Inefficiencies of Messy React

1. interface Props extends *BoxProps*

   BoxProps is neither imported nor inside the file itself.

2. Unused Variable

   The *children* variable is destructured from props but not used anywhere in the component. This is unnecessary and can be removed.

3. $useWalletBalances()$ undeclared

   The function stated here is neither imported nor inside the file itself.

4. $useState$, $useMemo$, and $useEffect$ are not imported

5. Hardcoded URL

   The URL for fetching prices is hardcoded within the *Datasource* class. It's better to make this URL configurable or pass it as a variable to make the code more flexible.

6. Error Handling

   In the $useEffect$ block, there's an error handler using console.err. It should be console.error.

7. Magic Numbers

   In the *getPriority* function, there are magic numbers (e.g., 100, 50) that determine the priority. These numbers are not explained, making the code less maintainable. It's better to use named constants or comments to explain the significance of these values.

8. Any Type Declarations

   Type declarations for *blockchain* is any. Properly typed code can help catch errors at compile time.

9. Usage of uninitialized $lhsPriority$

   This is a mistake where $lhsPriority$ is used, but never initialized. This can confuse the reader as well as we have $leftPriority$ on the next few lines.

10. Code is not written efficiently

    The $filter$ function and *sort* logic can be shorten to one line without nested if blocks. This makes it compute faster and makes it easier to read.

11. $formattedBalances$ is never used

12. $WalletRow$ is not imported

13. *classes* is not initialized