eCommerce with Bootstrap: Fruits & Veggies

Authors: Abu Sayeed, Guilherme Carra, Jorge Garcia, Sandra Mitjans

Summary

Description **3**
- Objective 3
- Fruits & Veggies 3

Requirements **4**
- Basic requirements 4
- Project features requirements 4
  - Project 4
  - User operations 5
  - Extra step 6
  - Bootstrap 6

JSON Structure **8**

Incidents Record **10**

Lessons learned **15**

Git Strategy **16**

Daily overview **17**

Description

Objective

This project objective is to put in practice the knowledge learned from HTML, CSS, JS & GIT during the master using the Bootstrap framework to create the structure and design of the entire project.

Fruits & Veggies

This project is based on an online store of fruits, vegetables, juices and more where customers can place orders and administrators can create new products, categories and other administrators.

Requirements

Basic requirements

- It is essential that your code is well documented
- All code including comments needs to be written in English
- Use a camelCase code style
- Never use inline styles in HTML
- Delete unused files and code
- The web application must be responsive
- Organize CSS files in a clear and orderly manner

Project features requirements

Before starting the development you will have to take into account the following sections in which the project requirements will be specified, the minimum requirements when using Bootstrap and the way in which you must distribute the work in your team.

Project

The project will be divided into two main parts, the **client side** and the **managers backend side**. The two parts will have to take into account the following characteristics:

- **Product**
    - Title
    - Image
    - Price
    - Description
    - Quantity
    - Category

        - The product can belong to more than one category

User operations

The user must be able to perform the following operations:

- **Client**
    - See all products
    - Search a specific product
    - Filter products by category
    - Add a product to the shopping cart
    - Purchase a product
        - Before buying the product you will have to go through a small purchase process where the user will have to fill in the following information **(this part is not present in the wireframes, you are free to implement it however you want)**:
            - Name
            - Surname
            - Email
            - Phone number
            - Address
            - Postal code
            - Province
            - City
            - Country
- **Manager**
    - Login into manager backend side
        - To access the manager side login you will have to write "/admin" to your URL.
            - Example: “www.myecommerce.com/admin”
    - In the backend side you will have to create pages that managers can use to:

    - You must implement all the input elements in the wireframes for each of the following screens:
        - Create products
        - Update products
        - List all products
            - Clicking on a products **Edit** button will allow you to edit the product
        - Create categories
        - Update a category
        - List all managers
            - Clicking on a managers **Edit** button will allow you to edit the managers details
        - Create a new manager
        - Update a manager

Extra step

The optional part of the project is detailed below:

- Add weight to products
    - Depending on the weight of the product, different shipping costs must be applied
    - There will have to be at least three cases of different weights that will apply a different price to the product as shipping costs
- Depending on the postal code entered, the shipping price will change

Bootstrap

When using **Bootstrap** in your project, you must take into account the following **requirements** that you must implement **as a minimum**:

- You will have to make use of the following **design elements**:
    - Containers
    - Rows

    - Columns
        - Use of columns specifying the width of each column with a number
        - Use of columns using a specific breakpoint
- You must use at least **5 bootstrap components**
- Besides using the **media queries** that Bootstrap provides you by default, you will have to create **three new custom ones**

JSON Structure

```
json = {
        products: [
        {
                id: 1,
                title: 'Watermelon',
                img: 'src/img/watermelon.jpg',
                price: 4.65,
                description: 'Whole watermelon about 6kg',
                stockQty: 1000,
                category: ['Fruits','Organics']
        },
        {
                id: 2,
                title: 'Apricot',
                img: 'src/img/apricot.jpg',
                price: 1.8,
                description: 'Basket of apricots 6 units',
                stockQty: 1000,
                category: ['Fruits'],
        },
        {
                id: 3,
                title: 'Melon',
                img: 'src/img/melon.jpg',
                price: 4.0,
                description: 'Whole melon about 1.5kg',
                stockQty: 1000,
                category: ['Fruits'],
}],
```

```
  categories: [
    { id: 1, name: 'Fruits', color: 'blue' },
    { id: 2, name: 'Vegetables', color: 'green' },
    { id: 3, name: 'Juices', color: 'orange' },
    { id: 4, name: 'Organics', color: 'yellow' },
  ],
  admins: [
    { id: 1, name: 'admin', surname: 'administrator', email: 'master', password: '1234', }
  ]
}
```

Incidents Record

- Is JSON Data Structure too complex?

Solution: the JSON structure was discussed and agreed by all group members. The initial proposal was reviewed and slightly adjusted.

Lessons: sharing your thoughts with the rest of team members to explain the solution you came up to and listen to them about possible improvements/changes.

- Requirements: “Use only one HTML file”, but to access the admin page the URL is “myecommerce.com**/admin**”. Isn’t it a different .html file?

Solution: apparently that was a mistake in the project statement. Asked during peer helping for clarification and concluded that there were two different HTML files necessary.

Lessons: ask as many questions/doubts as possible during peer helping lessons to clarify every aspect of the project.

- Hints on how to start with Bootstrap?

Solution: Reading the Bootstrap's documentation we found out that all the framework's features are very well explained with good examples.

Lessons: Bootstrap's documentation is a good source to learn how to use it.

- Problems with size when trying to add cart icon (svg) into the header section

Solution: Bootstrap's SVGs have ***"width="***..." and ***height="..."*** attributes to adjust its size, plus while working with grids you can adjust its size with the class ***.col-x***.

Lessons: Working with grid cols and SVG properties.

- What is the best way to dynamically create Bootstrap elements with Javascript?

Solution: Creating string variables with the HTML code of the specific Bootstrap's element you're going to use. So you can change values dynamically with jQuery ***.attr()*** function, for example.

Lessons: Variable with strings can be really useful when you want to maintain a Bootstrap element pattern and print them with a JS function.

- How to insert margin/padding with Bootstrap?

Solution: Using the class ***".m"*** on a HTML element. With a combination you can set margin on any side you want, for example: ***.mx-4*** inserts left and right margin with a size of 4.

Lessons: Search on the framework documentation what you need and how Bootstraps magins works and their size from 0 to 4.

- How to enlarge a drop-down menu to make the Cart Open button?

Solution: If you want to change the size of a drop-down menu you have to apply your own styles on the CSS

Lessons: Bootstrap does not offer an option to change the size of a drop-down menu.

- How to close a Bootstrap Modal through a custom javascript function?

Solution: There is a specific function for this: **$("#modalName").modal("hide");**

Lessons: Bootstrap has a variety of javascript functions to work with its elements.

- Default form validation from Bootstrap doesn't fit our needs. Additional libraries or plugins such as "Validator.js" are available but still it is easier to code a custom function.

Solution: Bootstrap offers by default some classes to style the forms when valid/invalid, but the validation itself is based on the *checkValidity()* function or the validation made by the browser (very basic, such as fields not empty). We had to create our own custom functions to validate the different forms and combine them with the styling provided by Bootstrap.

Lessons: Bootstrap has its own limitations. In the case of form validations it is not so powerful/useful.

- Responsive design for a table.

Solution: we had to investigate how Bootstrap faces the responsiveness of tables (by adding a horizontal scroll to them and adjusting the table size to the available space).

Lessons: adapt/make use of the solutions (styles) provided by Bootstrap to avoid spending extra time on it.

- Toggleable sidebar causing size of the main window to mismatch the footer.

Solution: creating a custom style in CSS for the footer to re-adjust the width of the element depending on whether the sidebar is deployed or not.

Lessons: double check the effects/implications in the rest of the elements of the page when re-using/importing a complete element to your project.

- Making smaller images to fit on ***"col"*** (thumbnails)

Solution: We used ***".thumbnail"*** class that makes the image responsive and fits images in the ***"col"*** width.

Lessons: Generally when you insert a tag <img> on a "***col"*** it takes the space equal to the image size, but when you use ***".thumbnail"*** class it makes it more responsive and fits on the ***"col***" size.

- Animations jQuery (fade in, slide toggle) interfering with Bootstrap.

Solution: apparently it is not possible to use jQuery animation functions such as *fadeIn*/*fadeOut* or *toggle*. However, Bootstrap seems to have resolved the issue by adding data-attributes to the elements you want to animate.

Options

Options can be passed via data attributes or JavaScript. For data attributes, append the option name to `data-`, as in `data-animation=""`.

Name	Type	Default	Description
animation	boolean	true	Apply a CSS fade transition to the toast
autohide	boolean	true	Auto hide the toast
delay	number	`500`	Delay hiding the toast (ms)

Lessons: official documentation provided by Bootstrap is very complete and provides answers for common situations that developers face in their daily work. It is worth having a deeper look for examples of use and/or alternative solutions.

- Prevent refreshing page when a Bootstrap validation form is filled correctly

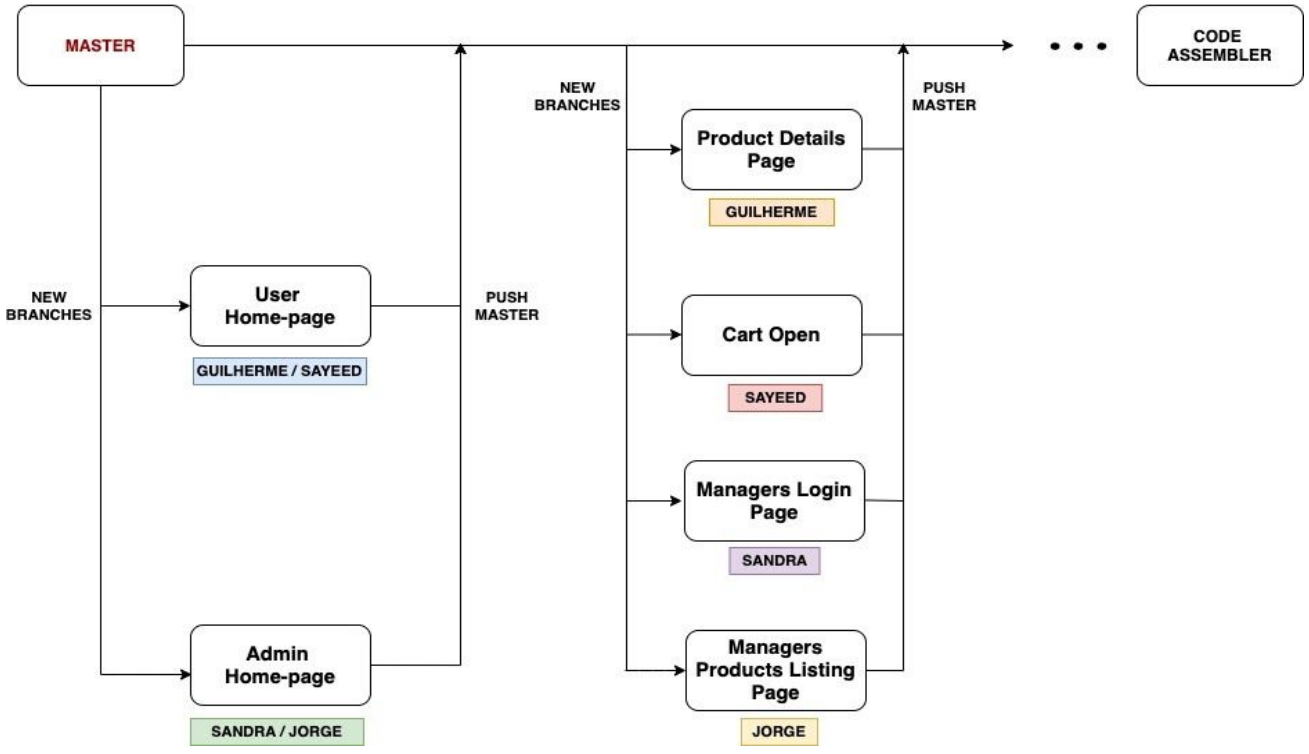
Solution: Modifying the validate function provided by Bootstrap to use ***preventDefault()*** when a form is correctly filled too.

Lessons: Bootstrap provides your own javascript code to validate a form fill, but it uses ***preventDefault()*** only when there's an error when filling a form. So we can change some functions to make it work as you need.

Lessons learned

- Understanding Bootstrap! How does its class work to change the HTML structure without writing too much CSS at the beginning.
- Form validation with bootstrap.
- Communication with your colleague while working on Git in different branches.
- Set an objective for the day and Review after the works.
- Creating complex HTML structure dynamically from javascript using jQuery
- Always need to be so attentive with HTML otherwise if you miss a single line while you are pulling code from your colleagues update it can give you a lot of stress when you are working with bootstrap.
- Setting regular follow-up meetings as a good practice to discuss the progress of the project and give support or take counter-measures in case of issues.
- Importing/re-using complete modules or elements from Bootstrap, such as a sidebar, buttons, tables, modals, dropdown menus,...

Git Strategy



- Team was divide between User side / Admin side

- At first to work on the look of the main pages we worked with pairs, after this each team member was assigned to work on individual pages on a new branch with the name of this page section.

- Even though we have started as principle we worked on four different branches but later on to avoid too much complexity we have added a few extra branches so we could merge easily when all the functionality works well.

- Later we found out that we needed a "Preview" branch to merge all the finished work and take a look at how the project was before adding it to the Master branch.

Daily overview

Day 1 - June 16, 2020
● Git strategy definition ● Products category definition ● Git repository creation ● To-do list creation ● Basic file tree creation (folders, html, js…)

Day 2 - June 17, 2020
● Team reunion to decide fonts, colors and how to work with Bootstrap ● **User Home-Page** creation (pending category filter, search bar filter) ● **Product Details Page** finished ● **Cart Open Page** starting creation ● **Admin Login Page** creation (pending login validation)

Day 3 - June 18, 2020
● **New Product Page** creation (pending validation) ● **New Category Page** creation (pending validation) ● **New Admin Page** creation (pending validation) ● Login validation completed ● Bug in responsive design for tables + sidebar corrected ● Table design in Bootstrap for Products, Categories and Admins ● Append functions to insert items in the different tables ● Created **Shopping cart** by Bootstraps and added functionality ● Starting to work on responsive **User home-page**

Day 4 - June 19, 2020
• **New Category** validation completed • **Edit category** within category list completed • **New Admin Page** validation started • We have merged all the Branches • **Cart Modal** finished • **Registration page** finished • **Ending order page** finished • Working on responsive design • Working on colors styles • **Happy Weekend!!!** ☀ 😃😊🥳

Extra (Jorge) - June 20, 2020
• **New Product** validation completed • **Edit Product** within product list completed

Day 5 - June 22, 2020
• New Admin validation completed (by Sandra on June 21st) • Edit Admin within admins list completed • Responsive design of **user pages** completed • **All pages** styles finished

Day 6 - June 23, 2020
• Presentation scheduled at 11:30h