Antra assignment – Class components

1. What are some differences between class and functional components?
   - ⇨ Some of the differences between class and functional components can be mentioned below:
     - Class Components:
       - They are ES6 classes that extends from “React.Component” and have a “render” method that returns the JSX.
       - “this.state” and “this.setState()” is used to manage and update the states.
       - To handle side effects and other operations, they have built-in lifecycle methods like “componentDidMount”, “componentDidUpdate” and so on.
       - Class based components with construtors to handle props, and built in lifecycle methods rather than functions.
     - Functional components:
       - They are based on simple functions (typically arrow functions in modern times) which returns JSX. They support modern ES6+ syntax and the code is easier to read and maintain.
       - States and side effects are managed using lifecycle hooks such as “useState”, “useEffect”, and so on.
       - Lifecycle behavior is easier to manage than in class components with using hooks like “useEffect”, which replaces several methods in class components such as “componentDidMount”, “componentDidUpdate”, etc.

2. Explain what lifecycle is in a simple way. How do you manage it in class and functional components?
   - ⇨ The lifecycle in React can be referred to as the series of events that occurs from the moment a component is created and added to the DOM to any update that happens on it until to the point it is removed from the DOM.

     In class components, it is managed using in built lifecycle methods:

- Mounting, is when component is created and added to the DOM, “componentDidMount” method is used.
- Updating, is when component updates on response to change in prop or state, “componentDidUpdate” method is used.
- Unmounting, is when the component is removed from the DOM, “componentWillUnmount” is used.
- There are other methods which can be used to manage the lifecycle in class components.

In functional components, the lifecyle is managed using lifecycle hooks:

- Mounting, “useEffect” is used with an empty dependency array ‘[]’ which acts similar to “componentDidMount”.
- Updating, “useEffect” with the component names mentioned in the dependency array to watch update on reflects the same action as “componentDidUpdate”.
- Unmounting, we use the cleanup function in “useEffect” to act as “componentWillUnmount”.
- See, how useEffect, one single hook in functional components covers three to four methods in class components.

3. Explain immutability in one sentence.
   ⇨ Immutability can be explained as the process where we do not modify the original data but rather create a new copy with any changes we require.