**Entelegis ReactJS/Javascript Quiz #1: Shopping List**

**You should be able to complete this quiz in about 4hrs. I suggest you study with this video before taking the test:**

[**https://youtu.be/5Xew--ycx0o?list=PL55RiY5tL51oyA8euSROLjMFZbXaV7skS**](https://youtu.be/5Xew--ycx0o?list=PL55RiY5tL51oyA8euSROLjMFZbXaV7skS)

[**https://youtu.be/pgAvVxowaYU**](https://youtu.be/pgAvVxowaYU)

**The test requires you to build a shopping list application that you can take to the grocery store to remember what to purchase.**

1. auto generate a new react application called quiz. then if the App.js is a stateless function convert it to a stateful class component so you can get access to full state capability.

2. I'm testing your knowledge of state, application lifecycle and component reuse, some css by asking you to create an app to do the following:

- on start of app a shopping list will automatically have milk, eggs and bread added to it with qty of 1

no other components are visible on the page

- 3 seconds after startup make a button visible to the user to add new items to the list, and a button to clear list items. You will be using if conditions when building the UI to determine to show the buttons or not.

Use a material-ui hover action button for these buttons. see: <https://material-ui.com/components>

* If the clear items button is clicked then confirm with user you want to remove all items before doing it.

- since I like fruit we are only going to add fruit to the shopping list. On click of the add button randomly

choose a fruit from the list and display in the list with a randomized qty of between 1 and 5. Don’t worry if you’ve already added the fruit to the list. I like fruit I’ll take as many as you can give me.

fruitList = ["Bananas", "Pears", "Cherries","Apples","Peaches","Watermelon",”Pineapple”];

- Make use of re-usable components, create a ShoppingListItem class that you can use over and over

to add new items to the list. It’s default background color should be light blue. pass your shopping list item data as a property to be used by the component for rendering. Make use of Parent to child data passing and add a material-ui icon to delete the item from the list and let the parent app know and do the deletion for the item. Also add material-ui checkbox icon and checkbox outline icon and depending on the state of complete or not show the appropriate icon. Use a material-ui Grid component to evenly space out the checkbox icon, the list info and the delete icon.(see screenshots below). Clicking on the checkbox outline icon will complete the item in the the partent app, and the list item will show with a light green background. You can search for icons here: <https://material-ui.com/components/material-icons/>

- make use of array.map command to draw your list of items from a template

- make sure that, when you append new items to your shoppinglist array that you watch out for invalid direct state access and respect the immutability of state objects. Due to async nature you may have to find a solution with the state to handle unexpected behaviour appending to the array. Hint you need to check the previous state to chain the new state properly.

- use inline style (css) to add a border around the main list and some padding

- use external css (in App.css) to put a border with rounded corners around the list item and a bit of padding to make it look nice, with a light blue background

- remove any unused imports and make sure there are no warnings on the npm terminal when you run the app, also make sure there are no warnings when you hit F-12 in the browser and look under console. The code should run cleanly.

- add a launch.json for testing in chrome

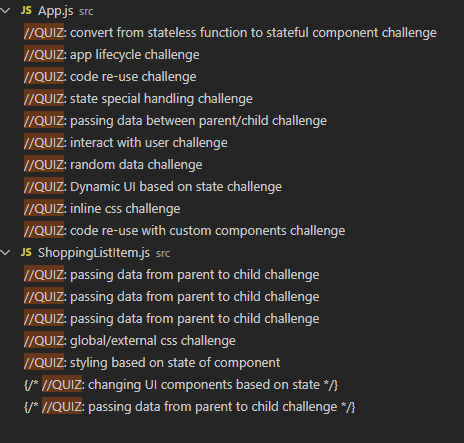
- use a little code as possible. Do add as many comments as you like.

3. i will be checking well formatted code, and comments. Use modified camel notation for

names of variables and functions (start with lowercase letter, each word in the variable name has starting

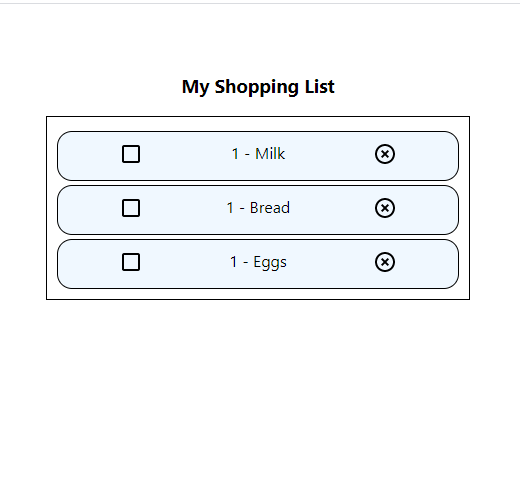
uppercase example: myVariableName. Classes are uppercase first letter such as ListItem class

All the challenges you will find in this quiz:

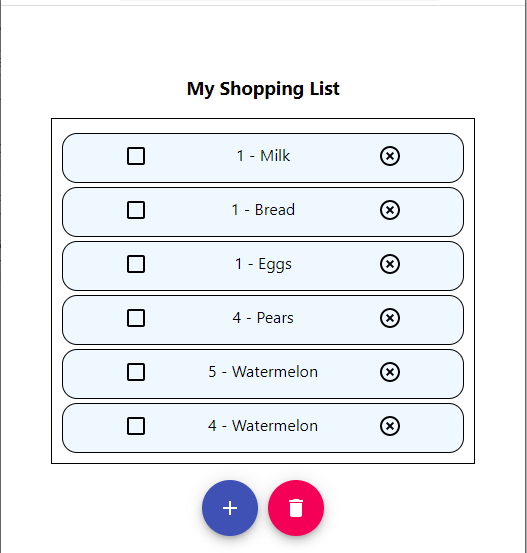


**Here is the completed UI from the app:**

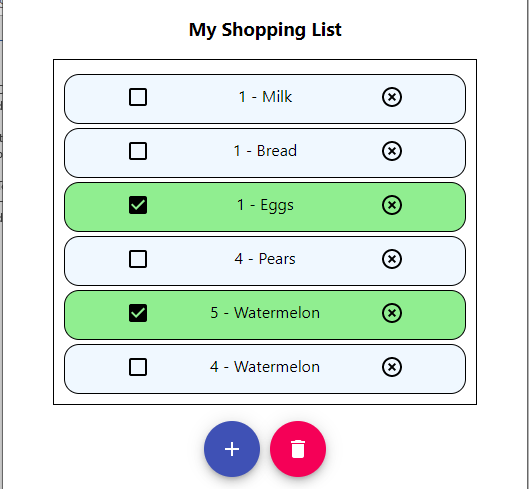
**Startup:**

****

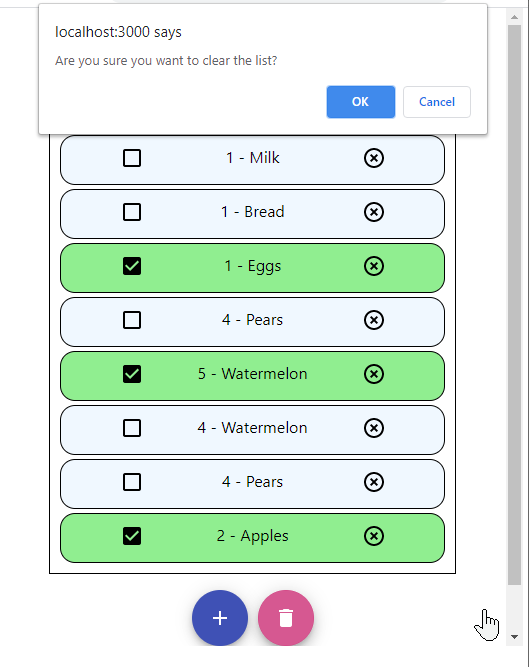
**3 seconds after startup, and after adding a few fruits:**



Completing some items:



Clearing the list:



No warnings or errors

