

Test for Frontend Developer

Module 1

Set up a React Router for our app. We should be able to do the following:

1. Click on the words "Pic Some" in the header to go to the "/" route, which should display the Photos component (found in the pages folder)
2. Click on the shopping cart icon in the header to go to the "/cart" route, which should display the Cart component (found in the pages folder)

Module 2

Set up the Context for our app.

1. In a new file, create a new context with React
2. In that same file, create a custom component that renders the Provider of the context you created
3. For now, just pass in an empty string "" as the context provider's value prop
4. Export the custom Provider component and the full context object (so we can pass it to the use Context hook eventually)
5. Set up your reduxreactindex.js to use the custom context Provider you created.
(You can wrap it as a parent of the Router component)

Module 3

Add state to our context and pass it through the Provider

1. Add state to hold the array of all photos our app gets from the API
2. Pass the array of all photos through the value of the provider so it's available anywhere the app accesses the context

Module 4

Get the JSON data with the photos information from the API and save it to context State.

1. As soon as the Context Provider component renders, get the JSON data from this url:

<https://raw.githubusercontent.com/bobziroll/scrimba-reactbootcampimages/master/images.json>

2. Save the array of data that comes back to state. Review data fetching in React using `fetch`:

Module 5

Using the data in context state, map over it in the Photos page and display the images.

On the Image component, track the hover state

1. Create state boolean for "hovered"
2. When the mouse enters the Image's div, set "hovered" to true
3. When the mouse leaves the Image's div, set "hovered" to false
4. Log "hovered" to the console so you know it's changing successfully.

Module 6

Conditionally render the heart and plus icons when an Image component is being hovered on

1. Icon to render for the heart:

```
<i className="ri-heart-line favorite"></i>
```

2. Icon to render for the plus:

```
<i className="ri-add-circle-line cart"></i>
```

Module 7

Add ability to toggle an image's `isFavorited` property by clicking the heart icon (filled heart doesn't need to display on the image yet)

1. Add a toggleFavorite method to context. It should take an `id` parameter and update the array of allPhotos by flipping the `isFavorited` property of the photo with the matching `id`
 - a. Have this function also console.log something so we know it's running correctly
 - b. Don't try to modify the individual image object only. Make sure to provide a whole new array to context with the one item with the matching `id` being changed.
2. Make it so clicking the heart icon on any given image runs this method

Module 8

Make the image display the filled heart icon when it's already favorited

Logic to follow:

- If the image is favorited, display the filled heart icon always
- Otherwise, if the image being hovered on, display the heart outline instead * If it isn't favorited OR hovered on, don't display anything

Remember to make it so clicking the filled heart icon ALSO runs the toggleFavorite function!

Filled heart icon:

```
<i className="ri-heart-fill favorite"></i>
```

Module 9

Add propTypes to the Image component

1. className should be a string
2. img should be an object, specifically an object with `id`, `url`, and `isFavorite` properties

a. Hint: Specifying the properties of an object is called an object's "shape"

<https://reactjs.org/docs/typechecking-with-proptypes.html#proptypes>

Module 10

Setup context to manage items in an array called `cartItems`. This will be an array of image objects.

1. Add the `cartItems` state to context. (Array)
2. Add function to add an image to the cart. (Takes the full image object as parameter)
3. Make it so clicking the plus icon on the image adds the item to the cart.
(Console.log the cart items array to see that it's working)

Module 11

Change the plus icon to a full shopping cart icon when an image is already in the cart. This should display whether the image is being hovered or not (like the favorite icon).

Icon to use when item already in cart:

```
<i className="ri-shopping-cart-fill cart"></i>
```

Hints:

https://developer.mozilla.org/enUS/docs/Web/JavaScript/Reference/Global_Objects/Array

/find

https://developer.mozilla.org/enUS/docs/Web/JavaScript/Reference/Global_Objects/Array

/some

<https://stackoverflow.com/a/8217584>

Module 12

Make it so clicking the little shopping cart icon on the image removes the item from the cart

Module 13

Change the cart icon in the header to display the full cart icon if there are any items in the cart

Full class name to use WHEN ITEMS IN CART:

"ri-shopping-cart-fill ri-fw ri-2x"

Full class name to use WHEN CART IS EMPTY:

"ri-shopping-cart-line ri-fw ri-2x"

Module 14

Make it so clicking the trash icon on the Cart page removes that item from the cart #
Challenge

Calculate the total cost of the items in the cart and display it on the Cart page

1. Usually the item in the database will have it's own cost saved, but we're assuming every item we sell costs \$5.99, so you can just hard code that cost in
2. To very easily display the total cost in US dollars (or whatever currency you want), use the following:

```
`<number>.toLocaleString("en-US", {style: "currency", currency: "USD"})`
```

https://developer.mozilla.org/enUS/docs/Web/JavaScript/Reference/Global_Objects/Number.prototype.toLocaleString

Module 15

Let our user place their order!

Clicking the "Place Order" button should:

1. Change the text to "Ordering..."
2. Timeout for 3 seconds (to simulate an order being placed)
3. Log "Order placed!" to the console
4. Empty out the cart

Module 16

1. Only render the "Place Order" button if there are items in the cart
2. Change the trash icon to a filled-in trash icon when hovering over it

Filled trash icon to use when hovering:

```
<i className="ri-delete-bin-fill"></i>
```

Empty trash icon to use when NOT hovering:

```
<i className="ri-delete-bin-line"></i>
```

: -----All the Best-----: