Front-end challenge

Challenge overview

We will be building a simple e-commerce application.
The project is primarily focused on front-end development, for that purpose, the products data is defined in the **products.json** file. This is the data that must be used to display and manage information throughout the application.

General requirements and functionality overview

We will be developing the app in steps as shown below:

1. **Products overview**
   a. On **HOME** page we need to show a list of product cards containing the product title, product desc and available quantity along with the quantity of the of the products in the cart.
   b. When the user taps on one of the cards he is sent to the details page where other than the product title and description, we have to show an image of the product.
   c. On **DETAILS** page we have a button - Add to cart and when the user taps on the button he is sent back to the **HOME** page where the total amount is increased by the amount of the last product we added in the cart.
   d. If we come back to a product that has been already added to the cart, we disable the button Add to cart on the details page.
2. **Products filter**
   a. On **HOME** page we need to implement product filter. The filter will consist of
      i. **a search bar** - where we filter products depending on the input string
      ii. **filter by amount** – slider from price range (ex. Show products form 10 € to 20 €)
      iii. **order alphabetically** – descending or ascending
3. **Cart**
   a. On CART page we have overview of the items we put into where we can remove them from cart. Also, we have the total amount + shipping details and a button for checkout (which at this point will only simulate that the payment was successful after which we show a notification for successfully order, we send the user back to HOME page and we clear the cart)

NOTE: Since for this challenge we are not working with a back-end for persistence purposes you will need to use **web storage** which web applications use to store data locally within the user's browser