Q. What are the advantages of using React?

→ • Use of virtual Dom to improve efficiency:
React uses virtual Dom to render the view. As the name suggests, virtual dom is a virtual representation of the real Dom. Each time the data changes in a react app, a new virtual Dom gets created. Creating a virtual Dom is much faster than redering the UI inside the browser.

• Gentle learning curve :- React has a gentle learing curve when compared to frameworks like angular. Anyone with little knowledge of javascript can start building web applications using React.

• SEO friendly :- React allows developers to develop engaging user interfaces that can be easily navigated in various search engines.

• Reusable components :- React Uses component based architecture for developing applications. components are independent and reusable bits of code. These components can be shared across various applications having similar functionality.

Q. what are the limitation of React?

→ - React is not a full-blown framework as it is only a library.
- The components of React are numerous and will take time to fully grasp the benefits of all.

- It might be difficult for beginner programmers to understand React.
- coding might become complex as it will make use of inline templating and Jsx.

Q. What is useState() in React?

→ The useState() is a built-in React Hook that allows you for having state variables in functional components. It should be used when the DOM has something that is dynamically manipulating controlling.

Q. What are keys in React?

→ A key is a special string attribute that needs to be included when using lists of elements.

The React Way to render a List.

- Use Array.map
- Not a for loop
- Give each item a unique key
- Avoid using array index as the key.

Importance of keys -

- keys help react identify which element were added, changed or removed.
- keys should be given to array elements for providing a unique identity for each element.
- without keys, React does not understand the order or uniqueness of each element.

- with keys, React has an idea of which particular element was deleted, edited and added.
- keys are generally used for displaying a list of data coming from an API.

Q. What are the differences between functional, class components.

→ Before the introduction of Hooks in React, functional components were called stateless components and were behind class components on a feature basis. After the introduction of Hooks, functional components are equivalent to class components although functional components are the new trend, the react team insists on keeping class components in react. Therefore, it is important to know how these components differ.

3. Why was virtual Dom introduced?

→ Dom manipulation is an integral part of any web application, but dom manipulation is quite slow when compared to other operations in Javascript. The efficiency of the application gets affected when several Dom ~~or~~ manipulations are being done. Most Javascript frameworks update the entire Dom even when a small part

of the Dom changes.

e.g :- consider a list that is being rendered ~~How does virtual Dom work ?~~ inside the dom. If one of the items in the list changes, the entire lists gets rendered again instead of just rendering the item that was changed / updated. This is called inefficient updating.

To address the problem of inefficient updating, the react team introduced the concept of virtual Dom.

Date
Page

Q. What is React Hooks?

→ React Hooks are the built-in functions that permit developers for using the state & lifecycle methods within React components. These are newly added features made available in React 16.8 version. Each lifecycle of a component is having 3 phases which include mount, unmount & update. Along with that, components have properties & states. Hooks will allow using these methods by developers for improving the reuse of code with higher flexibility navigating the component tree.

Using Hook, all features of React can be used without writing class components. For eg, before React version 16.8, it required a class component for managing the state of a component. But now using the usestate hook, we can keep the state in a functional component.

Q. What are props in React?

→ The props in React are the inputs to a component of React. They can be single-valued or objects having a set of values that will be passed to components of React during creation by using a naming convention that almost looks similar to HTML-tag attributes. We can say that props are the data passed from a parent

component into a child component.
- The main purpose of props is to provide different component functionality such as:
- Passing custom data to the React component.
- Using through this.props.reactProp inside render() method of component.
- Triggering state changes.

Q. Explain React state & props.

→ Props are Immutable, Has better performance, can be passed to child components.

State are owned by its component, Locally scoped, writeable/mutable, has setState() method to modify properties, changes to state can be asynchronous, can only be passed as props.

React state - Every component in react has a built-in state object, which contains all the property values that belong to that component.

In other words, the state object controls the behaviour of a component. Any change in the property values of the state object leads to the re-rendering of the component.

Q. Explain about types of side effects in React component.

→ Two types :-

Effects without cleanup : This side effect will be used in useEffect which does not restrict the browser from screen update. It also improves the responsiveness of an application. A few common examples are network requests, Logging, manual Dom mutations, etc.

Effects with cleanup : Some of the Hook effects will require the cleanup after updating of Dom is done. for example - if you want to set up an external data soure subscription, it requires cleaning up the memory else there might be a problem of

Date .....
Page .....

memory leak. It is a known fact that React will carry out the cleanup of memory when the unmounting of components happens. But the effects will run for each render() method rather than for any specific method. Thus we can say that, before execution of the effects succeeding time the React will also cleanup effects from the preceding render.