**Summary:**

I have made quite a few comments on this code, which are primarily focused around stylistic choices. Don’t be disheartened by the amount of feedback, overall your logic is solid and you understand what you’re trying to achieve, which is awesome!

My changes will help make this more ‘human-readable and adopt some better practices.

I’ve linked some articles/documentation to bring home my points, if you’re already aware of them there is no need to read, just bear in mind for next time.

I’d like to review this again with you once the changes have been made to make sure that I didn’t miss anything.

If you’d like to go through any of my feedback or challenge anything, please reach out.

Good job!

**Line 7:**

There is a spelling mistake in your component name.

**Line 11:**

Rather than setting a new variable of fuel on line 12, we can destructure the value in the props with - ({.fuel })

Here is an article on destructuring and its benefits which could help -<https://medium.com/@gazzaazhari/destructuring-props-in-react-d8f163d3ef84>

**Line 13:**

The variable names for getter and setter of your useState hook are not very descriptive, and don’t convey their meaning.

I would consider renaming them to isOverFuelLimit, setIsOverFuelLimit - or some variation of this. This will make your intentions with the state easier to understand.

Also, rather than setting the initial state to 1 I would use a boolean of false. Further down on line 17 we are calculating a true/false value from the number set initially, therefore it makes more sense to booleanise the value from the beginning.

**Line 16:**

The number 1200 is not descriptive, could you replace with a variable with a name indicative of its value, like so: const FUEL\_LIMIT\_IN\_LITRES = 1200;

**Line 21:**

In these situations where we have a if/else decision to make, I like to use ternary operators -<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Conditional_Operator>

They are really clean and help improve readability.

Combined with the comments above it would look something like this:

Return (

{isOverFuelLimit ? <h2 style={{color:"red"}}>Alert</h2> : <h2>All is fine</h2>;}

)

**Line 28:**

You have gone from using Functional Components to Class Based, could you provide some clarity on why you believe a Class based component was necessary in this context?

Personally, I would have opted to continue using a Functional Component. But I would like you to challenge that if you feel strongly about it. If you don’t I would like this to be refactored to use Functional.

**Line 40:**

We set an interval here, and initiate on on line 43, which is fine. However, we don’t clear the interval when the component unmounts, this could lead to memory leaks.

See here for more information -<https://developer.mozilla.org/en-US/docs/Web/API/clearInterval>

Additionally, we’re using another non-descriptive number at the end of the interval, consider changing to a variable like - const ONE\_SECOND\_IN\_MS = 1000;

**Line 48/49:**

We’re using var to assign a variable, can you switch this to let to use the ES6 standard.

The variable name x1 is not descriptive, could you change it to something more appropriate.