**Xamarin**

Let's you code across multiple platforms at the same time, IOS, Android & Windows 10.

Made by Microsoft

Coding in C# & XAML

Just use visual studio to create the application

They say you can use around 80/90% of the same code across all platforms

Uses .NET so handles all of the memory allocation etc itself

Xamarin.Forms looks the best - can build IOS, Android and Windows apps from the same codebase. XAML in the front with C# behind

Machine generated alternative text:
O 
Xamarin. Forms 
O 
Shared C# Logic 
With Xamarin.Forms: 
More code-sharing, all native 

They kinda want you to use their own Azure app service for backend but can use anything. Out of AWS, Google & Azure, Azure tends to be the cheapest anyway

Supposedly get a mobile app and a SQL server for free as a student - $100 credit as a student - <https://azure.microsoft.com/en-us/free/students/> - need to look more into it - says 250gb free SQL database

API walkthrough - <https://github.com/brminnick/XamList>

Pros:

* Native-like performance
* Able to design view quickly
* .NET and C# all you need to build apps
* Strong design tools - Visual studio helps to create good UI's
* Multiple platforms

Cons:

* Big app size - adds 5/20mb for converting the C# to native
* Good for apps with simple AI but can't do complex games or apps
* Need knowledge in native programming languages e.g. Java, Swift, Kotlin, in case you need to rewrite code in native
* Takes a while for new updates to come out or problems to be fixed
* Only free for small teams

**PWA**

PWA's are interesting too - don't need to push to the app store etc, they can just add it straight from the website - <https://web.dev/what-are-pwas/>

Wider look at PWA's - <https://web.dev/progressive-web-apps/>

Good place to start - <https://www.freecodecamp.org/news/build-a-pwa-from-scratch-with-html-css-and-javascript/>

A PWA is basically just a web app, converted to a traditional app, with the advantages of both. It is able to be installed on mobile devices extremely quickly, you're able to access it even if you're offline and you can do device specific tasks like notifications etc.

**React Native**

React native is another good place to start. It's all coded in JavaScript, so easy to do as we have a decent amount of prior experience and it's extremely easy to code with on desktop.

Only have to write the code once and it works on Android & IOS

It's very similar to Xamarin - Facebook's version of it

Pros:

* Best user interface - easy to make and look good - Known as open-source JavaScript
* Reliable - developed and supported by Facebook
* Access to native functionalities such as camera accelerometer etc
* Can make quick changes to apps and see the changes extremely quickly as don't need to recompile every time - Hot Reload
* Free

Cons:

* Not good for transitions and animations - not a problem
* Not amazing security

**Flutter**

Similar to both Xamarin and React Native with it being easy to create apps across multiple platforms at the same time - Google's Alternative

It's quite new

Doesn't use the native UX/UI - is actually faster but it draws it all itself

Uses the Dart language which is very similar to both Java and C#

Pros:

* Can fix bugs and see changes quickly with Hot Reload - emulators to quickly see what works etc
* Very customizable - has a giant library of prebuilt widgets, APIs etc
* Free
* Reliable - developed and supported by google

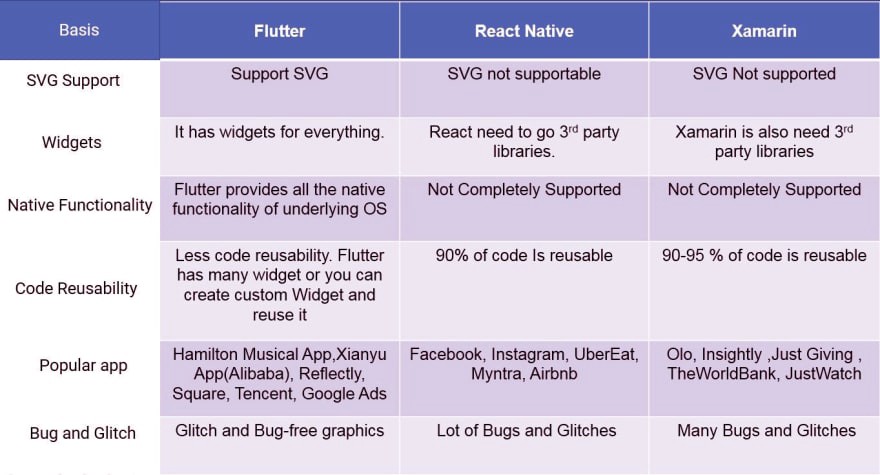
Cons:

* Large App Size
* Quite new so some native APIs are not accessible

**Comparison**

<https://medium.com/@stenalferd/flutter-vs-xamarin-vs-react-native-let-the-battle-begin-d3e783bb4bf1>

<https://blog.logrocket.com/flutter-vs-react-native-vs-xamarin/>



**Qs & Conclusions**

<https://www.simform.com/react-native-ecosystem-backend-database-best-libraries/>

<https://www.simform.com/react-native-database-selection-guide/>

Would like to use React Native/Flutter - Xamarin looks good and has a lot of features but it takes a long time to test code as you have to physically install the app when something changes it also needs a lot of specific experience with each system as things can go wrong and you'd have to write code in the native language. Although Xamarin could be better due to it’s support with other technologies.