Monthly Test - July

Welcome to your first React monthly test, where you are going to build an eCommerce apparel application. It is going to be fully realistic, you are not going to fake any data, but really persist it on a database. The docs are available **here.** Now, don't panic by seeing that behemoth of endpoints. I will say which are important.

1. **Categories** => A particular group of products are present here
2. **Attributes** => Attributes related to a product, like size and color.
2. **Products** => All product related endpoints, like product by ID, post review.
3. **Shopping Cart** => Endpoint based on Shopping Cart, like add products, delete products, edit quantity for a particular product. Empty cart.
4. **Customer** => Auth related, like login, register, logout.
5. **Orders** => Order related, like view orders for a particular customer, create an order etc.

That's it. The evaluation would be around these endpoints. You don't have to worry about payments. Once you click checkout, just generate an order and say purchase successful, and make sure you include a button to return to home page.

What I would like you to do is craft a scalable, non buggy React-Redux app around these endpoints. I want the following conditions to be fulfilled.

1. Ability to view all products.
2. Ability to filter products based on category (French, Italian, etc).
3. Ability to view a particular product along with its attributes. (Size and color. You have to fetch a separate API request to the attributes endpoint)
4. Ability to log in, log out, register a user and also persist his/her auth credentials, so that he/she does not have to log in again and again.
5. Ability to add/update/delete an item to cart and update a particular product's quantity.
6. Ability to purchase items by generating orders.
7. Ability to visit the authenticated user's orders (Only when logged in).
8. Ability to have protected routes for non-authenticated users.
9. Ability to have smooth error handling. (Plain alerts are completely fine)

By the way, if you are not sure what the response object would contain, there is a section called **Models at the bottom,** where it explains the object structure. For example, the Category model suggests the response object properties for one particular category like French, Italian.

The design is not at all watched, so come up with Bootstrap, Material-UI, Semantic-UI, Cats and Dogs UI, if that existed lol. Basically, make sure that the functionality stands first. Come up with perfectly explaining action names, clean reducer arrangement, proper route management, route

guards, state changes, form handling. Trust me, this is going to be a perfect portfolio project if you hit it right.

ALL THE VERY BEST !!