**React Native vs Native(Android iOS)**



### **1. Single Code base**

The best part about beginning mobile app development with React Native is that it provides developers with an opportunity to write a single codebase that then runs on both Android and iOS platforms.

### **2. Low Development Time**

Since the React Native app builders have to write the code just once, it saves a lot of development efforts in terms of app development time.

### **3. Ease for Web Developers**

Since a major part of the React Native app development is based on [JavaScript framework](https://appinventiv.com/blog/javascript-framework-guide/), the transition for a developer to become a [mobile app developer](https://appinventiv.com/) from a web developer is much easier.

### **4. Open Source**

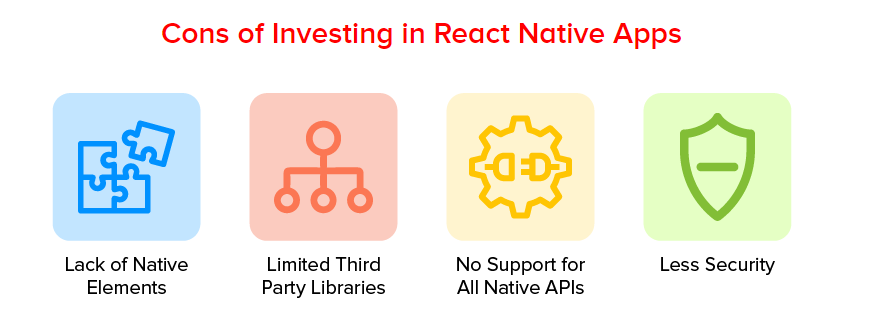
React Native, being open source comes with the benefit of inspecting the code. Also, since it is open source, the library is deemed to be a lot more compatible with other platforms like TvOS and Windows.

### **5. Hot Reloading**

React Native comes with the feature of [Hot Reloading](http://reactnative.dev/blog/2016/03/24/introducing-hot-reloading) using which any mobile app development company can view the changes they are making to the app in real time on their screens.

### **6. Cost Effective**

Since this cross-platform framework allows developers to reuse app components to enter both Android and iOS platforms, the [React native app development cost](https://appinventiv.com/blog/react-native-app-development-cost/) is also quite low.



### **1. Lack of Native Elements**

Since we do not access the native api directly, there may be some natives api for which we may not have React Native wrapper.

### **2. Limited Third Party Libraries**

Community wise React Native Development lags a lot behind Native one and because of this, there is a serious lack of third-party libraries. To make use of the native libraries, the incorporation would have to be made in native modules which increases the development efforts.

### **3. No Support for All Native APIs**

React Native has introduced major updates like making the Flipper debugging tool as default in the latest version, 0.62. But, still it doesn’t have the freedom to use all the APIs.

With this attended to, let’s focus upon the advantages and disadvantages of Native approach to get a clarity of Native vs React Native comparison.



### **1. API Accessibility**

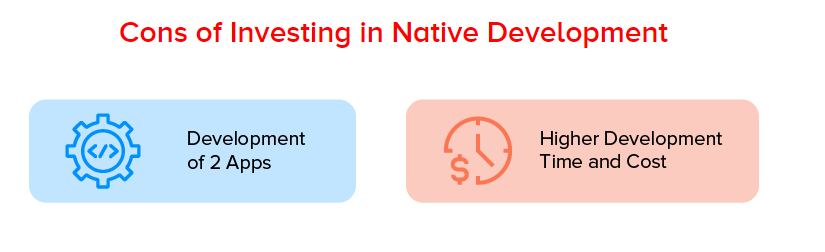
All the device’s [APIs](https://appinventiv.com/blog/complete-guide-to-api-development/) and its inbuilt functionalities can be integrated with the native development environment.

### **2. Use of Third Party Libraries**

Another [reason why Native app development](https://appinventiv.com/blog/seven-reasons-native-app-development-better-solution/) is preferred is that native Android and iOS are much bigger and supported than their React Native counterpart. With the support comes the presence of a lot of third-party libraries to select from.

### **3. Robust Language**

Both Swift and Kotlin are considered to be robust, both in terms of stability and when it comes to revealing the hidden errors in code. This robustness in the programming language is what makes every business leader and mobile app development company interested in the process of how to build a native app.



### **1. Development of Two Apps**

One of the biggest cons of Native development for iOS and Android is that businesses have to invest in both [Android app development](https://appinventiv.com/android-application-development/) and [iOS app development solution](https://appinventiv.com/iphone-application-development/) individually.

### **2. High Development Time and Effort**

Because the app development happens from scratch twice, the [cost to build a mobile app](https://appinventiv.com/guide/mobile-app-development-cost/) and the associated app development efforts grow by manifold.

**3: Complex for same set of developers who develop Web App**

**PERFORMANCE**:

At the end, React-Native generates the mobile native code so performance is same.

**React Native Tools**

There are two popular ways to create React-Native App.  
1: Bare React-Native App

For this we need Android Studio/Xcode for development and to run and test the code.

It is complex time taken to setup to run even the minimal app.

2: React-Native App covered with Expo.

Expo is itself a mobile app, we can install it from Play Store or App Store.

Using Expo we can run our app on browser and see the preview in our own mobile app.

During the development, we can publish the app to Expo site, and we can get it checked by anyone anywhere in the world, jus that to see it we have to have the Expo app installed in our phone and need the credentials.

Expo is a tool which provides wrapper over many complex Api to ease the use.

Using React-Native-Web, Expo app can be developed and debugged on browser.

**React-Native different approaches and techniques to develop and re-use code from React js web app.**

1: React-Native-WebView

If there is any already developed and production running site, we can wrap that site in React-Native-WebView to open it as an app.

2: Render Props

We can use this technique to share the code between React js and React Native.

Pros: Single code for logic parts and two different views for web and mobile app.

Cons: Not scalable, hard to manage. In later phase of development it will take more time than writing separate code for ReactJS and React-Native.

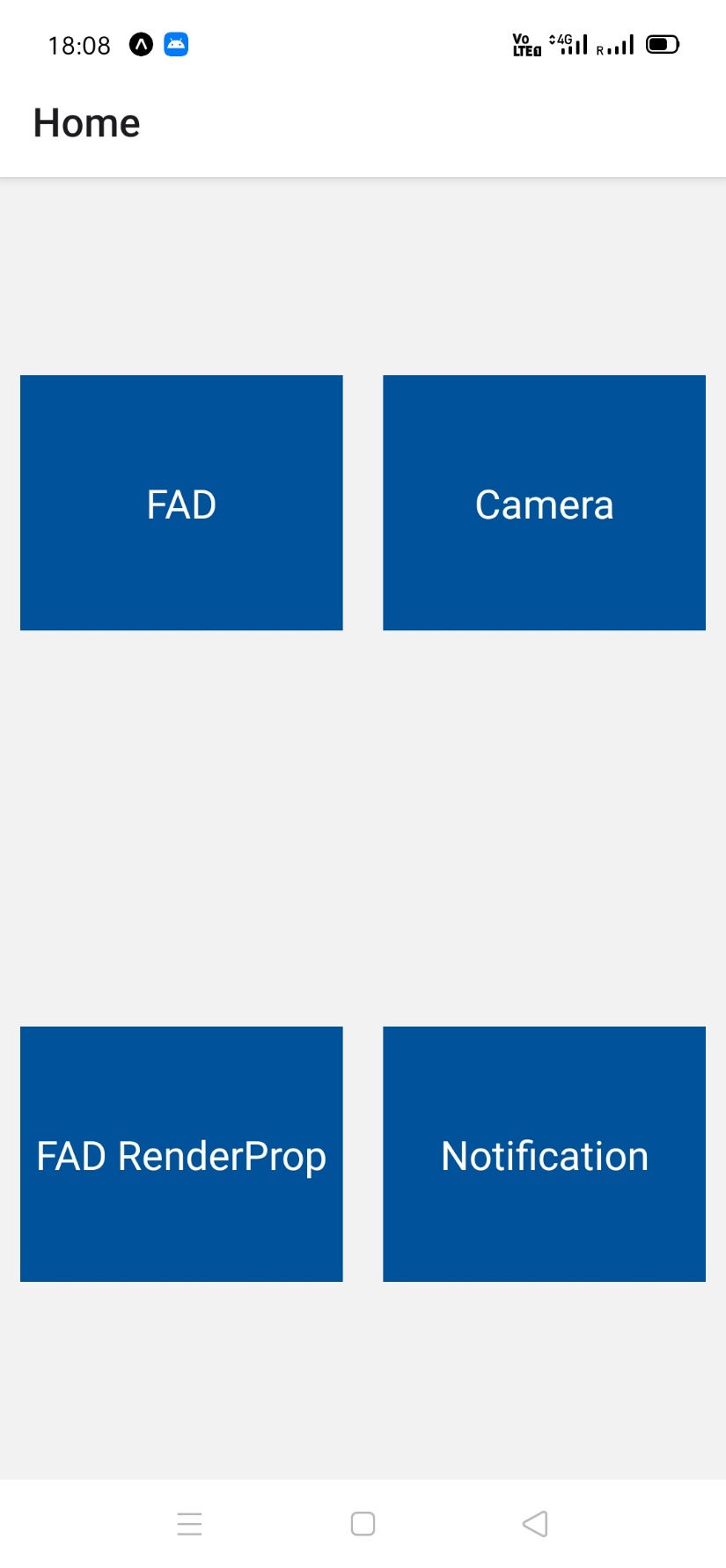
Hard to use one of the strongest features of React JS, composition of components rather inheritance.

3: Create two separate code base, two separate views and logic. Use one common folder/files to keep only complex logics, heavy computation, server/api call, data manipulation…etc and share it. How to share the code this is different thing.

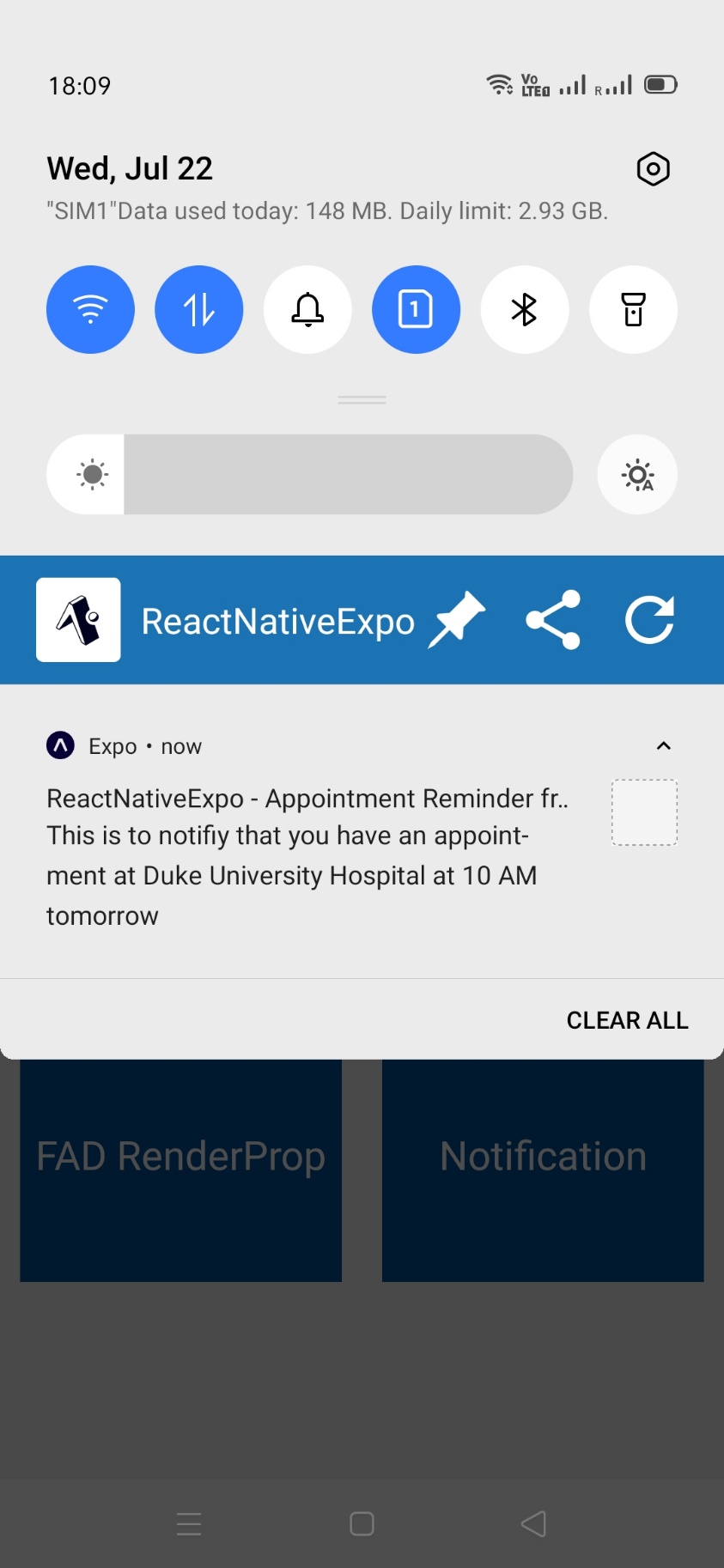
Pros: Easy to manage and scale

Cons: Sometime some code may be write again what has been written in other app

Home Page

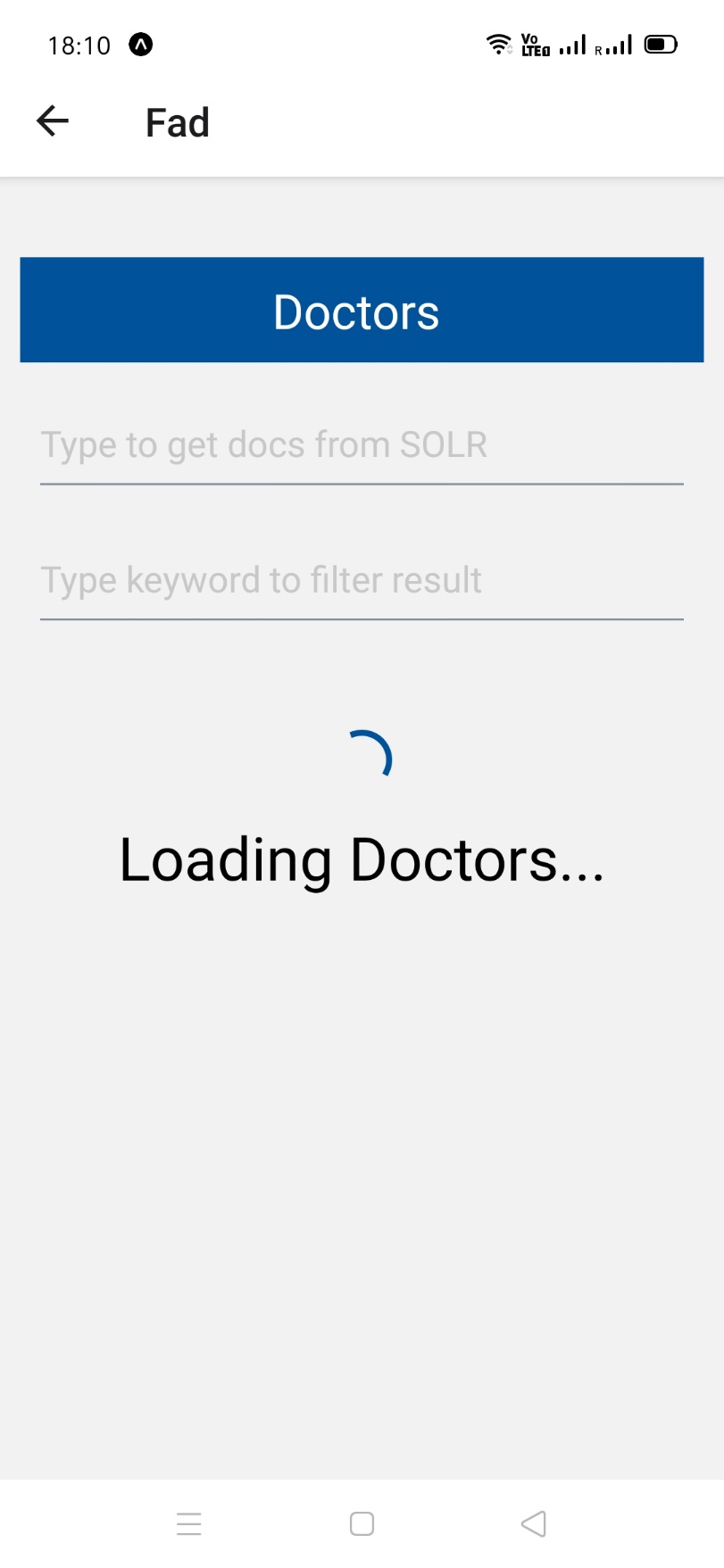


WHEN WE LAND ON HOME PAGE, THERE IS A NOTIFICATION

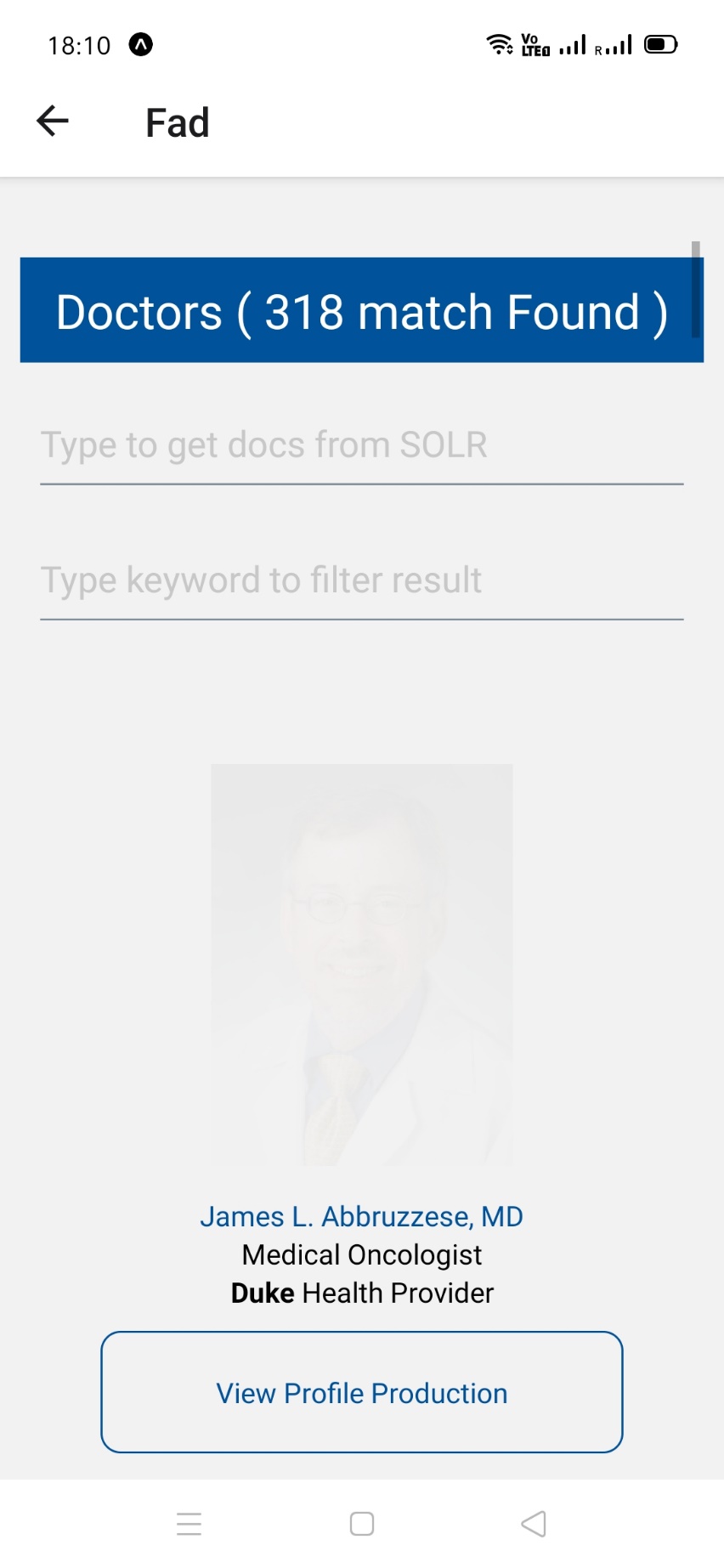


WHEN WE TAP ON FIRST APP “FAD” FROM THE LIST ON HOME PAGE, WE LAND ON FAD PAGE

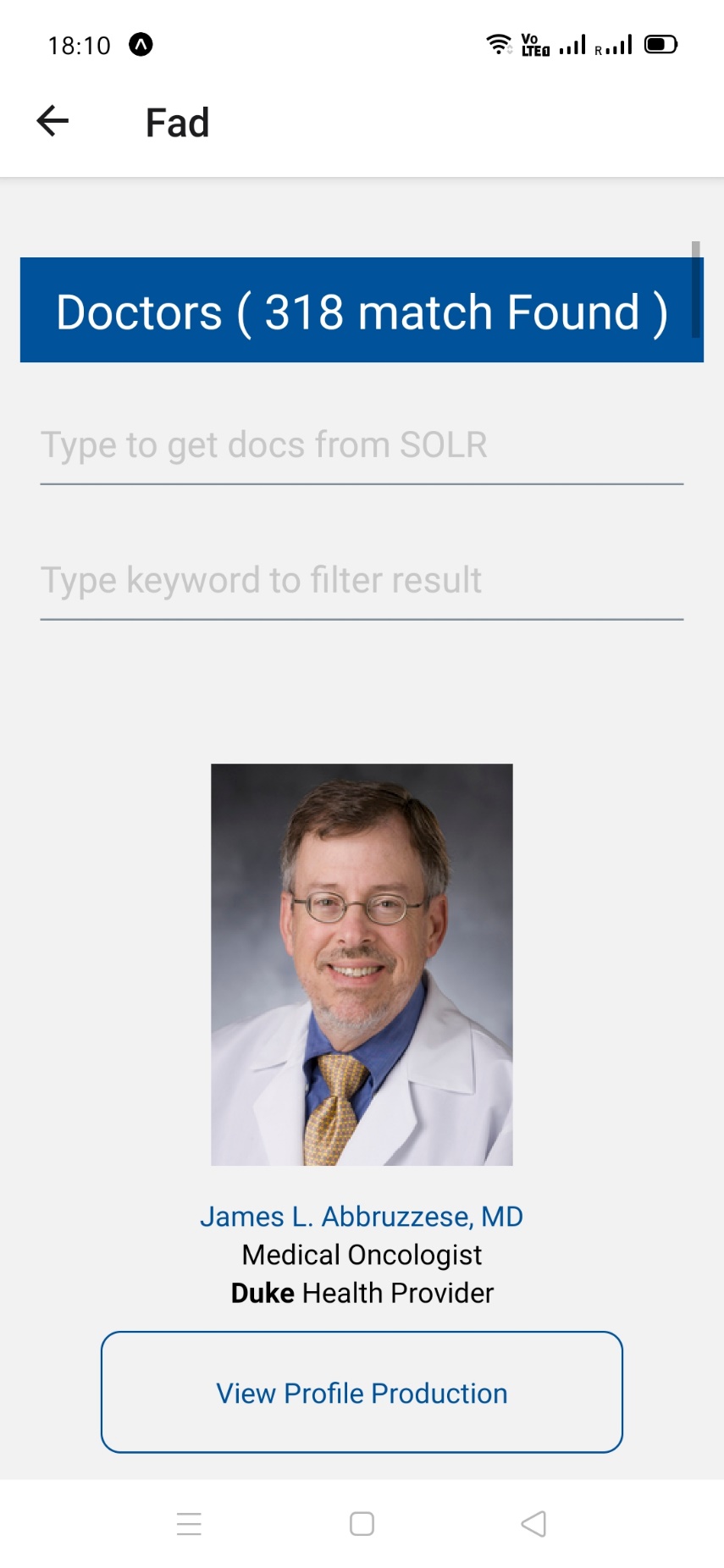
THIS IS THE INITIAL STATE, LOADER’S APPEARING



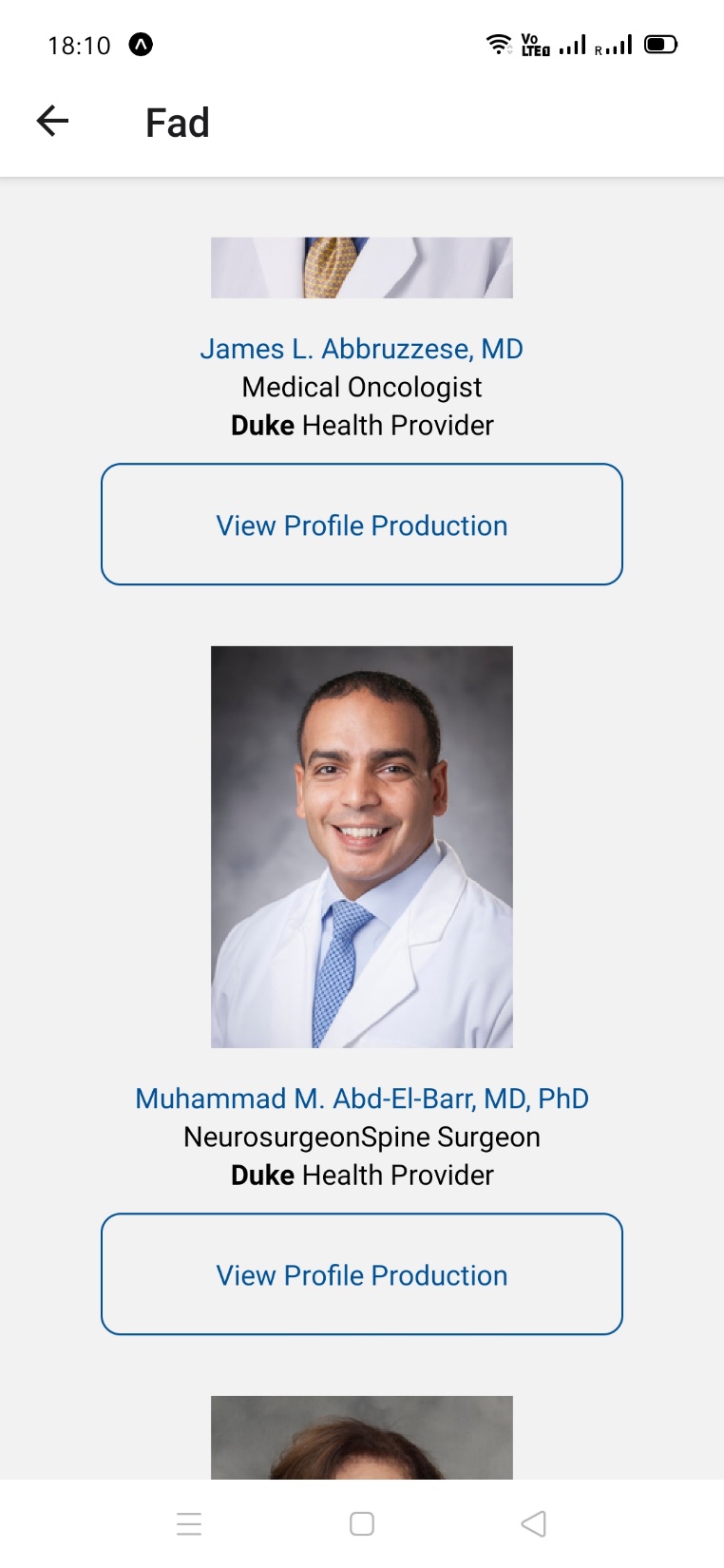
FAD PAGE ALMOST LOADED



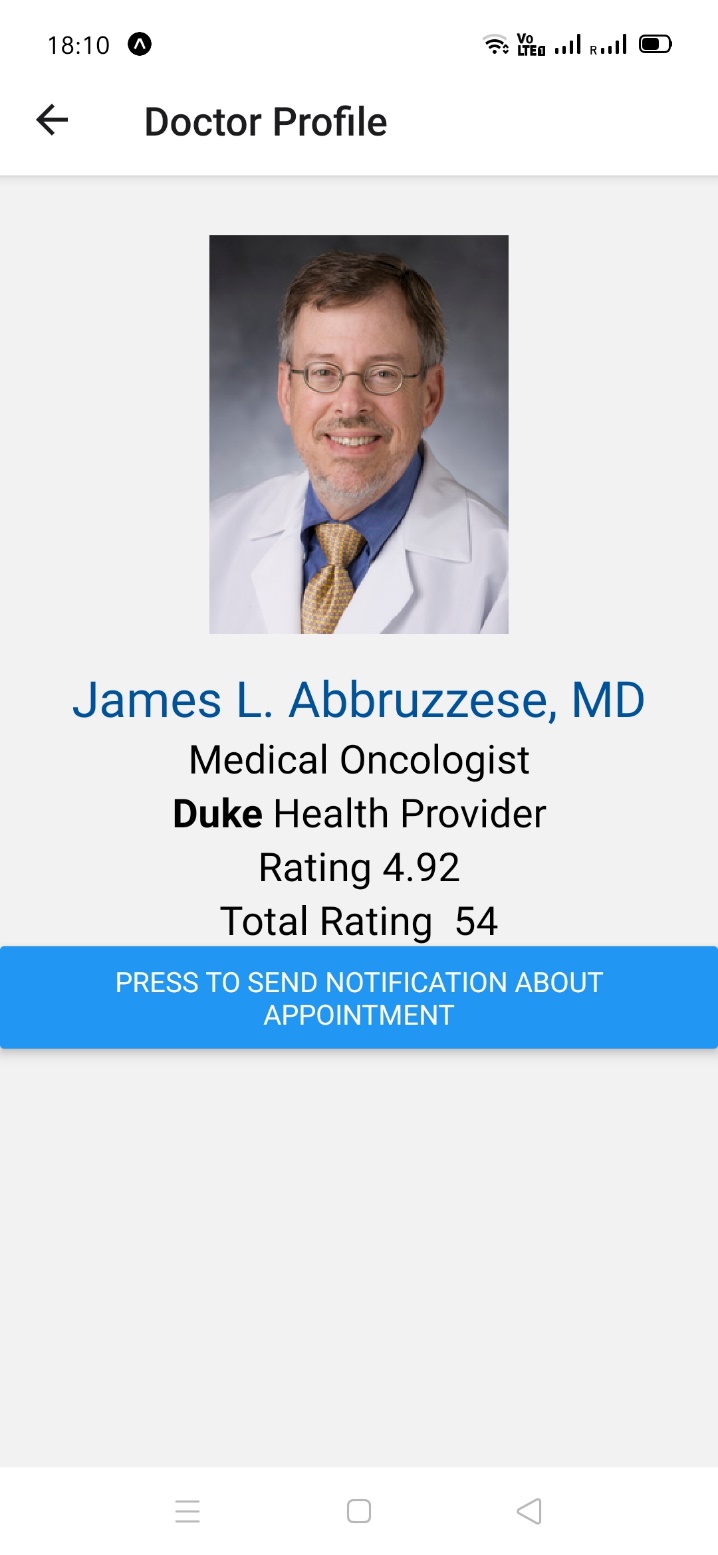
FAD PAGE COMPLETELY LOAD



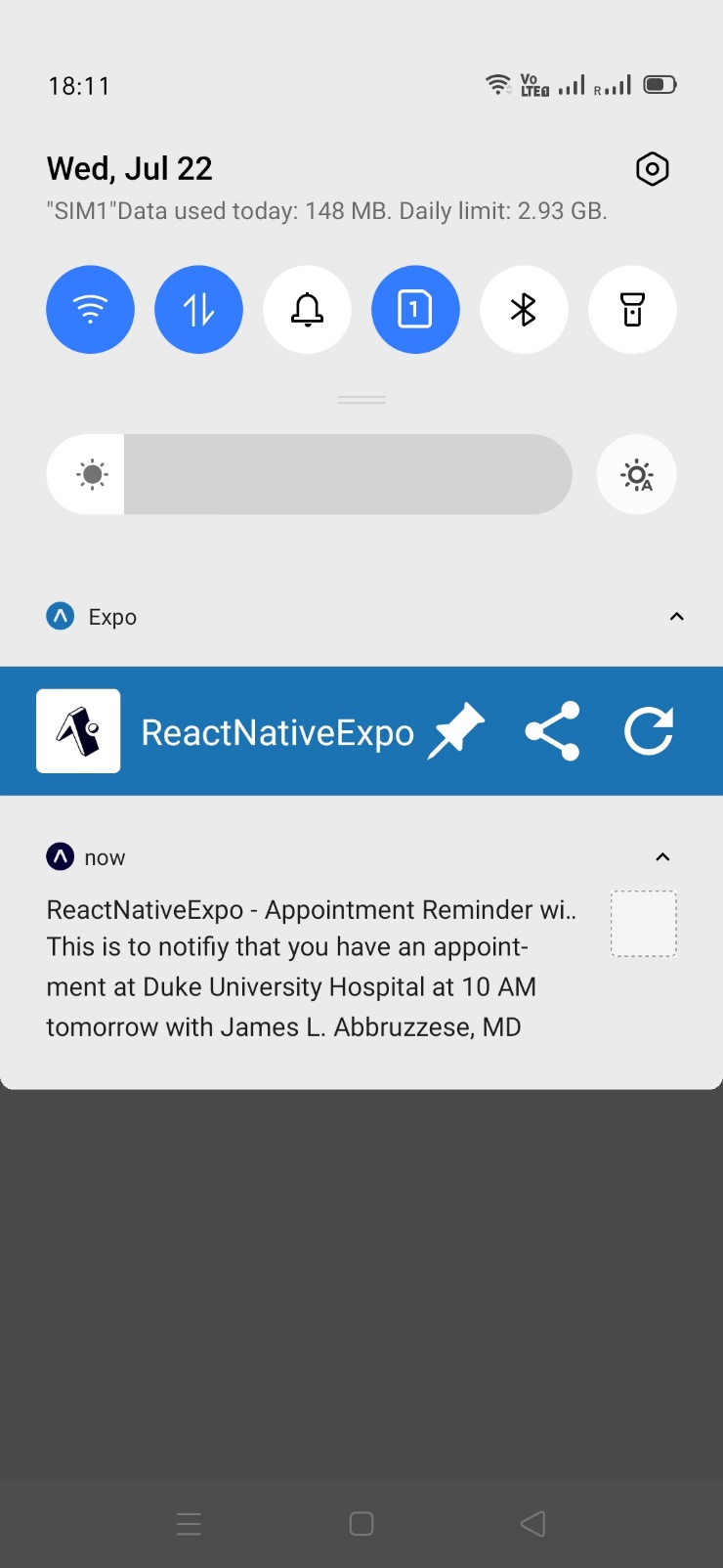
SCROLLED A LITTLE BIT



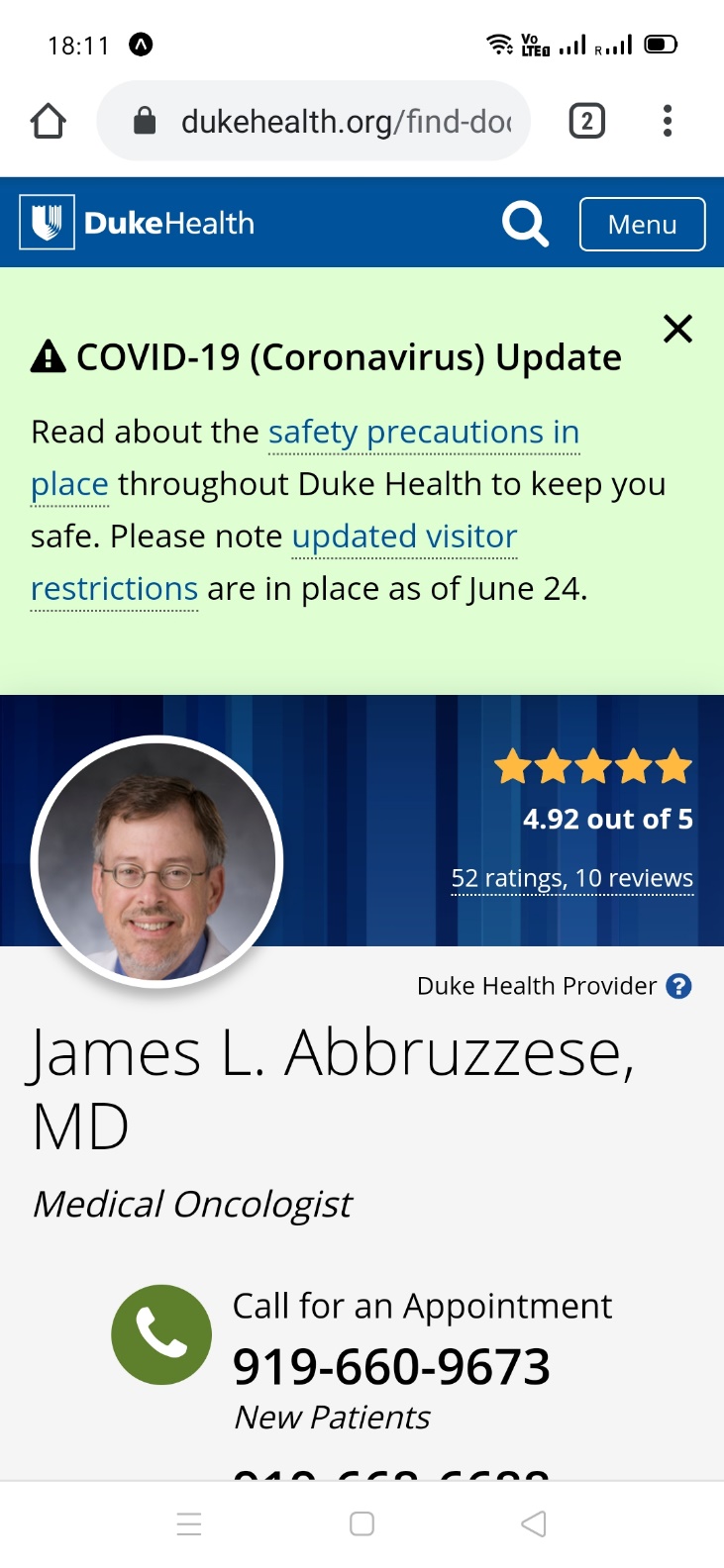
TAPPING ON ANY DOCTOR(PHOTO, TITLE OR VIEW PROFILE)



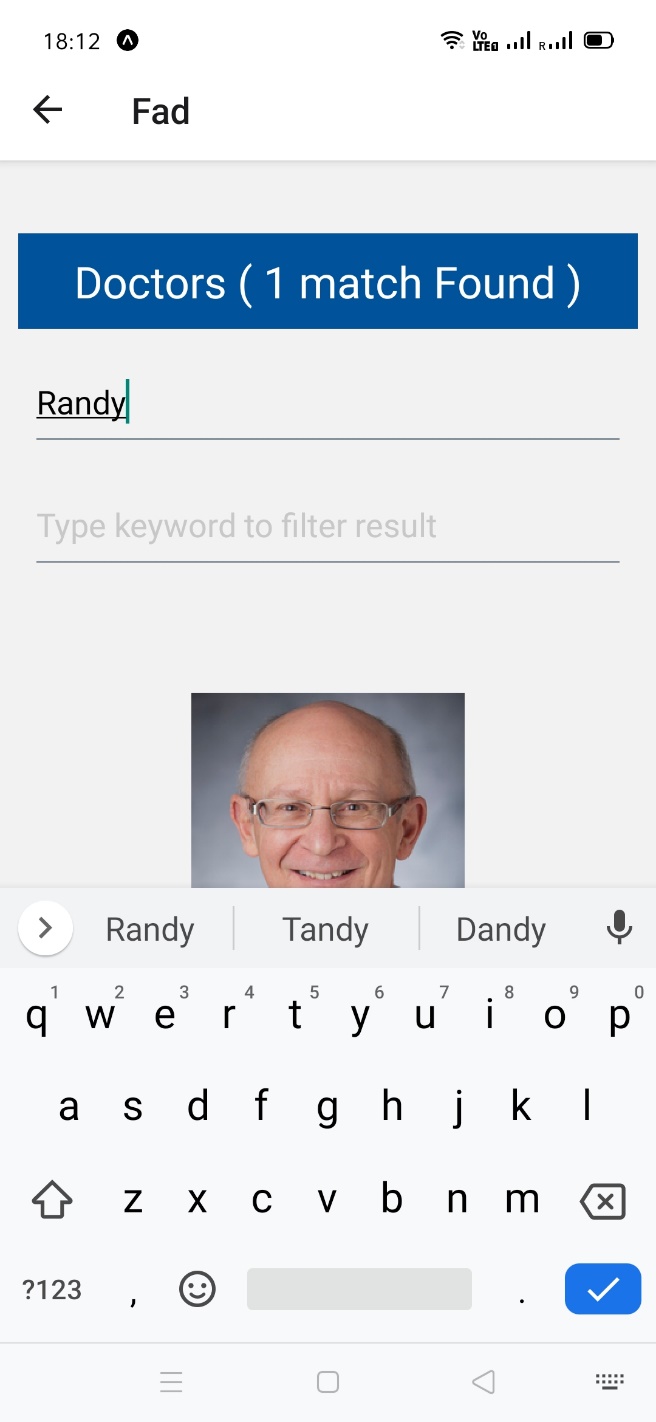
TAPPING ON THE BUTTON “PRESS TO SEND NOTIFICATION ABOUT APPOINTMENT”



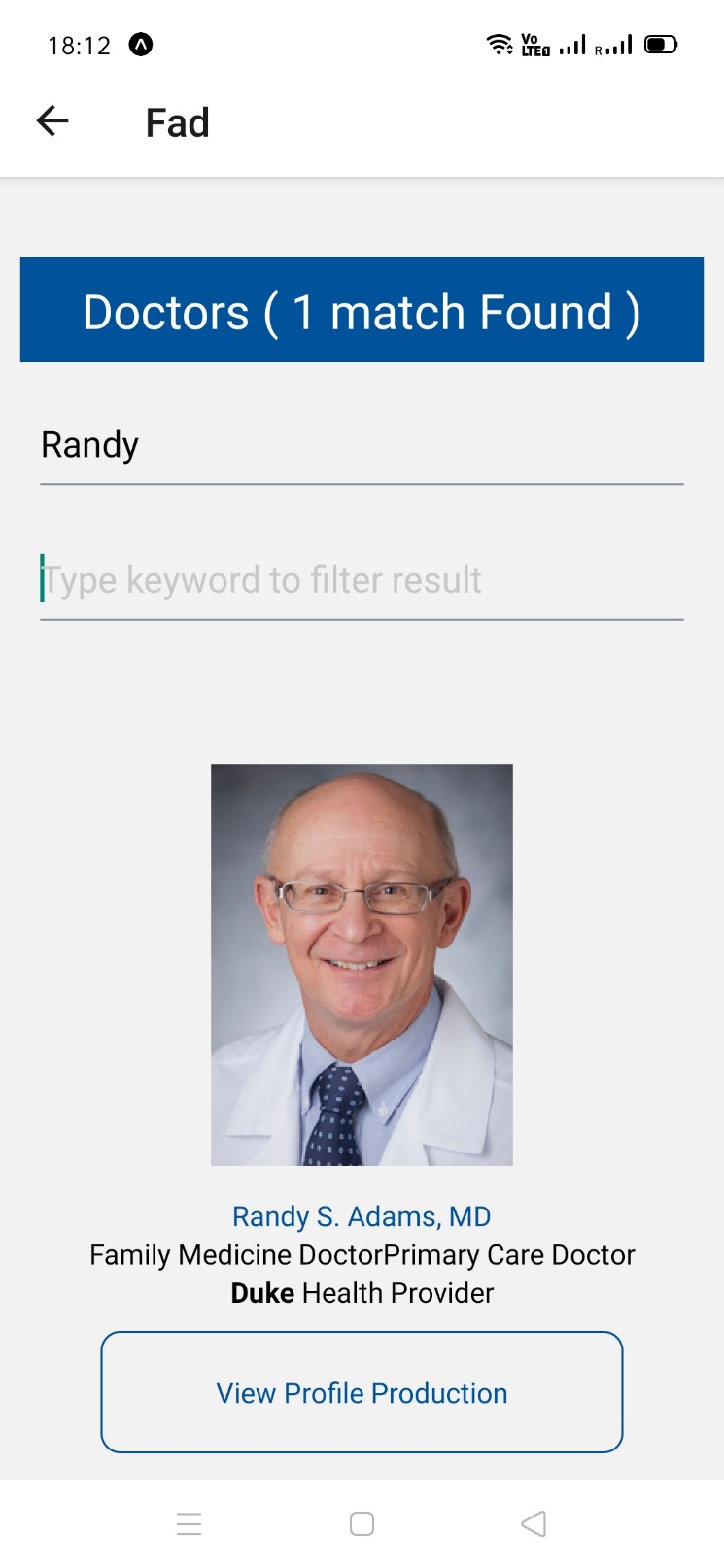
ON PROFILE PAGE, TAPPING ON TITLE WILL OPEN THE DOCTOR’S PROFILE PAGE IN IN PRODUCTION IN BROWSER



ON FAD PAGE, TYPE SOMETHING TO SEARCH



SEARCH RESULT LISTED



REACT NATIVE WEB VIEW (PRODUCITON SITE)

