PHP & FORMS

Christian Callau
Guilherme Carra
Carlota Gallart
AlejandroPalomes

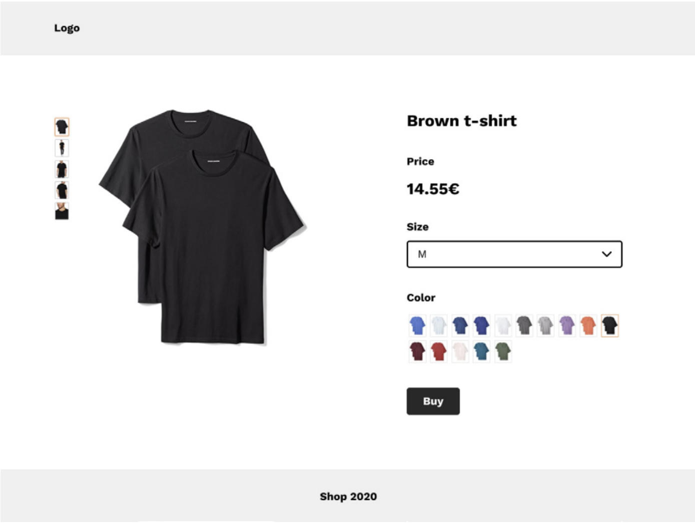
INTRODUCTION

In this project we have learnt to work with PHP and HTML forms. The idea was to understand how to send information and files from the client (browser) side to the server (PHP) by sending forms.

For this project we had to start from a previous project. In that project we had to create a form and validate it with JS, but all the information entered was not stored. The purpose of this project is to apply an improvement and use PHP to validate the data and store the information in a JSON file.

GENERAL ANALYSIS

The main objective of this project is that to apply a control of each record that is generated sectioned by the steps of the form, in this way the process can be recovered by the current step even if the user has closed the browser window.



Brown t-shirt

Price

14.55€

Size

M

Color

Previous project

REQUIREMENTS

- You must use GIT
- You must use the PHP v7
- Create a clear and orderly directory structure
- Both the code and the comments must be written in English
- Use the camelCase code style to define variables and functions
- In the case of using HTML, never use inline styles
- In the case of using different programming languages
- always define the implementation in separate terms
- Remember that it is important to divide the tasks into several sub-tasks so that in this way you can associate each particular step of the construction with a specific commit
- You should try as much as possible that the commits and the planned tasks are the same
- Delete files that are not used or are not necessary to evaluate the project

APPROACH

Fase 1

In this first fase, the tasks were divided as follows, related to different sections of the webpage:

Home Page

- Load data from JSON
- Design
- Print on screen the products
- Listener to redirect to product page
- Send id through url

Product Page

- Get data (id) from url (GET)
- Fill all data

Refactor Validate

- Send info with POST
- Recover info
- Validate info
- Send info + current step
- Send response to JS
- Receive response and proceed with form

Save Data

- Receive data
- Store data in json
- Check for the user
- Check if the data exists and update
- Store data
- Store step of form

Admin Section

- Login (admins.json)
- Modify products (products.json)
- Modify product
- Delete product
- Add product
- Modify Admins (create/delete)

Fix Errors

- No max time
- Placeholder contraseña
- Radio buttons
- General design
- Progress bar (center)

Fase 2

After diving tasks into subtasks, each team member chose which group would like to start with.

Once everyone had a role, we created together the structure of the JSON files, so everyone had in mind how it was build and how to acess it's values.

The tasks were interchangeable, meaning that if some team member has a block, he or she can swap the task with some other group member.

APPROACH

Fase 3

In this third stage, everyone started developing their tasks, working on its own branch to prevent confilicts between them.

There was a methodology to follow, regarding to organization:
Each morning, from 9.00 to 9.30, each team member should send a message in our Teams Chat, explaining three different points:

- Achievements done the previous day. Explaining all the tasks successfully completed.
- Objectvies of the current day. Detailing the tasks that were supposed to be finished at the end of the day.
- Blocks. Explaining all the blocks that may be occuring during the development of the tasks. These blokcks can be from different origins, like problems understanding the code, the need of some code from a collegue, or even lack of motivation.

After the team lead has reviewed those mesages, everyone joins a meeting to discuss each case. To clarify any doubt that may appear during the task development, or to find a solution to the existing blocks.

Fase 4

Finally, anytime a task is completed, a new merge is done to develop. To finally merge that branch to master.

INCIDENTS RECORD

Day 1

There weren’t major incidents during the first day. After tasks organization, documentation reading, information searched, etc. There wasn’t much time to start programming. This day was reserved to develop the json files structure.

Day 2

On Monday, some problems appeard to the members working presentially. For some reason, Xampp and Apache were having troubles executing the server, so after clearing cache and running some test, we finally could start working.

Some other problems emerged with design, specifically with the carousel in Homepage, but nothing really important, solved quite quickly.

Finally, after some blocks with coding, one team member decided to change her approach towards the code, restructuring some previous work. Something time consuming that delayed some task deliveries.

Day 3

On Tuesday, one member was travelling, so there was little communication with him for solving problems or reviewing some work. He finally joined in the afternoon, catching up with the tasks untill the night.

Another major problem emerged, regarding the inherited code from our classmates. It could be related to the lack of knowledge at that time, the could was quite a mess, having some inline styles, or other bad practices we should be avoiding right now. As one of the requierements is not to have those code blocks, we spent a lot of time to remove them.

Day 4

As on Tuesday, the problem related to fix the errors on the code we were given, consumed us a lot of time, blocking our progress in our own tasks.

One team member continous struggling with its tasks, so finally, another team member works side by side to fullfill the

INCIDENTS RECORD

tasks.

Beyond that, no other major problems happened that day.

Day 5

In this last day, the inherited code problems and it’s time consuming workflow, lead to the impossibility to finish one of the tasks and requirement of the project.

But not only this has been the problem, also the incomunation of one team member to alert about this situation to all the other classmates ended in this unfinished feature.

Everyone was focused on their tasks, aiming to finish the project by the end of the day.

Obviously, some conflicts happened when merging the new features to the actual deveop branch, but they were easy to resolve.

TASKS RECORD

Day 1

Carlota

- JSON structure

Christian

- JSON structure

Guilherme

- JSON structure

Alex

- Tasks organization
- JSON structure

Day 2

Carlota

- AdminValidations PHP
- Log in/ Sign out
- Products (Add/Update)
- Users(Add/Update)

Christian

- Send info with POST (JS)
- Validate info (PHP)
- Send response to JS (PHP)
- Receive response and proceed with form (JS)

Guilherme

- Pass “id” variable to JS with get-product.php and eliminate “id” property from catalog.json
- Create home-page HTML structure

Alex

- 404 Page
- Back Soldier

TASKS RECORD

Day 3

Carlota

- Finish validation new user.
- Validation of new products.
- Function of search bar of products and admins.

files .php of update products and admins.

Christian

- Send info with POST
- Validate info
- Send response to JS
- Receive response and proceed with form

Guilherme

- Help Carlota with Admin
- Work on home-page style
- Work on product-page first step style
- Fix placeholders

Alex

- Readme.md
- PDF documentation
- 404 page

Day 4

Carlota

- Finish validation new product

Christian

- Store form data in leads.json
- Load from data from lead.json if any

Guilherme

- Eliminate all HTML inline styles
- Fix buy timeout function (major problem)
- Insert a footer on home-page

Alex

- PDF documentation
- Review code
- Back Soldier

TASKS RECORD

Day 5

Carlota/Guilherme

- Add one to the admin and product ID
- Print the validation data in the update table (user and product)
- Delete function (product and user)

Alex

- Finish PDF layout
- Merge final commits
- Review code for requirements
- Pass daily reports from paper to computer

Christian

- Finish the tasks from day before

Click me for the project!

BIBLIOGRAPHY

Tasks documentation
https://docs.google.com/spreadsheets/d/1BtFZ-3Ar6A66T-mhf8xs7KoNs4eY8W2hiMG_JobY4vM/edit?usp=sharing

Project documentation
https://docs.google.com/document/d/1VTzosGrXqj-jv8LpiTNrgy2X9pSmid0M74WCfoVRfno/edit#