**ASSIGNMENT 1**

Masters- Mobile Development

**IHUOMA FAVOUR AGBARU**

Task 1

1 mark for full completion - to verify you've done this task, just write that you've done so in your answer (we'll trust you). This task is really just to make sure you have smooth sailing for the entire course.

Make sure you have installed the following items! This is incredibly crucial - some of these stuff won’t be used for weeks, but having it installed (or at least being aware of installation issues) far in advance is massively useful regardless. Take a look below!

**Core items required:**

* [ ] Expo Go (on your personal phone)
* [ ] Node.js
* [ ] NPM (Node Package Manager)
* [ ] Watchmen (for Mac / Linux students)
* [ ] npx (Node package execute - often comes incl. with npm)
* [ ] Slack
* [ ] Zoom
* [ ] Any IDE or code editor you prefer

**Tertiary Items:**

* [ ] Lodash
* [ ] Git
* [ ] React Native Elements
* [ ] React Native Navigation
* [ ] React Native Lottie Web
* [ ] React Native Redux

**Answer:** All requirements listed have been installed.

**Task 2**

❓Questions:

* What is a transcompiler?

**ANSWER:** A transcomplier is defined as a specific kind of compiler that changes the source code of a program into a different language. Additionally, it may process an earlier program written in a programming language and convert it to a more recent version of that language. It is also called a transpiler or a source-to-source compiler.

A transcompiled state is where many programming languages start. For instance, there was no native compiler in the first iterations of C++. Instead, C++ programs were transcompiled into C programs, which were then converted to binary using the C compilers already in use. Converting already compiled code into the source code of a different language is another application of transcompilation. As an illustration, consider Emscripten, a transcompiler that turns LLVM-compiled C and C++ programs into JavaScript.

**BENEFITS OF USING TRANSCOMPLIER**

* Adding new features to an existing language. For example typescript adds static typing and compile-time type checking.
* Porting a codebase to a new language or converting code to a new version of the language.
* Being more efficient when coding. E.g Coffeescript is more readable than Javascript and also during compilation mistakes can be detected earlier.
* What is the difference between Native, Hybrid and Transpiled approaches for Mobile Development?

**ANSWER:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S/N** | **NATIVE APPROACH** | **HYBRID APPROACH** | **TRANSPILED APPROACH** |
| 1 | A native app is written using the development languages and tools that are specific to each platform. | A hybrid development adopts a write once, run everywhere approach to building an application. | * A transpiled approach adopts Porting a codebase to a new language or converting code to a new version of the language. |
| **2** | The languages used for iOS development - swift or objective C and for Android Java or Kotlin. | The languages used in hybrid applications are HTML, CSS and Javascript. | Javascript is converted to Coffeescript. |
| **3** | They require high maintenance. | They require less maintenance. | They require less maintenance. |
| 4 | Native User Interface/ User Experience. | The User experience with hybrid applications are poorly comparative. | User experience is poorly comparative. |
| **5** | They have multiple codebases. | They have single codebase. | Have single codebase. |
| **6** | The playstore or app store can be used to update mobile native applications. | Mobile applications updates are centralized. | Mobile application updates are centralized. |

* What is a bridge (in React Native)?

**ANSWER: T**he layer known as the Bridge in React Native is what allows JavaScript and Native modules to communicate with one another. It serves largely as a carrier layer for non-parallel chronological grouped feedback signals from JavaScript to Native modules. Suppose we need reutilize few existing Java library without implementing it again in JavaScript. So, we can reutilize it in our React Native application with the use of Native Bridge. Sometimes, to build an application of production level, we will most likely need to use Native Bridge.

**Working of Bridge:** When we click on our app icon to launch it, the operating system creates the main thread (also known as a UI thread) and assigns it to our app. The JavaScript thread and the shadow thread are created by this main thread (also known as the shadow tree). Calculating layouts described on the JavaScript side and sending that information to the Native side is the responsibility of the shadow thread. JavaScript is used to lay out views, and the Shadow tree is used to compute them before routing them to the UI thread.

* Why do we develop for mobile platforms? What are some reasons you think we've done this, especially since computers already exist and are considered accessible for many?
  + Feel free to argue against the question or disagree - simply just make a few points why you think mobile development from off in our world.

**ANSWER:**

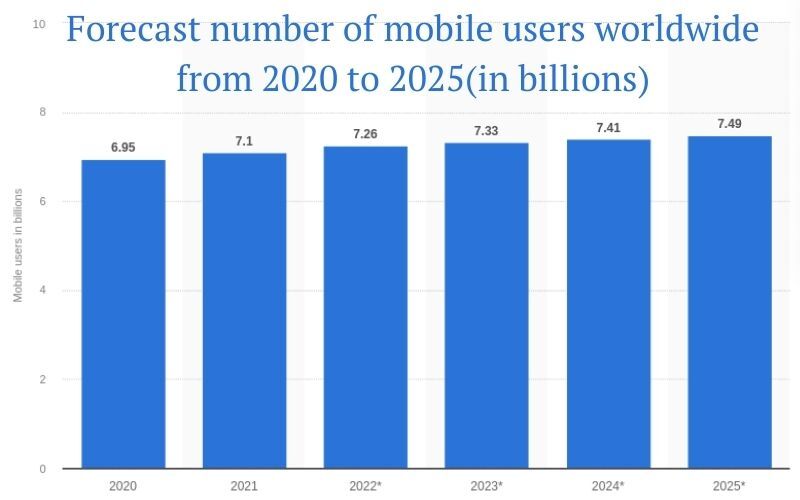
The creation of mobile applications has recently experienced tremendous growth. There are currently thought to be 2.3 million mobile app developers working hard to meet market demand.

In actuality, according to Apple, 1.25 million apps were registered in the Apple app store in 2013 and accounted for 50 billion downloads and $5 billion in developer payments. With these kinds of market statistics, it is easy to see why mobile application development is essential for a company's success.

**BENEFITS OF USING MOBILE APPLICATION**

* Increases brand visibility for business owners and their businesses.
* Increases accessibility since most smartphone users will want to access whatever information they want at a go.
* Global audience reach: People from everywhere can download the applications and use at a go.

As the statistics say:

Source: [Statista](https://www.statista.com/statistics/218984/number-of-global-mobile-users-since-2010/" \t "https://www.emizentech.com/blog/_blank)

* Why do we write our React Native code in JSX?
* Although not required, using JSX helps developing React applications simpler. There are various ways to accomplish the same thing.
* The expression language JSX supports is{}. Any JS expression or variable for React may be used as the expression. A parenthesis must be used to enclose a substantial chunk of HTML, as in ().
* JSX adheres to XML standards.
* JSX expressions transform into standard JavaScript function calls after compilation.
* HTML attribute names in JSX are written in camelcase. fontWeight in JSX, for instance, is the equivalent of font-weight in HTML.

What is JSX as well?

**ANSWER:** JSX is a JavaScript Extension Syntax used in React to easily write HTML and JavaScript together. JSX stands for Javascript Extended Markup Language

Basically, by using JSX you can write concise HTML/XML-like structures (e.g., DOM like tree structures) in the same file as you write JavaScript code, then Babel will transform these expressions into actual JavaScript code. Unlike the past, instead of putting JavaScript into HTML, JSX allows us to put HTML into JavaScript.

* Why do we create components?

ANSWER: Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but **works alone and return HTML**.

React collects and organizes the data passed to a given component as a single object. The name for data passed to a component, such as title, is ****props.**** All prop values exist within the function component itself on a props object.

* What's the point?
* To avoid cold repetition. DRY(Don’t Repeat Yourself).
* Why not just create the entire code logic in one file instead of what is effectively mini lego parts we're combining together?