1. Use a Pascal case or Upper -camel-case Example - SubmitForm
2. Single responsibility – use an abstract type of complex component.
3. Use the container pattern to separate concerns. Data loading logic is separate from presentation logic.
4. Use stateless functional components and they are predictable.
5. Stateless components are simple functional components without having a local state but remember there is a hook in react to add state behavior in functional components as well.
6. Stateful components can contain the state object and event handling function, and user actions as well.
7. Stateless components are pure in nature which does a very specific task.
8. Good pattern is one that consists of two or more components.
9. Anti-pattern is bad practice to use.
10. When modifying the prototype of an object the react component will inherit from the other objects.
11. Destructing props
12. React lends itself to a functional programming paradigm.
13. Use stateless components – to reuse across different projects.
14. Use lazy loading for images and videos with React LazyLoad Package.
15. Implement code splitting to split your app into smaller chunks with Webpack.
16. Minimize the number of HTTP requests with code splitting and tree shaking
17. Use performance analysis tools like React Profiler or Chrome DevTools to identify performance bottlenecks.
18. Compress your assets with Gzip or Brotli compression to reduce page load times.