**Assignment 05**

**Performance Tuning**

Name : Shreya Khambe

PRN : 21510025

Batch : T-2

Aim : To perform performance tuning on Assignments 3 and 4

Theory:

Time is a valued asset that no user will like to waste if your web application demands unnecessary patience for their attention. React is a popular framework that developers believe for rendering performance. This is why big names like LinkedIn, DropBox, CSDN, Baidu, etc. use React for their web applications.

However, React even renders many irrelevant components addressing such performance issues; developers can smartly and consciously ensure React performance optimization methods by considering some prime measures and implying them.

There are fine React performance optimization techniques which we recommend to overcome the costly DOM operations.They are :

1. Windowing or List Virtualization in React Applications

Many React applications having or displaying long lists usually bear performance issues. Before loading the app, the entire list will be rendered in the DOM, causing a UI lag and drastically affecting the React.js app performance.

One way of overcoming this bottleneck is by List Virtualization or Windowing. Here, instead of rendering the complete long list of components on the app screen, we allow only a restricted list of items to be rendered on the DOM as much is visible.

The two libraries available for windowing are react-window and react-virtualized; you will enable rendering a small subset of the extensive list on the app screen. Your react app performance will improvise.

2. Key Coordination for List Rendering

When working with lists in React, you can assign key attributes to an element that will help render the upcoming ist items.

Assign key attributes to elements

In the case of dynamic lists, if the developer has wrongly assigned component keys to list elements, it becomes useful for the user, making it a performance hindrance for React app. Here, the new list entry will automatically suggest the previous list entry, which is unnecessary.

You must assign a unique key value to your list component to solve this bottleneck. Thus, use Key={ } for your dynamic lists to boost React app performance.

3. Lazy loading Images in React

When your React application contains many images, there are high chances that your React app performance will degrade. It happens because the DOM will render all images altogether before displaying the user screen. Hence we suggest using Lazy loading images, which will wait until the turn of the image appears on the user screen and only render that particular image.

Lazy loading images prevents creating unnecessary DOM nodes, just like we discussed for windowing. The popular libraries used for lazy loading to boost React performance are react-lazyload and react-lazy-load-image-component.

Do you still think images are creating havoc loading your React Js application?

Hire React Developers from us and set back as you witness them enhance your React app performance

4. Functional Components & Component Interaction

The subtle way of optimizing performance of React applications is by using functional components. Though it sounds cliche, it is the most straightforward and proven tactic to build efficient and performant React applications speedily.

Our React experts have some advice when using Components.

Tech Advice: Keep it Short!

Our experienced React developers suggest keeping your components small because smaller components are easier to read, test, maintain, and reuse.

Some advantages of using React components like React Devtools (extension) are

🟠 Require less code,

🟠 Easy to understand,

🟠 Components are stateless,

🟠 Easy to test,

🟠 Flexibility to extract smaller components, and

🟠 Interactibility

5. Understand How to Handle 'this'

Functional components do not require ‘this’ binding, you might wish to use them whenever possible. But, if you are using ES6 binding, React will not auto-bind your functions within components. However, you may manually achieve the binding. Here are some ways to bind your components and functions:

🟠 Bind in render

🟠 Allow arrow function in render

🟠 Bind in constructor

🟠 Bind arrow function in the class property [Not in official ECMAscript]

6. Use a Function in 'SetState'

We recommend using a function and not an object in the setState function. This suggestion is because state changes aren’t implied immediately, as conveyed by React docs. Thus, instead of this:

Copy Text

this.setState({correctData: !this.state.correctData});

Use this way:

Copy Text

this.setState((prevState, props) => {

return {correctData: !prevState.correctData});

}

The above function will receive the previous state as its first argument, and the props at the time the update is applied, as teh second argument.

7. Utilize Prop-Types

Prop-types is a library for type checking of props. The below code snippet shows how you can import the function from the prop-type library:

Copy Text

import PropTypes from 'prop-types';

class Welcome extends Component {

render() {

return <h1>Hello, {this.props.name}</h1>;

}

}

Welcome.propTypes = {

name: PropTypes.string.isRequired

}

8. Trim Javascript Bundles

If you wish to eliminate code redundancy, learn to trim your Javascript packages. When you cut-off duplicates and unnecessary code, the possibility of your React app performance multiplies. You must analyze and determine bundled code.

9. Server-Side Rendering (SSR)

Try to use SSR consciously and check whether you actually need SEO or not for your application. SSR takes up an immense load, so if you avoid using it when not required, you will be blessed.

NextJS is the best among the available SSR. It is getting popular amongst developers & so is the usage of the NextJS-based React Admin Dashboard. NextJS integrated React admin templates can help you boost the development process with ease.

10. React Redux Optimization Tips

A well-known flaw faced by Yahoo is a classic example to consider when React apps are built with Redux. Indeed the combination is deadly and enables complex situations to structurize, but when you use Redux, your React app rerenders and slows down your performance.

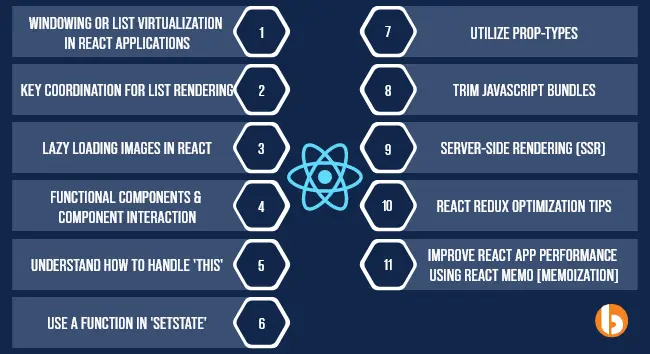
We are sharing two ways how you can overcome this challenge with React Redux applications. The first one is using RESELECT library when the higher-order components in your React app are allocated for rendering operations. Yahoo immensely benefitted by using this library.

Another method to optimize React Redux app performance is by using Immutable.js. The performance of an immutable list was much more (upto 4 times) than a mutable list. When you use mutable data structures in a Redux app, the Redux state tree consumes a lot of memory for copying data, hence hampering the app’s performance.

Using an immutable data structure will not update the original data and instead generate a new version of the updated data structure whenever requested. This technique improves the React performance drastically.

11. Improve React App Performance Using React Memo [Memoization]

Here, we would be covering on a basic overview of how to work with react memo for React app performance optimization.



Testing:  
Performance test for Assignment 03 can be viewed [here](https://pagespeed.web.dev/analysis/https-adsl-3-jsc-netlify-app/j5sm21kmog?form_factor=desktop) :

Performance test for Assignment 04 can be viewed [here](https://pagespeed.web.dev/analysis/https-adsl-3-jsc-netlify-app/j5sm21kmog?form_factor=desktop) :