Mobile vs Web Stack FAQ

Which is easier (1) Design a mobile app or (2) design a web site optimised for a smartphone

Building any sort of native mobile application is very difficult for students with limited programming abilities. Assuming the application does not need access to device functionality such as Bluetooth, direct disk storage, real-time AV etc., it is best to build either a responsive or mobile only web app.

Option 2 is definitely easier, especially if a front-end framework such as Bootstrap is used. The sandboxed nature of mobile apps makes debugging difficult so that the programmer needs to be accurate when writing code, otherwise things may not work and there may not be clear error messages to help debugging.

A normal Web site is more open so it's easier to debug for beginners. Using platform-specific SDKs (e.g. Swift programming for iOS and Java programming for Android) is even harder because the programmer needs to learn a proprietary platform/API with its own idiosyncrasies.

Which stack would be best for beginners ?

Unfortunately, experience with a low-level programming language such as C/C++ will contribute little to understanding the architecture and deployment of a web application. If the aim is to produce an app with some sort of business value or visual appeal, one needs to learn some basic HTML/CSS and Javascript at a minimum.

The hardest for beginners is to understand the differences between front-end and back-end frameworks, server-side and client-side rendering and database options.

For beginners, it is simpler to:

1. In terms of reducing the workload of having to learn CSS and HTML syntax, and keeping the front-end looking clean and intuitive, Bootstrap CSS frameworks are recommended.
2. For the front-end, consider a JavaScript based stack. For more advanced features, potentially use a front-end framework like Angular, React or Vue.js.
3. Adopt a Python back-end using the Flask web framework (which comes with the Jinja templating engine by default) and use JSON files as a database. Students can find very basic boilerplate or code snippet online that would just read and write from a JSON file.
4. For database (if needed), using SQLite is recommended in terms of database functionality as other databases will be more complex to learn from scratch and will involve much higher risk of getting stuck with setup and configuration.

An alternative to (3) is to adopt JavaScript for the back-end but the language has few concepts that are likely to cause beginners trouble. These include arrow functions, asynchronous functions and promises. As well as this, the package management system for Node ‘npm’ can initially be quite confusing. In comparison, while Python doesn’t have the same curly brace syntax as the students may be used to, it is still very approachable in terms of code cleanliness and conciseness.

There is good documentation for both the Flask web framework and Jinja templating engine online.

E.g..: **https://pythonprogramming.net/bootstrap-jinja-templates-flask/**

In combination, these components would allow the students to produce a basic dynamic web application

Other tools can be useful. As the development environment (IDE), while Visual Studio Code is the industry standard go to IDE for web development these days, PyCharm has very nice debugging capabilities built in and students should be able to get a free education license. For Devops (continuous integration), Github Actions is free and integrated into Github. For a platform with free tier and easy to get started. Heroku is recommended if you want to host application online.

What if I want to use a framework or platform specific for mobile app development ?

The following frameworks allow mobile app development:

- React Native (conventional): https://reactnative.dev/
- Nativescript-Vue: https://nativescript-vue.org/
- Flutter (new one by Google): https://flutter.dev/

There are platforms for both Web and mobile app development:

- Cordova (previously known as PhoneGap): based on the Node.js platform with a lot of JavaScript programming and JSON configuration files. I don't think it's easy to get started because it's almost entirely controlled via command line, uses lots of modules developed by different people, and quite often the developer needs to understand how things work under the hood to get programs to compile and run. Debugging is also complicated because there are no tools, just straightforward browser debugging. There is a tutorial here that https://www.tutorialspoint.com/cordova/index.htm should give you a taste of the framework, and the official software homepage is https://cordova.apache.org/
- Xamarin: An app platform for building Android and iOS apps with .NET and C# (https://dotnet.microsoft.com/apps/xamarin). A good Xamarin introduction can be found at https://www.youtube.com/watch?v=pc0sOFHdOcA