Chapter 09 - Optimizing our App

Theory -

- When and why do we need lazy()?
- 1. Chunking ,
- 2. Code Splitting
- 3. Dynamic
- 4. Lazy Loading
- 5. dynamic loading

- What is suspense?
- React suspense lets you display a fallback component until you finish loading your component

- 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?
- We got this error because we were using a component that return promise using a lazy method , the jSX could is synchronous so it displays this error
- To fix this issue we can use a suspense to wrap our component with a fall back ui
- Advantages and disadvantages of using this code splitting pattern?
  Advantages of Code Splitting
  1. less bundles size and app render faster
  2. App optimization

  Disadvantages
  1. Code Complexity
- When do we and why do we need suspense?
  When to Use Suspense

  1. Fetching Data:
     - When you need to fetch data from an API and render a component only after the data has been loaded.

  2. Code Splitting:
     - When you are using dynamic imports to load components lazily, and you want to show a loading indicator while the component is being loaded.

  Why Use Suspense

  1. Improved User Experience:
     - Suspense provides a built-in way to show loading states, making your app feel smoother and more responsive to users.

  2. Declarative Data Fetching:

- Instead of managing loading states and errors imperatively, Suspense allows you to declare what should happen while your component is waiting for data or code to load, leading to cleaner and more maintainable code.

Coding -

- Create your custom hooks
- Try out lazy and suspense
- Make your code clean.

coding assignment

References:

- https://reactjs.org/docs/hooks-custom.html
- https://beta.reactjs.org/apis/react/lazy#suspense-for-code-splitting