React JS Application

1. **What React JS is ?**

**It is a library which used to build user interfaces and UI components and is developed by Facebook. It follows Component based Architecture which allows developers to re-use the UI components making complex web applications easier to manage and maintain. It is used to create SPA (Single Page Application).**

1. **How to setup React: Installation & Environment Setting**
2. **We have to download & install the Node.js**
3. **Using CRA(Create React App) command we can create the react application like**

**cmd>npx create-creact-app applicationname**

1. **Using npx instead of npm as npx will install all the required packages for the application.**
2. **Runing the react application using npm start command and seeing the output on** [**http://localhost:3000**](http://localhost:3000)

**React JS Components LifeCycle**

1. **Initialization Phase: This is the stage where the component is constructed with the given Props and default state. This is done in the constructor of a Component Class.**

**Methods in this phase:**

**Constructor**

1. **Mounting Phase: Mounting is the stage of rendering the JSX returned by the render method itself.**

**Methods in this phase:**

**Constructor**

**Static getDerivedStateProps**

**render()**

**componentDidMount()**

1. **Updation: As the name itself suggests that it is the phase in which application components will be updated as per the data or any even happening**

**Methods in this phase:**

**getDerivedStateFromProps**

**setState() Function**

**shouldComponentUpdate()**

**getSnapshotBeforeUpdate() Method**

**componentDidUpdate()**

1. **Unmounting: As the name itself suggests that it is the final step of the component lifecycle where the component is removed from the page.**

**Methods in this phase:**

**componentWillUnmount() function**

**React Hooks**

In our React Application, Hooks are resusable functions that provide access to state in application and they uses state and manage side effects

When to use Hooks in our application, so whenever we are working with functional components and later we need to add state to them then in that case we need to convert existing functional component to class level component, but now with the feature of hooks we can easily maintain state in functional component also by the help of hooks.

In react we have built in hooks which are as follow:

1. State hook: It uses useState which provide state variable with direct update access.
2. Context Hook: It uses useContext which makes possibility to access information without being passed as a prop.
3. Ref Hook: It uses useRef which contains the variable having information not used for rendering i.e DOM
4. Effect Hook: It uses useEffect which helps the components to sync with the system. It includes changes in browser DOM, networks and other libraries.