Computational inefficiencies and anti-patterns and how they were improved.

1. getPriority
   getPriority is called multiple times, which means it is re-invoked every time it is called. So I moved the values outside of WalletPage and used map to make the lookups more efficient.
2. sortedBalances, formatted Balances, rows
   Both these functions iterate over the balances array four times. I combined all the functions so that the code only iterates over balances once.

Minor bugs and issues

1. FormattedWalletBalanace is not used correctly. It appears that sortedBalances' balance type is WalletBalance. I changed this but left FormattedWalletBalanace incase it was used somewhere else.
2. console.error was called as console.err.
3. Sort returns null when leftPriority equals rightPriority.
4. lhsPriority was never defined. Assuming it was a typo and is infact balancePriority, returning -99 in getPriority and checking whether the value is more than -99 is unnecessary. Instead, just checking whether the blockchain exists should be enough.