

1. **React Hooks**, React bileşenlerinde durum (state) ve yaşam döngüsü (lifecycle) yönetimini daha kolay hale getiren fonksiyonlardır. Klasik sınıf (class) tabanlı bileşenler yerine, fonksiyon bileşenlerinde de durum ve yan etkileri (side effects) yönetmemizi sağlarlar.
2. **useState**, bileşenlerde durum (state) eklememizi sağlar. Örneğin, bir sayaç uygulaması yapmak istiyorsun. Kullanıcı bir butona tıkladığında sayacı artırmak istiyorsun. İşte useState burada devreye giriyor. **useEffect**, bileşenin yaşam döngüsünde yan etkileri (side effects) yönetmemizi sağlar. Örneğin, bileşen ilk kez yüklendiğinde veya güncellendiğinde bir API çağrısı yapmak istiyorsan useEffect kullanabilirsin.
3. **View** modülü mobil uygulamada görünüm oluşturarak buton, resim gibi diğer nesneleri içine alır.
4. **StyleSheet.create** komutu görünüm için stiller oluşturur.
5. **export** komutu uygulamayı dışa aktarmayı sağlar.
6. **“npx expo install firebase”** uygulamaya firebase için gerekli bileşenleri ekler.

```
// firebase.js
import { initializeApp } from "firebase/app";
import { getFirestore } from "firebase/firestore";

// Firebase yapılandırma anahtarları
const firebaseConfig = {
  apiKey: "YOUR_API_KEY",
  authDomain: "YOUR_AUTH_DOMAIN",
  projectId: "YOUR_PROJECT_ID",
  storageBucket: "YOUR_STORAGE_BUCKET",
  messagingSenderId: "YOUR_MESSAGING_SENDER_ID",
  appId: "YOUR_APP_ID"
};

// Firebase uygulamasını başlat
const app = initializeApp(firebaseConfig);
const db = getFirestore(app);

export { db };
```

7. `import { initializeApp } from "firebase/app";`
`import { getFirestore } from "firebase/firestore";`
Gerekli kütüphaneleri çağırmaı sağlar.
8. **firebaseConfig** kısmı yapılandırma ayarlarıdır. “apiKey” bağlantı anahtarıdır bir nevi şifre, “authDomain” bir nevi kullanıcı adıdır.
9. Uygulamayı başlatmak için kullanılan metod **initializeApp**
10. Veritabanına bağlanma metodu ise **getFirestore**