The purpose of this article is to give you tips which will help you and your group in your React Native activities. A portion of these tips are carefully React Native related, while others are commonly progressively versatile improvement arranged. In this way, how about we get directly into it!

Try not to waste time with Expo

You ought to never utilize Expo on a genuine venture, for various reasons.

Initially, the whole form procedure isn't done locally, yet in Expo cloud, which means practically no control and potential server issues while building. Furthermore, you are adding another layer of multifaceted nature to your task, which backs it off and builds the application size.

Above all, local iOS/Android modules, which you are sure to require at a certain point, are not bolstered. The main local modules that are accessible to you are the ones given by the Expo SDK.

At some point or another, you will arrive at a point where you should discharge the Expo application into an ordinary React Native application, so you should not burn through your time and start the venture right way in any case.

Use ESLint

Utilizing a linter on any task is an absolute necessity, particularly for a wild language, for example, JavaScript. ESLint is an awesome augmentation which is incredibly simple to introduce, set up and use, so make certain to incorporate it in your task from the earliest starting point. In the event that you can't help contradicting a portion of the guidelines, you can generally incapacitate them in the .eslintrc design record, yet having some sort of code quality control is going to support you and your group colossally over the long haul.

Expel all logs from your discharge constructs

Having numerous console.log articulations can back your application off, particularly in the event that you are utilizing logging libraries, for example, revival lumberjack. Make certain to debilitate all logs (physically or with a content) when making a discharge manufacture.

Use Flexbox

This should be an easy decision, however tragically I've seen **React Native** code that utilizations outright component situating rather than flexbox, which breaks the design and gives no reusability at all. Regardless of what your structure necessities are, utilizing flexbox is quite often the right answer. For the individuals who originate from web foundation: Flexbox in React Native works precisely equivalent to its CSS partner, the significant distinction being that flexDirection defaults to segment rather than line in React Native.

Name your style protests reliably

In the event that your root view has a style article named compartment, at that point utilize that name for each root see on the undertaking. In the event that your submit catch has a style item named submitButton, at that point stay with it and don't utilize saveButton, submitBtn or submit for different cases.