

Report for problem 3

Candidate: Le Anh Kiet

Contents

1/Problems:	1
1/Line 11:	1
2/Line 17:	1
3/Line 37 – 42:	2
4/Line 45 - 54:	2
5/Line 66 - 74:	3
6/Line 56 – 80:	4
II/Suggested fix:	4
1/Line 11:	4
2/Line 17:	4
3/Line 37 – 42:	5
4/Line 45 - 54:	5
5/Line 66 - 74:	5
6/Line 56 – 80:	6

1/Problems:

1/Line 11:

```
11 interface Props extends BoxProps {}
```

Issue: Anti pattern, the interface Props is unnecessary extends BoxProps due to it didn't add anything new.

2/Line 17:

```
17 const getPriority = (blockchain: any): number => {
18   switch (blockchain) {
19     case "Osmosis":
```

The variable blockchain is not clearly defined, which will cause issue if that variable is wrongly passed not as a string, after I analyze the code, I see that the blockchain should be a string.

3/Line 37 – 42:

```
35     return balances
36     .filter((balance: WalletBalance) => {
37         const balancePriority = getPriority(balance.blockchain);
38         if (lhsPriority > -99) {
39             if (balance.amount <= 0) {
40                 return true;
41             }
42         }
43         return false;
44     })
45     .sort((lhs: WalletBalance, rhs: WalletBalance) => {
```

The interface “WalletBalance” not have attribute “blockchain” and the variable “balancePriority” is not used. Beside that, “lhsPriority” is also not defined and maybe “balance.amount <= 0” is wrong logic. At here, I assume that the developer want to get the priority of the wallet, if it’s priority is valid then the program check if there is any money in that wallet or not, in this case, the logic for checking money is wrong.

4/Line 45 - 54:

```
45     .sort((lhs: WalletBalance, rhs: WalletBalance) => {
46         const leftPriority = getPriority(lhs.blockchain);
47         const rightPriority = getPriority(rhs.blockchain);
48         if (leftPriority > rightPriority) {
49             return -1;
50         } else if (rightPriority > leftPriority) {
51             return 1;
52         }
53     });
54     }, [balances, prices]);
```

This piece of code is missing a case is leftPriority == rightPriority and plus, at the line 54 we can see a dependency “prices” at here but it wasn’t used to calculate anything in this useMemo() so it will cause a performance issue is that it will rerender unnecessary whenever “prices” is changed

5/Line 66 - 74:

```
63     const rows = sortedBalances.map(  
64       (balance: FormattedWalletBalance, index: number) => {  
65         const usdValue = prices[balance.currency] * balance.amount;  
66         return (  
67           <WalletRow  
68             className={classes.row}  
69             key={index}  
70             amount={balance.amount}  
71             usdValue={usdValue}  
72             formattedAmount={balance.formatted}  
73           />  
74         );  
75       }  
76     );  
77  
78     return <div {...rest}>{rows}</div>;  
79   }  
80 }
```

This code is anti react pattern due to use index as key because the list is sorted so it will change the value of each index, at here we should choose the field "blockchain" since the hash blockchain will be unique

6/Line 56 – 80:

```
55
56     const formattedBalances = sortedBalances.map((balance: WalletBalance) => {
57         return {
58             ...balance,
59             formatted: balance.amount.toFixed(),
60         };
61     });
62
63     const rows = sortedBalances.map(
64         (balance: FormattedWalletBalance, index: number) => {
65             const usdValue = prices[balance.currency] * balance.amount;
66             return (
67                 <WalletRow
68                     className={classes.row}
69                     key={index}
70                     amount={balance.amount}
71                     usdValue={usdValue}
72                     formattedAmount={balance.formatted}
73                 />
74             );
75         }
76     );
77
78     return <div {...rest}>{rows}</div>;
79
80
```

“formattedBalances” is not used, in the “WalletRow” it also access the “balance.formatted” while the “WalletBalance” not have the field “formatted”.

II/Suggested fix:

1/Line 11:

```
10
11     interface Props extends BoxProps {}
12     const WalletPage: React.FC<Props> = (
```

Directly use the “BoxProps” instead of creating a new interface “Props”.

2/Line 17:

```
16
17     const getPriority = (blockchain: any): number => {
18         switch (blockchain) {
19             case "Osmosis":
20
```

Set the type of “blockchain” to string

3/Line 37 – 42:

```
35     return balances
36     .filter((balance: WalletBalance) => {
37         const balancePriority = getPriority(balance.blockchain);
38         if (lhsPriority > -99) {
39             if (balance.amount <= 0) {
40                 return true;
41             }
42         }
43         return false;
44     })
45     .sort((lhs: WalletBalance, rhs: WalletBalance) => {
```

Add the “blockchain” as a string into the interface “WalletBalance”, change lhsPriority into balancePriority, switch the balance.ammound from compare “<=” to compare “>=”

4/Line 45 - 54:

```
45     .sort((lhs: WalletBalance, rhs: WalletBalance) => {
46         const leftPriority = getPriority(lhs.blockchain);
47         const rightPriority = getPriority(rhs.blockchain);
48         if (leftPriority > rightPriority) {
49             return -1;
50         } else if (rightPriority > leftPriority) {
51             return 1;
52         }
53     });
54     }, [balances, prices]);
```

Add a new case “==” and delete the dependency “prices”

5/Line 66 - 74:

```
63     const rows = sortedBalances.map(
64         (balance: FormattedWalletBalance, index: number) => {
65             const usdValue = prices[balance.currency] * balance.amount;
66             return (
67                 <WalletRow
68                     className={classes.row}
69                     key={index}
70                     amount={balance.amount}
71                     usdValue={usdValue}
72                     formattedAmount={balance.formatted}
73                 />
74             );
75         }
76     );
77
78     return <div {...rest}>{rows}</div>;
```

Change the “key” from using index to use “blockchain”

6/Line 56 – 80

```
55
56   const formattedBalances = sortedBalances.map((balance: WalletBalance) => {
57     return {
58       ...balance,
59       formatted: balance.amount.toFixed(),
60     };
61   });
62
63   const rows = sortedBalances.map(
64     (balance: FormattedWalletBalance, index: number) => {
65       const usdValue = prices[balance.currency] * balance.amount;
66       return (
67         <WalletRow
68           className={classes.row}
69           key={index}
70           amount={balance.amount}
71           usdValue={usdValue}
72           formattedAmount={balance.formatted}
73         />
74       );
75     }
76   );
77
78   return <div {...rest}>{rows}</div>;
79
80
```

In this issue, I will assume that we need the formattedAmount so we will use the “formattedBalances” instead of “sortedBalances”

All the issues have been fixed in refactored.tsx

Thanks for your tests, have a nice day! :D