Brief - Realtime Coding Exercise: Typescript Contractor

Introduction

We will be working through the test with you, we do not expect you to complete it all, we are looking to understand your approach, coding styles and designs

Please complete the test on your own machine and in the same way that you would if you were doing it for work. Please have your IDE set up and ready to go with a blank project but do not start your implementation until we are all on the zoom call together.

We will be asking you to complete the task in Typescript.

Note: this test was originally written for Object Oriented languages. If you are completing this test in a language that lends itself well to functional programming, feel free to ignore the suggested Object Oriented interface and complete the test in a functional way.

Task

We are working with a client who wants to launch an e-commerce site to promote their brand, below are some of the products that they will be listing:

Products

ID	Name	Price
0001	Water Bottle	£24.95
0002	Hoodie	£65.00
0003	Sticker Set	£3.99

The marketing team would like to run the following promotions;

- If you spend over £75 then you get a 10% discount
- If you buy two or more water bottles then the price drops to £22.99 each

Multiple promotions can be applied to the same checkout

The checkout system needs to be able to scan the items in any order then apply the promotion rules. These rules should be allowed to change over time. The pseudocode below outlines the interface for the checkout

```
checkout = new checkout(promotionalRules)
checkout.scan(items) //List of item objects created from the products above
println(checkout.total()) //e.g. £68.99
```

Implement a checkout that meets these requirements.

Do not worry about the storage or retrieval of any items in a database, for the purpose of this test they only need to be processed.

Test Data

Items: 0001,0001,0002,0003

Total Price: £114.97

Items: 0001,0001,0001

Total Price: £68.97

Items: 0002,0002,0003

Total Price: £120.59