● **When and why do we need lazy()?**

The lazy() function is a feature in React that allows us to load components dynamically, or lazily, only when they are needed . This can be beneficial for improving the performance and load times of our web application, especially if it contains a large number of components or if some components are rarely used.

Here's when and why we might need to use lazy():

1. Code Splitting and Reducing Initial Bundle Size
2. improved Performance
3. Faster Initial Load
4. Better User Experience

● **What is suspense?**

In React, Suspense is a feature that allows us to declaratively manage asynchronous data fetching and code-splitting in our applications. It is primarily used in combination with the lazy() function for dynamic imports and with the React.lazy() component to improve the user experience when loading data or components asynchronously.

● **Why we got this error : A component suspended while responding to synchronous input. This will cause the UI to be replaced with a loading indicator. To fix, updates that suspend should be wrapped with startTransition? How does suspense fix this error?**

This error is thrown as an Exception by React when the promise to dynamically import the lazy component is not yet resolved and the Component is expected to render in the meantime. If only the dynamic import is done and there is no <Suspense /> component then this error is shown.

React expects a Suspense boundary to be in place for showing a fallback prop until the promise is getting resolved. If showing the shimmer (loading indicator) is not desirable in some situations, then start Transition API can used to show the old UI while new UI is being prepared. React do this without having to delete or remove the Suspense component or its props from your code.

● **Advantages and disadvantages of using this code splitting pattern?**

Code splitting is a technique that splits an application's JavaScript bundle into smaller chunks, which are loaded dynamically as needed. This allows an application to load only the code it needs at a given time, and load other bundles on demand.

Code splitting can have many benefits, including:

Faster initial load time: By only loading the necessary code for the initial view, code splitting can significantly improve the time it takes for the page to load. This can be especially helpful on slower network connections or devices.

Improved user experience: Code splitting can allow users to interact with the application sooner, and non-essential code can be loaded asynchronously in the background to improve the overall responsiveness of the application.

Improved performance: Code splitting can reduce the amount of JavaScript that needs to be parsed and executed.

However, code splitting can also have some drawbacks, including:

Increased complexity in development and testing processes

More network requests that can affect performance

Additional code and dependencies that can increase the bundle size

● **When do we and why do we need suspense?**
Suspense is a React feature that allows developers to display a temporary UI while waiting for data to load. It's best used when you want to display a fallback while waiting for something to load, such as when waiting for data to be fetched from an API after the initial page load.

Suspense is often used in conjunction with React's dynamic import mechanism called lazy(). Lazy loading refers to the requirement that a component or portion of code will load only when it's needed. This functionality helps to minimize your application's loading speed and lower the initial bundle size.