16. REACT Best Practices

In this chapter, we will list React best practices, methods, and techniques that will help us stay consistent during the app development.

- **State** − The state should be avoided as much as possible. It is a good practice to centralize the state and pass it down the component tree as props. Whenever we have a group of components that need the same data, we should set a container element around them that will hold the state. Flux pattern is a nice way of handling the state in React apps.

- **PropTypes** − The PropTypes should always be defined. This will help is track all props in the app and it will also be useful for any developer working on the same project.

- **Render** − Most of the app's logic should be moved inside the render method. We should try to minimize logic in component lifecycle methods and move that logic in the render method. The less state and props we use, the cleaner the code will be. We should always make the state as simple as possible. If we need to calculate something from the state or props, we can do it inside the render method.

- **Composition** − React team suggests to use a single responsibility principle. This means that one component should only be responsible for one functionality. If some of the components have more than one functionality, we should refactor and create a new component for every functionality.

- **Higher Order Components (HOC)** − Former React versions offered mixins for handling reusable functionalities. Since mixins are now deprecated, one of the solutions is to use HOC.