1. **What is Single Page Application?**

Single page applications work within a single browser session and allow users to jump between parts of the system without requiring the entire page to reload.

1. **Difference between npm & npx ?**

* Node package Manager is a tool: used to install, maintain, uninstall javascript modules (packages)
* NPX: (Node package Execution) If we want directly run a package for once
* NPM is for management -vs- NPX is one time execution

1. **What is babel & Web-pack?**

* Bable is a transpiler & also a compiler.
* It will take ES6 input & converts it to ES5 for our browsers to understand.
* It also converts JSX (JavaScript Extended) language to JS.
* Web-pack is basically bundling tool used by react:
* It will bundle all the CSS & html files & will add to Index.html file. (Using <link> tags)

1. **Explain Class based components -vs- functional components?**

|  |  |
| --- | --- |
| **Functional Components** | **Class Components** |
| Functional components are written as a JavaScript function. | Class components are written as a JavaScript class. |
| Functional components do not have a state or lifecycle methods. | Class components have a state and can implement lifecycle methods like componentDidMount and componentDidUpdate. |
| **Performance** is Faster as they do not have state and lifecycle, react needs to do less work to render these components. | **Performance** is Slower as they have state and lifecycle, react needs to do comparatively more work to render these components. |
| Functional components codes tend to be shorter and more concise | Class components code require the boilerplate code, such as a constructor method and the use of “this” to access props and state. |

1. **What is Hooks? What are different types of Hooks?**

Hooks are functions that let you “hook into” React state and lifecycle features from function components. Hooks don't work inside classes — they let you use React without classes.

Types of Hooks are:

* ‘useState’: It is the most commonly used React Hook. It allows functional components to have state variables. It takes an initial state value as the argument and returns an array with two elements — the current state value and a function to update that state.
* ‘useEffect’: It enables performing side effects, such as data fetching, subscriptions, or DOM manipulations after the component has been rendered.
* ‘useRef’: Provides a way to create mutable references to values or DOM elements that persist across renders. Often used for accessing or manipulating DOM elements.