PART TWO: THEORY QUESTIONS

1. What is a common mistake when using useEffect, and how would you avoid it?

Answer

A common mistake with useEffect is not managing dependencies properly. If any dependency is not included in the dependency array the effect can either run too often or not run when it should. To avoid this, I always ensure the dependency array contains all the variables the effect depends on.

2. In what scenarios would you use each, and how do they help with performance (useMemo & useCallback)?

Answer

useMemo is used to memoize the result of expensive calculations, ensuring that these computations only happen when necessary, reducing unnecessary re-renders while useCallback is used to memoize functions to prevent them from re-rendering unnecessarily.
I would use useMemo in scenarios where I am processing large datasets or performing heavy computations while I would use useCallback when I am passing functions as props to child components.
Both help to improve performance by limiting when React needs to re-calculate values or re-create functions.

3. When fetching data from apis in a react app, what best practices do you follow, especially around performance and user experience?

Answer

I make use of React Query for handling data fetching in my React Apps. It helps me to handle loading and error states, implement caching mechanisms and avoid unnecessary re-fetching, which improves performance.

PART THREE: OPINION QUESTIONS

In your opinion, what do you think is the biggest advantage of using React for modern web applications? Are there any scenarios where you might prefer a different frontend framework or library?

Answer

In my opinion, I think the biggest advantage of React is its component based architecture. I think so because this architecture promotes reusability which allows me to break down complex UIs into smaller and manageable components that can be used across different parts of an application.

Scenarios where other technologies might be preferred include:

1. Smaller and simpler projects: I can make use of Vuejs or Svelte because they are easy to learn and have a smaller bundle size compared to react.
2. SEO: Nuxtjs or Nextjs might be a preferred option because they offer Server Side Rendering which boosts SEO performance.
3. Very Complex Enterprise Apps: Angular, because of its dependency injection feature which provides additional scalability and maintainability.