Milestones for Web Developer Project 5

Build an e-Commerce website

In this document, you'll find a set of milestones to follow that will help you deliver your project.

General Recommendations

You'll be writing JS code which will be divided into various functions. The idea isn't to comment on each and every line of code but to comment on things in a more general way. You can comment on the code, function by function, thanks to JSDoc. If you need help with this take a look at the article Comment's Usage and Best Practices in JavaScript.

Milestone #1 : Taking over the HTML/CSS mockups

5% progress

Before you get started you should look through the various resources you have, especially the different web pages that have been set up.

🎯 **Once this step is completed you will have:**

- Gained knowledge of how the HTML/CSS pages work and how they are structured.

📌 **Recommendations:**

- Display the 4 HTML pages provided in your browser.
- Try to display the HTML elements which Frank has commented on in the code in order to try to imagine what the final version will look like. Before you dynamically insert the HTML elements into the DOM using JS, you need to know which elements to insert.

⚠️ **Issues to be aware of:**

- Ensure you analyse the HTML/CSS pages properly. Everything has been sorted out for you, there's no point recreating elements that he has already created.

Milestone #2: Working with the API

10% progress

Before you go any further you need to look through the various resources you have, notably the API you will have to work with for this project.

🎯 **Once this step is completed you will have:**

- Gained knowledge of how the API works and how it is structured.

📌 **Recommendations:**

- Take the time to read the README on the GitHub repo!
- Once the API has been launched, look at the URL in the functional and technical specifications for the project in order to ensure that it is working properly. The idea is that you should see, in your browser, the JSON being sent back by the API when you ask the API for all of the products or for just one product (using its ID).

⚠️ **Issues to be aware of:**

- Make sure the API has been correctly launched before you try to use it.

Milestone #3: Inserting the products into the homepage

20% progress

You can now insert all of the products from the API into the website.

🎯 **Once this step is completed you will have:**

- A homepage containing all of the products from the API

📌 **Recommendations:**

- In the JavaScript code, start by asking for all of the products from the API, collect the response and then take a look at it, inserting each element (each product) into the homepage (the DOM).

⚠️ **Issues to be aware of:**

- Please note that we are trying to display the products dynamically, not statically.

- Remember to use all of the necessary elements for each product, it would be easy to forget to use the alternative texts, for example.

📃 **Resources:**

- Communicating with a web service via an API - How to Make a GET Request end POST Request in Javascript.
- The chapter Use the right loop to repeat tasks from the course Learn programming with JavaScript should help you when looking through the response sent by the API.
- You may also find some interesting information in the chapter Edit the DOM of the course Writing JavaScript for the web.

Milestone #4: Making links between the products on the homepage and on the product page

30 % progress

Before you move on to the product page you should think about what you need to do on the homepage to ensure that you know which of the different products from the API you need to display on the product page.

🎯 **Once this step is completed you will have:**

- The ability to open a product page as long as the page knows which product it should display.

📌 **Recommendations:**

- Research the term URLSearchParams. It is thanks to this concept that your product page will “know” which of the products from the API it should display.
- For each of the products from the homepage you will have to set up the “a” tag and the “href” property.

⚠️ **Issues to be aware of:**

- Ensure that you use URLSearchParams to transfer the ID from one page to another, not LocalStorage.

📃 **Resources:**

- Here is a short, clear article about URLSearchParams: How to collect URL parameters in JavaScript
- The MDN documentation on this topic.

Milestone #5: Collecting the ID of a product you wish to display

35% progress

Before displaying the details of a product you will need to know which product we are talking about. To do this you will need to collect the ID of the product that was selected on the homepage.

🎯 **Once this step is completed you will have:**

- Knowledge of which product you should display on the product page.

📌 **Recommendations:**

- You are now aware of which of the products from the API we are going to display on the product page. You will then have to collect the relevant product ID from the URL (URLSearchParams).

📃 **Resources:**

- As mentioned for the previous milestone the article on URLSearchParams will be useful: How to collect URL parameters in JavaScript.

Milestone #6: Inserting a product and its details into a product page

45% progress

We now have the ID of the product we need to display, meaning that we can send a request to the API in order to collect the required information about this product.

🎯 **Once this step is completed you will have:**

- A complete product page based on the API data.

📌 **Recommendations:**

- Send a request to the API to collect the product data

- Insert these details into the product page (into the DOM).

⚠️ **Issues to be aware of:**

- Make sure you use the API correctly, the idea being to collect just one product at a time and not several.

📃 **Resources:**

- You may refer to the functional and technical specifications for this project to learn how to send a request to the API.

Milestone #7: Adding products to the cart

55% progress

The product page is in place, displaying the details of a product that was selected on the homepage. You now need to deal with the option of adding this product to the cart.

🎯 **Once this step is completed you will have:**

- The ability to add products to the cart.

📌 **Recommendations:**

- From a technical perspective the cart can be made as an array containing three things: - the product ID, - the quantity of the product, - the colour of the product.
- You must use LocalStorage to access this array from the product page.
- When you add a product to the cart for the first time, add a new element to the array.
- When you add the same product to the cart (the same product ID and the same colour), increase the quantity of the product in the array.

⚠️ **Issues to be aware of:**

- In LocalStorage take care to avoid having several of the same elements for no reason

📃 **Resources:**

- You should read this article about LocalStorage. You may also refer to this MDN document on this topic: Window.localStorage.

Milestone #8: Displaying a recap table of purchases on the cart page

65% progress

The products have been added to the cart but the user has not yet seen this. During this step we will display the cart's contents on the cart page.

🎯 **Once this step is completed you will have:**

- A cart page which displays all previously-added articles.

📌 **Recommendations:**

- From the cart page you should collect the cart (the array) via LocalStorage.
- Have a look at the array.
- Create and insert the elements on the cart page.

⚠️ **Issues to be aware of:**

- Make sure you don't create any duplicates of the various elements in the recap table (the cart). If there are several identical products (same ID + same colour) they should be listed on the same row in the table.

📃 **Resources:**

- Again the chapter Use the right loop to repeat tasks from the course Learn programming with JavaScript should help you.

Milestone #9: Dealing with any modifications or removals of products on the cart page

75% progress

The products that are in the cart are displayed on the cart page. You now need to allow users to modify the quantity of product selected or to completely remove a product from the cart.

🎯 **Once this step is completed you will have:**

- The ability, on the cart page, to edit the quantity of product or to completely remove a product

📌 **Recommendations:**

- As regards modifications, you will have to use change event (addEventListener, change event) to enable changes in quantities of product.
- Furthermore, the Element.closest() method should allow you to target the product you want to delete (or modify the quantity) thanks to its ID and its color.

⚠️ **Issues to be aware of:**

- Make sure you modify both the DOM and LocalStorage, if you forget the latter then modifications made in the cart will not be saved when a user moves to a different web page or refreshes the page.

📃 **Resources:**

- This documentation from MDN on the dataset property could be of use.
- Here is an article about the use of addEventListener.
- And, last but not least, here is an MDN document on the Element.closest() method.

Milestone #10: Confirming the order

85% progress

We're almost done. Users should now be able to confirm their order, that is the objective of this milestone.

🎯 **Once this step is completed you will have:**

- The ability, on the cart page, to enter your contact details and to confirm the order.

📌 **Recommendations:**

- Collect and analyse the data entered by the user.
- Display an error message if necessary (such as if a user writes "hello" in the email field).
- Create a contact object (based on the data from the form) and a product table.

⚠️ **Issues to be aware of:**

- Make sure you closely check the data entered by the user
- When using regex to check the data make sure that you have carried out tests to ensure that regex is working properly
- Don't forget to display an error message if necessary

📃 **Resources:**

- Communicating with a web service via an API - How to Make a GET Request end POST Request in Javascript.
- This article on Regex should help you better understand how to check the data entered by a user. Regex can sometimes be difficult to write so feel free to research how to do this (e.g. "regular expressions javascript email").
- You could take a look at the functional and technical specifications for the project to learn how to send a request to the API.

Milestone #11: Displaying the order number

90% progress

We're getting towards the end of this process. Now that we can make an order we need to be able to display the order number.

🎯 **Once this step is completed you will have:**

- The ability to display an order number on the screen for a completed order

📌 **Recommendations:**

- Send a POST request to the API in order to collect the order ID.
- Redirect the user to the confirmation page by moving the order ID into the URL to display the order number.
- This number should be displayed on the screen but it must not be stored.

⚠️ **Issues to be aware of:**

- Make sure you double check the request sent to the API, we cannot tolerate any mistakes with the request.
- The number should be displayed on the screen but it must not be stored.

Milestone #12: Implementing the plan for the acceptance tests

100% progress

The website has now been put together so it is time to implement a test plan in order to check that all the features have been correctly set up and that they are working properly.

🎯 **Once this step is completed you will have:**

- A completed plan for the acceptance tests.

📌 **Recommendations:**

- Now that all of the JS code has been written the plan for the acceptance tests needs to be put in place. The main objective of this plan is to check that what was outlined in the functional specifications is reflected in the finished product.

⚠️ **Issues to be aware of:**

- Make sure that you don't forget any features when writing the tests.

📃 **Resources:**

- You can find a lot of information online about the various tests, especially about acceptance tests. Here is an article about acceptance tests.

🎉 Project complete!

Tell us what you think! This document is a new resource we give you to help you with your project. Take this 3-minute survey to give us your feedback so that we can improve it!