Choosing Between A Native & A Hybrid App

In this lesson, we will have an insight into how to choose the right type of mobile app for our use case?

We'll cover the following

When Should We Pick A Native App For Our Use Case?

Here are the scenarios, listed below in bullet points, when we should go ahead with a native app:

- When we have heavy, graphic and hardware requirements like for a mobile game or a game streaming app. This is the scenario when we need top-notch performance from the app & even a tad bit of lag is unacceptable.
- When we intend to write an app that has heavy UI animations, for instance, a fancy social app containing a lot of animations or a finance app containing a lot of real-time charts and graphs & stuff. In this scenario, it's just not okay to have any sort of lag in the application. The application needs to be as responsive & reliable as it can be.
- When the app is pretty complex & is reliant on hardware access, like camera, sensors, GPS etc. to function. In this use case, we generally have to write a lot of platform-specific code. A GPS, sensor-based health and a step-tracking app is a good example of this.
- When the look & feel of the app and the user experience should be just like the native OS. When the UI needs to be, not just functional, but flawless.
- When you have other businesses in the same niche competing with you with native apps. It would be a blunder to offer our service via a hybrid app. User's today aren't installing as many apps as they used to. Don't expect them to

today aren't installing as many apps as they used to. Don't expect them to

show mercy on you when you don't have a product that is better than your competition.

- When the app always needs to support new mobile OS features as soon as they are released.
- If you are a business that can afford dedicated teams for Android & iOS, you should go ahead with native apps. Don't even think about the hybrid app approach.

When Should We Pick A Hybrid App For Our Use Case?

- When the app requirements are simple, there is nothing complex. Also, in the future addition of any new complex features isn't expected. A news app is a good example of this. Developing a news app as a hybrid app will also provide the same look and feel of the app across all the platforms.
- When you just cannot afford dedicated codebases for platforms but still have to hit the market. There are two approaches to this either launch with a native app on one platform or write a hybrid app. This entirely depends on how you want to go ahead.
  Yes, the native apps provide top-notch performance but we cannot entirely discard hybrid tech on the grounds of performance and the availability of other native features. There can be instances where you won't even need dedicated apps, a hybrid app could fulfil all your requirements pretty well. It all depends on your requirements.
- When we just need to test the waters with a *pre-alpha* release or an *MVP* (*Minimum Viable Product*). In this scenario, it won't make sense to learn the native tech and then write the app. You can quickly launch the MVP via a hybrid app written with the open web technologies.
- When you have a team that is not fluent with the native technologies and it would take a lot of time to learn that tech. This scenario is a trade-off between costs and performance. Also, the developer sentiment is another aspect to this.

So, these are some of the general rules that we can follow when taking our pick of the two types of apps. Another good approach is to find the businesses in the same niche, research on what technologies they have used to write their apps.

With this being said, let’s move on to the next lesson where I’ll be discussing progressive web apps.