**ANSWERS**

**Q1) What is NPM used for?**

**Ans-** NPM stands for Node Package Manager and the purpose of which is managing dependencies i.e all the project dependencies can be specified inside package.json and when the project needs to be started, with npm install, the dependencies are picked and installed from package.json. There is also a provison to store the version of the dependencies, in order to prevent updates from breaking the project.

**Q2) What are the benefits of using SSR?**

**Ans-** SSR stannds for Server Side Rendering and in React this is used to improve the perceived page load time. That means when a request is sent by a browser for an URL, the node server immediately renders the page, instead of waiting for the dependencies to download which happens in non-SSR case. Then later when the scripts are completely downloaded the page is rendered again by attaching the event handlers to a perticular DOM elements but keeping the rendered DOM element intact.

**Q3) Can you provide a use case for SSR, other than performance gains?**

**Ans-** The other use case would be boosting user experiance. In case of SSR the blank page or screen is not shown during the time when browser is downloading and getting the dependencies ready, after sending requrest for an URL. The use case would be the loading time for the homepage of any e-commerece site like amazon.com, where the request is quickly served in the form of the full homepage instead of blank screen. And later after event handlers are attached and render completes.

**Q4) Please explain ReactJS’s effect hook, and what should you look out for when using the effect hook?**

**Ans-** useEffect Hook is one of state manegement library under React hooks. It is always called when page is rendered. So, in case of any event which is to be handled at the time of render, it is kept in the useEffect code block.

But at the same time we have to be careful enough and put the dependency list as a second parameter so, that it is called only when there is any change in the dependency list. In this way, the repeated unnecessary calling of the hook can be avoided in the backened. This problem usually arrises when useState hook is affecting a state change and triggering re-render which in turn calls the useEffect hook.

**Q5) What is state management?**

**Ans-** State mangement is a way to create communication and data sharing between the React components. It create a data structute (called as state) that represents the state of a component and which can even be changed based on a user action or system event.

The state here, is a JavaScript object that representes that part of a component which changes due to resultant action of user. This can be assumed as the memory of a component.

When a user performs an action in a React application, the component’s state changes which can be handled. The actual problem arrises when the application scales and due to this increased complexity, it becomes difficult to keep track of all the dependencies.

An example that can be taken up here in an e-commerce app where the adding an item to the cart action can influence other things like the change of cart component state value, adding th cart to the user cart history and product checkout. This is a small state change that is handled out of plenty of other things which comes with the scalability of the application.

Hence the state management libraries such as Hooks, Redux and Recoil are used to maintain the states.

**Q6) Can you provide some common use cases for handling state in a SPA?**

**Ans-** The simplest use case would be considering an application which handles the single toggle which changes the count value between 0 and 1. And this is handled using click event handler of a button. So the state here describes whether the count value is toggled on button click or not and this is represented by the vlaue of the state.