

Tech Test Full Stack Developer

Instructions

- We recommend the use of **test-driven development**, following the red, green, refactor steps;
- We encourage the use of one or more **design patterns** when developing your solution. This is your opportunity to show your skills, use it well;
- We are looking for the solution to be well factored and use good code principles like **SOLID**;
- Normally it takes around 5 hours to finish the test;

GitHub

- The test should be uploaded on your own GitHub
- When you start do a commit with the base project with the message starting tech test, so we can see only the changes that you made on the code in the following commits
- When you finish do a commit with a message finished tech test
- Commit as frequently as possible so we can see the development progress.
- Do not use branches
- After the challenge is completed please e-mail us with your personal Github information

The problem

You work for an IT company that has an e-commerce product. The e-commerce is a multitier solution built on the .NET platform, written in C# programming language using domain-driven design and good coding practices. The client application is a single-page web application in Angular and other open source technologies.

The Product Owner (PO) would like to create a prototype for a promotion feature that will be incorporated into the solution in the future.

In the current e-commerce version, the product entity has the following properties: Id, Name and Price. According to the provided documentation, the new feature should allow the e-commerce administrator to select a promotion for the product, which should be applied to the shopping cart depending on the promotion rules.

The following table of products was provided as an example:

Id	Name	Price	Promotion
1	Product A	€20	Buy 1 Get 1 Free
2	Product B	€4	3 for 10 Euro
3	Product C	€2	
4	Product D	€4	3 for 10 Euro

The PO has asked you to implement the promotions Buy 1 Get 1 Free and 3 for 10 Euro. He mentioned that in the future new promotions may emerge and they should be easily implemented and integrated into the solution.

As part of the prototype, you should implement a checkout service that receives the shopping cart items and calculates the total price of the shopping cart.

You should also implement a single-page web application in Angular 9 or later where the PO would be able to test the promotion feature based on the following requirements:

Shopping Cart

Given a static list of products, the PO should be able to create a shopping cart

- The PO should be able to add products to the shopping cart
 - When the added product has been previously added to the shopping cart it should increment the item quantity
- The PO should be able to remove products from the shopping cart
- The shopping cart should display the total price and applied promotions as per example below

Item	Quantity	Price	Total	Promotion Applied
Product A	2	€20	€20	Buy 1 Get 1 Free
Product B	3	€4	€10	3 for 10 Euro
Product C	5	€2	€10	
Product D	2	€4	€8	
		Total	€48	

Buy 1 Get 1 Free

In the shopping cart, for the products with Buy 1 Get 1 Free promotion

- The promotion should not be applied when the item quantity is 1
- The promotion should be applied once when the item quantity is 2
- The promotion should be applied once when the item quantity is 3
- The promotion should be applied twice when the item quantity is 4
- Any other quantity is allowed following the above logic

3 for 10 Euro

In the shopping cart, for the products with 3 for 10 Euro promotion

- The promotion should not be applied when the item quantity is less than 3
- The promotion should be applied once when the item quantity is 3
- The promotion should be applied once when the item quantity is 4
- The promotion should be applied twice when the item quantity is 6
- Any other quantity is allowed following the above logic