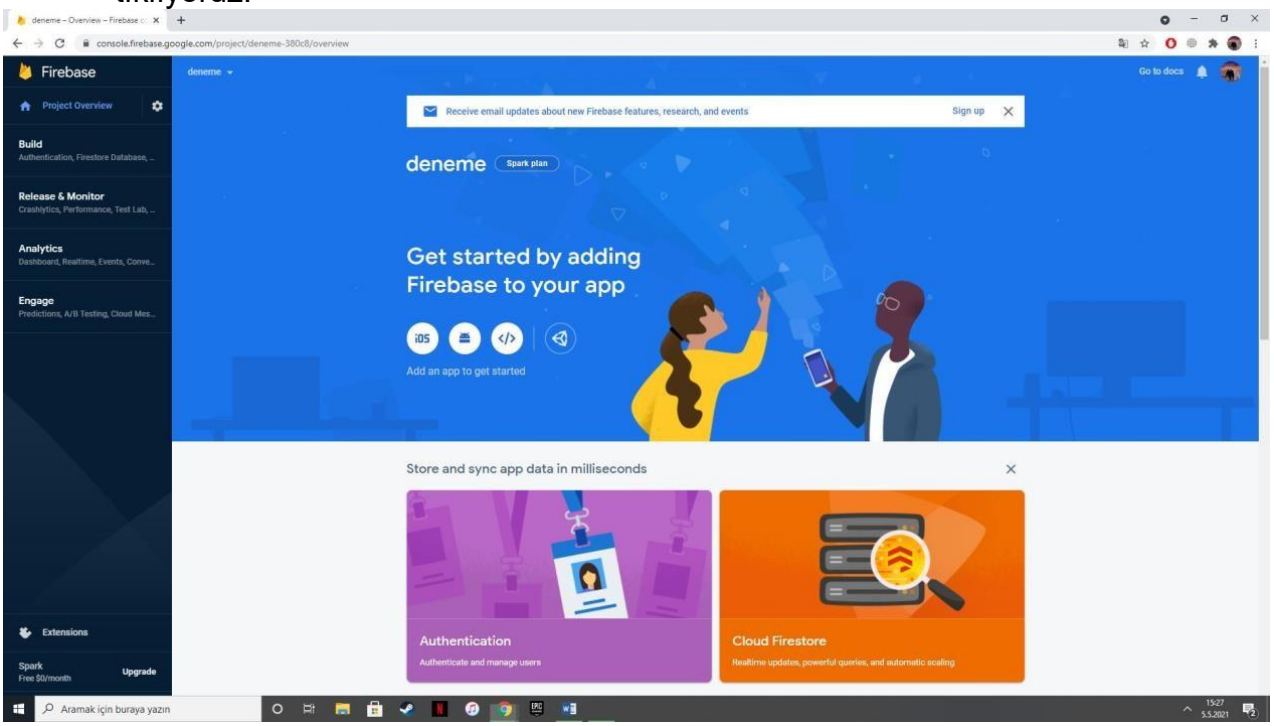


Burak Kalınkaya 171602011

3. “Enable Google Analytics for this project” seçeneğini seçtikten sonra “Continue” diyerek
devam ediyoruz.

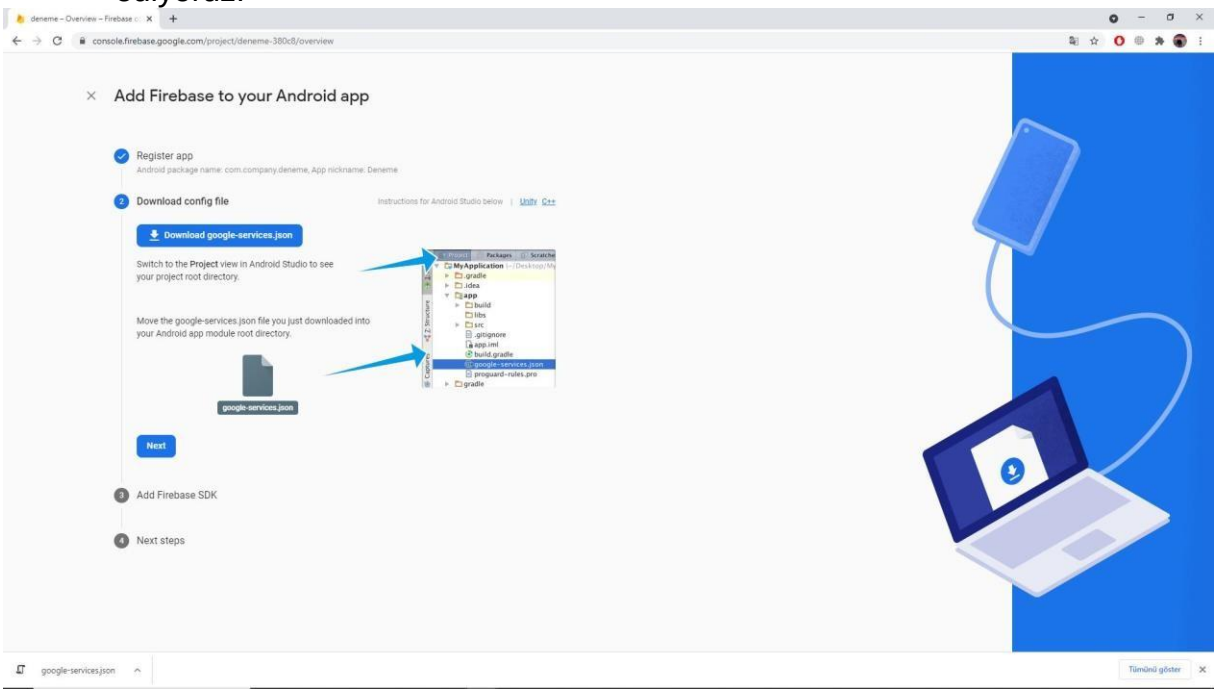


4. Gelen ekranda IOS ya da Android seçeneklerinden hangisini kullanacaksak ona tıklıyoruz.

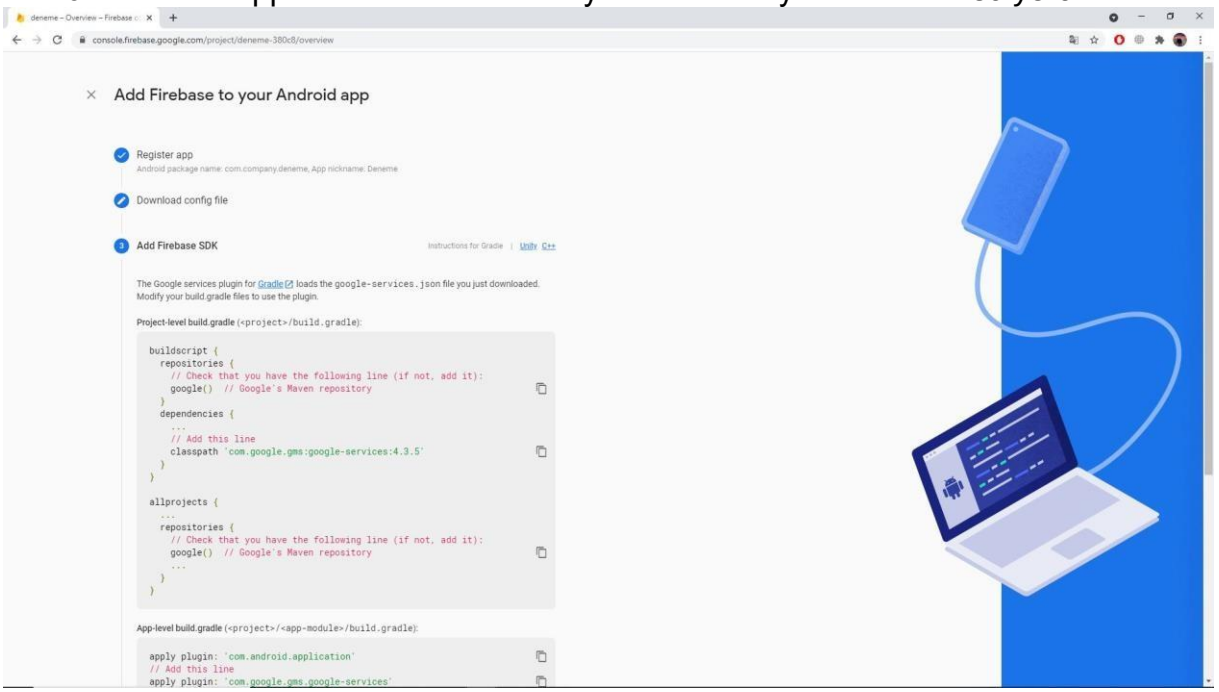


Burak Kalınkaya 171602011

5. “Download Google-services.json” dedikten sonra “Next”e tıklayarak devam ediyoruz.

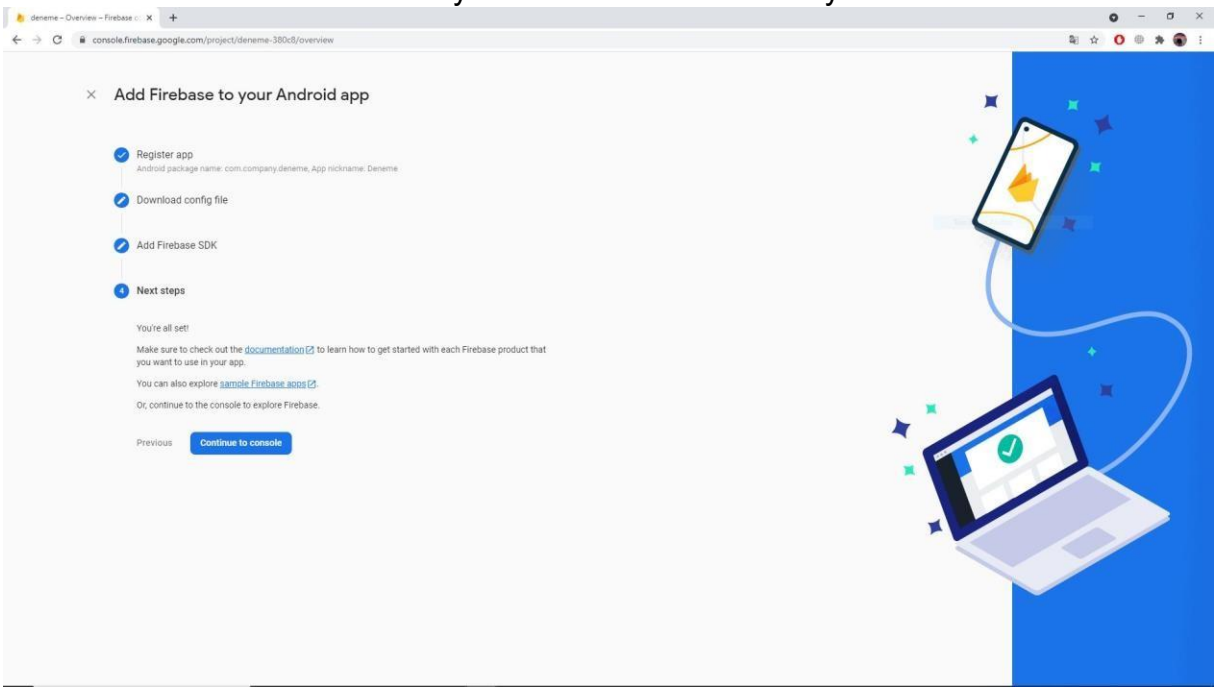


6. Android app ‘imize Firebase’i ekliyoruz. Next diyerek devam ediyoruz.

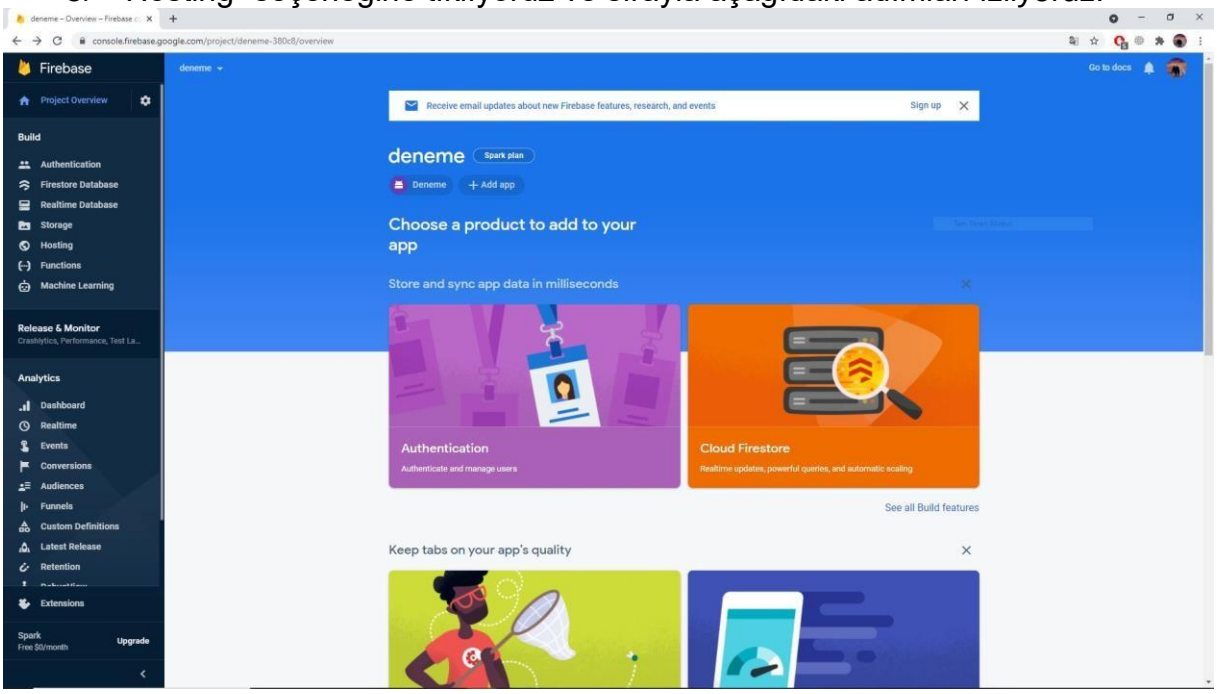


Burak Kalınkaya 171602011

7. “Continue to console” diyerek konsolda devam ediyoruz.

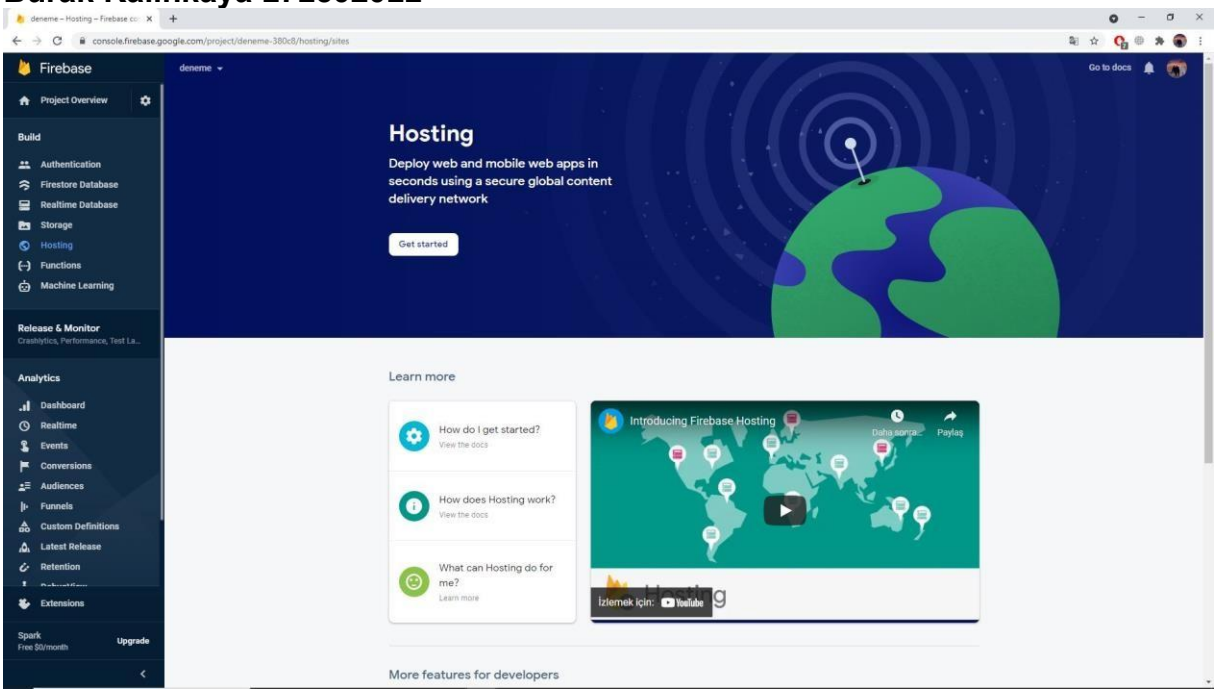


8. “Hosting” seçeneğine tıklıyoruz ve sırayla aşağıdaki adımları izliyoruz.

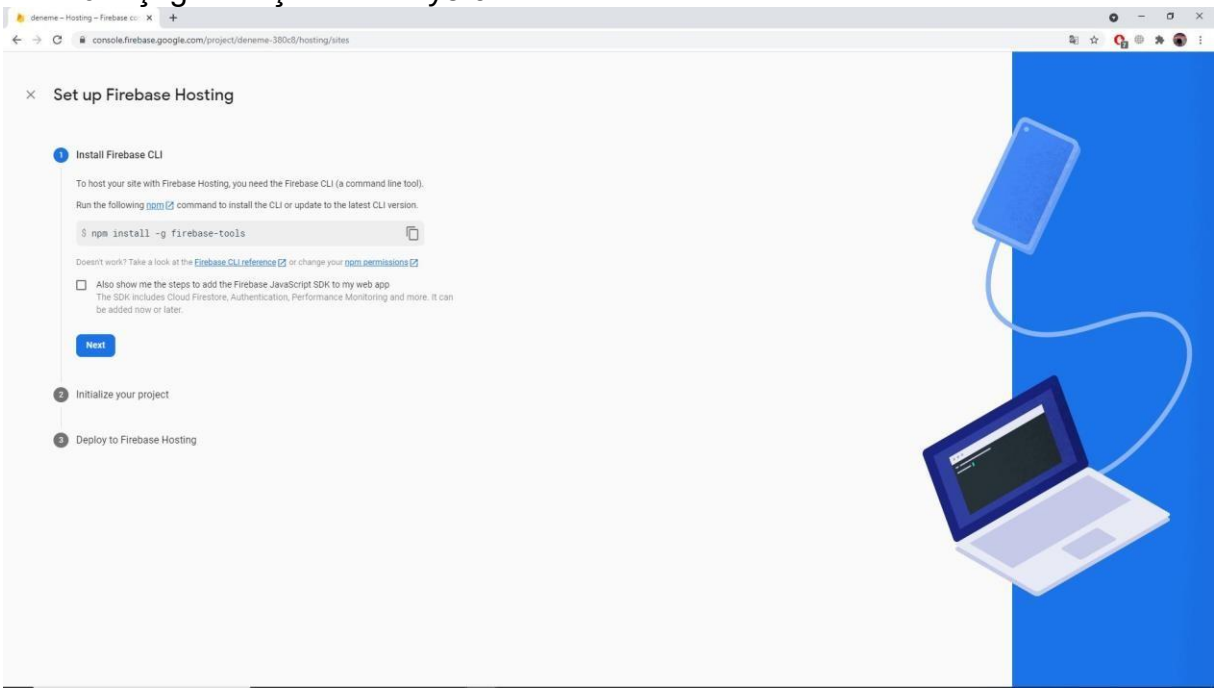


9. “Get started” diyoruz.

Burak Kalınkaya 171602011

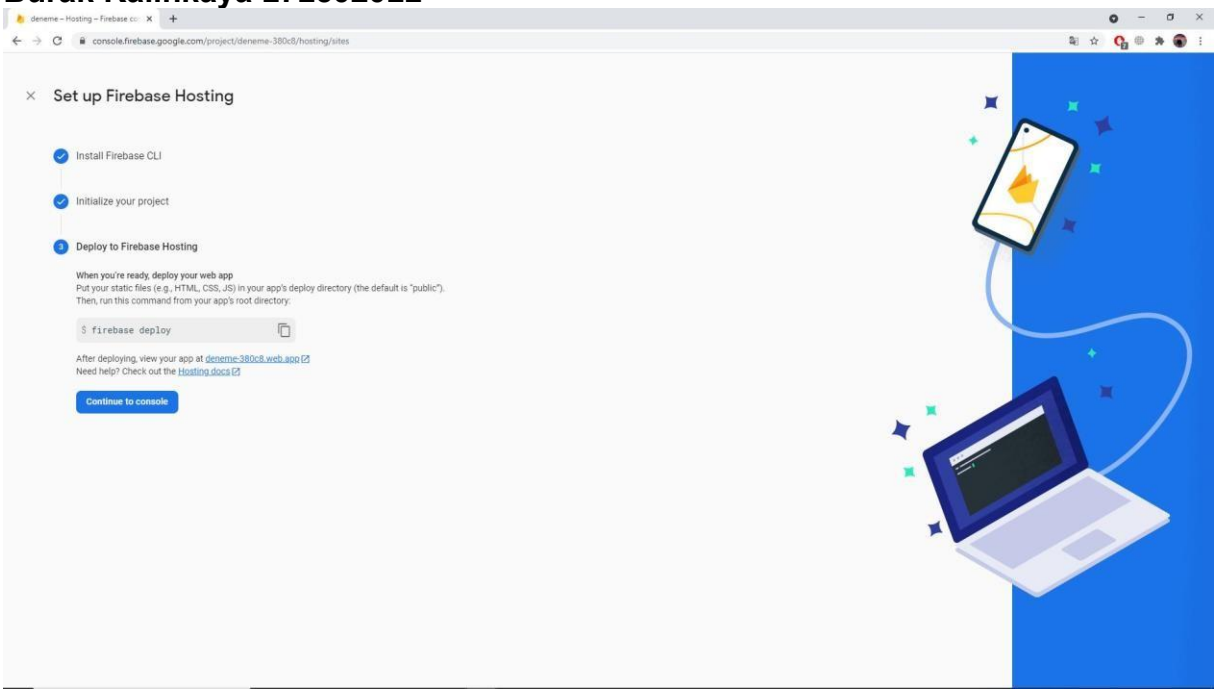


10. Aşağıdaki işlemleri izliyoruz.



11. “Continue to console” diyoruz.

Burak Kalınkaya 171602011



Kaynakça:

- https://devnot.com/2020/recoiljs-nedir/
- https://medium.com/bili%C5%9Fim-hareketi/react-hooks-nedir-9b7545c6c2ff#:~:text=React%20hooks%20bir%20class%20yazmadan,%C5%9Eimdi%20buna% 20gerek%20kalmad%C4%B1.
- https://turerkan.medium.com/react-redux-ve-durum-y%C3%B6netimi-state-management-%C3%BCzerine-935a39677e3a
- https://zaferayan.medium.com/react-native-instagram-redux-kullan%C4%B1m%C4%B1- 10e29c8e3f1b
- https://areknawo.com/top-5-react-state-management-libraries-in-late-2020/
- https://erdemuslu.medium.com/mobx-nedir-nas%C4%B1l-kullan%C4%B1l%C4%B1r- 78a2e9cd205b