Writing Automated E2E UI test suites

# Description

This part of the technical assessment is to write automated end-2-end test suites using a Framework of your choice. For this we have provided you with a web application that you need to setup. Your test framework will work against this setup and running web application.

# Instructions to get the web application up and running

* Ensure you have Docker desktop installed on your system

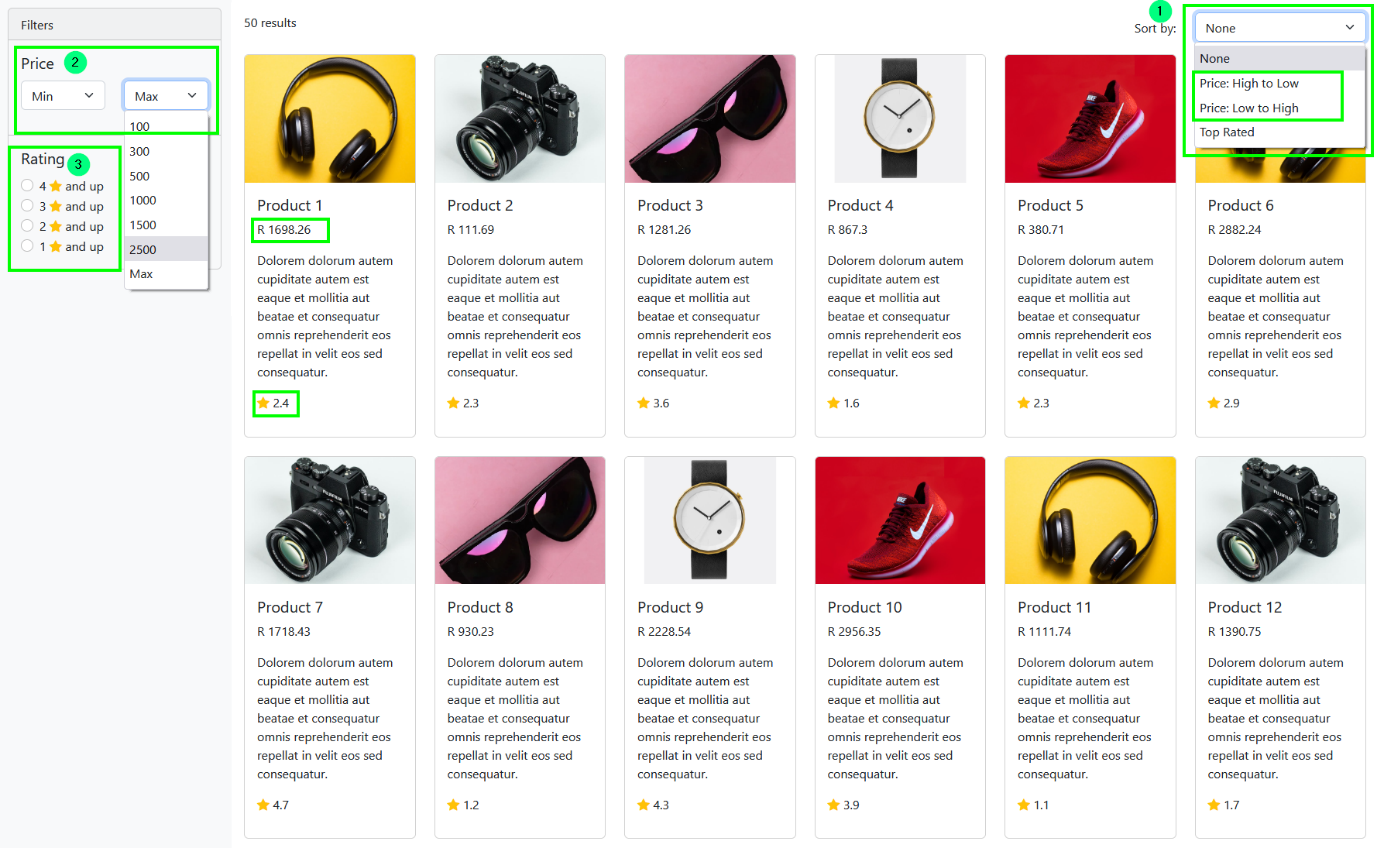
(available here: <https://www.docker.com/products/docker-desktop/>)

* Unzip the zip file to a location on your computer
* In the root of the folder, view the README.MD file for further instructions
* You will need to run the 4 docker commands from the project root folder as shown in the README.MD file
* Once all steps have been completed, then you should be able to access the web application via the following URL: <http://localhost/>

**Please see the next page for further instructions!**

# Once you’re up and running

When visiting <http://localhost/> you should see the following UI:



The UI is a typical E-Commerce web application that displays a list of products. Each product contains the following information:

* Image thumbnail
* Product name
* Product price
* Product description
* Rating star including average rating

The UI contains the following features (as shown by numbers in the UI) to allow the user to find products and drill down:

1. **Sort by filter**: This allows the user to sort the product list in certain ways. Selecting an option refreshes the page and applies the chosen sorting method. The sorting function includes:
   1. **None** (i.e. don’t sort)
   2. **Price high to low*:*** this sorts the displayed products in price descending order, i.e. the most expensive product first and the cheapest last.
   3. **Price low to high*:*** this sorts the displayed products in price ascending order, i.e. the cheapest product first and the most expensive last.
   4. **Rating*:*** this sorts the displayed products in rating descending order, i.e. the most highly rated product first and the least rated product last.
2. **Price filter**: This allows the user to filter out products based on price. The user can choose a minimum price, a maximum price or BOTH. Selecting either or both refreshes the page and applies the chosen price filter. For example, choosing the minimum price R1000 would hide all products < R1000. Choosing the maximum price R2000 would hide all products whose price is > R2000. Choosing both would apply both rules.
3. **Rating filter**: This allows the user to filter out products by user ratings. Selecting an option refreshes the page and applies the rating filter.

Example:

* Choosing *4 stars and up*, would only show products that have a rating of 4 stars or more.
* Choosing *3 stars and up*, would only show products that have a rating of 3 stars or more.

# Automated E2E Testing task

1. Write an automated end-2-end UI test suite that would fully test:
   1. the **Sort by: Price High to Low** feature.
   2. the **Sort by: Price Low to High** feature.
2. Write an automated end-2-end UI test suite that would fully test the **Price Filter** feature.
3. Bonus: if you still have time, write an automated end-2-end UI test suite that would fully test the **Rating Filter** feature.

**NOTE: please be sure to provide full instructions on how to run the automated test suite.**