**QUESTIONS**

1. What is the difference between ES5 & ES6?
2. What is the difference between React.js & React Native?
3. Please explain a promise lifecycle in JavaScript?
4. What is the difference between creating applications using React Native and using Next.js?
5. What is GraphQL and how to integrate it with React.js?

**ANSWERS**

1. ES6 is an improved version of the previous one (ES5). Elaborating enhancement features in ES6 could be the best answer to differ the two of them and the features the ES6 offers are:

* Block-scoped variables (let and const). Variables declared with let or const can only be accessible within the block where they are declared. The difference between these two is that the value of variables declared with let can be reassigned but const. However, variables declared with const with an object/array as its value can still be modified. Before let and const, developers use var to declare variables. It is globally-scoped (accessible for the entire function where they’re declared) and would be hoisted (readable even before they’re declared). Using var is not recommended since it has several ‘odd’ behaviors which could cause errors / bugs / unwanted results,
* Functions could have default parameter values rather than passing undefined values if there’s no argument passed,
* Class-based syntax could be used which has inheritance features,
* ES6 allows developers to organize source code into separate files (modules) which implements modular systems,
* Array and object destructuring features,
* Spread and rest operators to expand an array or objects,
* ES6 allows us to organize asynchronous operations in a clean and prettier way.

1. React.js (or React) is a JavaScript library while React Native is a framework built on top of JavaScript and React itself. The purpose of using React is that it allows developers to build user interfaces for the web, creating dynamic reusable UI components with its state features within it, and manipulating DOM to update some specific-preferred UI components efficiently using virtual DOM principle while the purpose of using React Native is to write codes using React.js syntax to build multi-platform mobile applications (of course, through compiling process).
2. A promise could exist in several cycle which are:

* Pending: promise will enter a pending state right after it is created which means the result of this operation will remain undefined yet until it is completely resolved by the async operation or rejected because of the failure occurred in the async operation,
* Fulfilled: in this cycle, the async function is finally completed without an issue and bring resolved value,
* Rejected: in this cycle, the promise brings rejected value due to the failure of the async function,
* Settled: this cycle refers to the final cycle of the promise after it has been fulfilled or rejected.

1. React Native and Next.js both are frameworks built on top of JavaScript and React.js (or React). The main difference is about the usage. React Native used to develop multi-platform mobile applications whilst Next.js used to develop either side-rendered or client-side rendered (could be both) web applications.
2. GraphQL is a query language for API developed by Meta (formerly Facebook) that enables clients to request only the data they need, and nothing more. It provides a more efficient way (reduces data transfer in order to improve performance), is powerful with a strong type system and simplified iteration, and provides more flexibility for clients to request data compared to traditional REST APIs. These are steps to integrate GraphQL with React project:

* First, we need to build GraphQL server. There are many packages we can use to build it and one of them is Apollo Server,
* Install Apollo Client in React project (client),
* Declare GraphQL queries and mutation (if needed) along with handling loading and error state.