# TASK DESCRIPTION

## Contents

## Synopsys

Your goal is to create minimalistic e-commerce website. Implementation is expected as set of static HTML files with responsive layout with (RWD techniques, Fluid layout). You should add necessary basic JS (you may also use utility libraries)

There is list of extra miles that are not required to implement, but will be a bonus when implemented.

# Pages to make

⬚ Homepage (`1_start`)

⬚ Catalog (`2_catalog`)

⬚ Item details page (`3_item`) (Tablet use same layout as desktop one)

⬚ Shopping bag (`4_shopping_bag`)


## JavaScript

### Homepage

⬚ Carousel at the top of Homepage should scroll across static images.

⬚ Click on images in carousel should navigate user to Catalog or Item Detail Page

### Item details page

⬚ Click on buttons in Size and Color sections should highlight it. Only one button per section can be selected.

⬚ Click on "Add to bag" item should update items count and total price in header. Items count should be increased by one, total price should be increased by an arbitrary value. Proper cart implementation is listed in Extra miles

Shopping bag
☐ Click on "Clear bag" link should remove all product items from Shopping bag and replace them with text "Your shopping bag is empty. Use Catalog to add new items"

☐ Click on "Buy now" should remove all product items from Shopping bag and replace them with text "Thank you for your purchase"

☐ In both cases listed above: remove items count and total price from header.

# Extra miles
This is list of bonus tasks that are not required to implement, but will give you a bonus if implemented.

## Shopping bag (full shopping bag functionality with JS)
☐ Create one more detailed item page (so you will be able to add different items to shopping bag)

☐ Create persistent storage for Shopping bag data so it is saved when user navigated across the site (use LocalStorage)

☐ Click on "Add to bag" button should add item to persistent shopping cart storage
o  If two items in bag has same name, size and color - they should be combined on Shopping bag page. Quantity field in shopping bag item should reflect number of similar items.
o  If two items in bag has same name, but size and/or color - they shouldn't be combined.

☐ Every time when user click on "Add to bag" button - items count and total price in header should be recalculated.

☐ Every time when user click on "Remove item" - quantity for this item should be decreased by one. If quantity is equal to 1 and user click on "Remove item" - item should be totally removed from the Shopping bag. Items count and total price (in header and on the shopping bag page) should be recalculated.

You can create and use your own stub/mock data

# Browser support

Your implementation should be compatible with latest versions of following browsers:

▢ Chrome

▢ Firefox

▢ Edge

▢ IE11

# Tools

▢ Code editor of your choice

▢ Chrome DevTools

▢ Adobe Photoshop

# List of fonts

Since templates may contain proprietary fonts - use free alternatives from list below

▢ Helvetica Neue - Roboto

▢ Futura PT – Open Sans

▢ Times - Crimson

# Recommended file structure

```
css/
fonts/
img/
js/
|- common/
| |- jquery.js
| \- ... .js
|
|- start.js
\- ... .js
catalog.html
start.html
... .html
```