Design a simple E-commerce web app with state management.

TASK 1

The web app must contain the following components-

1. Any 3 main categories
2. Cart service (can add as many items)
3. Order service
4. Mock a payment service
5. Simple User authentication (social logins not necessary)
6. Main page with all categories and listings
7. Products details page

You can use the following mock APIs for the backend -

1.https://mockoon.com/mock-samples/category/ecommerce/
2. https://fakestoreapi.com/

Feel free to use any other e-commerce backend - fake data etc.

Do not make a complex e-commerce web app. You can make as simple as a 6-10 products e-commerce app with 3 different categories.
You are free to use any kind of css styling framework.

TASK 2

Add the following capabilities to this e-commerce app -

1. State management library such as Redux / Flux / RxJS
2. PWA capabilities - such as offline viewing and caching

TASK 3

1. Unit tests
2. Integration testing of the web app

You can use any testing framework here.
Feel free to use as many NPM packages as you'd like.

The focus here is to show off your skills in React. Even though some concepts of React may not be applicable here in this web app, the idea is to gain an understanding of the React ecosystem and its fundamentals. Please feel free to justify why you have made certain decisions - for example if you chose to use the useReducer() hook over useState() hook.
Or if you chose to make a Higher order component, what was the need ?
Or for example if you choose to structure a component in a certain way, why was that ?
This would help to understand your thinking process better while building the web app.

Points to Keep in Mind / Points that will be judged:

1. Comments - sort of like documentation of the code you wrote, to easily understand
2. Focus only on programming and core React aspects
3. CSS styling is not judged - it is just for aesthetics

4. Coding style (naming conventions / function styles etc)
5. Component structure (class based vs function based / use of hooks)
6. Number of Lines of Code (less, the better)
7. DRY - (Don’t repeat yourself) code over WET (Write everything twice)
8. React principles used (HOCs / Pure components / Hooks / One way binding)

Time to complete: 5-6 Days.

Upload the code on a public github repository and please share the link to rahul@greenwaveit.in

For any other clarification regarding the assessment - please feel free to send an email to the above email address.