Milestone #2

For this second part of the project, we decided to help us with some of the functionalities that frameworks can offer us, mainly Bootstrap, for several reasons;

Functionalities that would otherwise be inaccessible

In addition, bootstrap allows us to access very practical features such as setting up a hamburger menu that would normally require the use of JQuery or JavaScript, languages that we don't know for the moment.

Considerable time saving

With a little research the functionalities of Bootstrap are relatively simple and quick to set up, considering the short time we had to do the whole of our site in CSS.
To this, it should be pointed out that the functionalities provided by Bootstrap are responsive most of the time. It is therefore a considerable time saving that we manage to make, a not insignificant detail compared to the short deadlines imposed on us.

A double-edged sword

The use of Bootstrap therefore opens up more possibilities in the creation of our site and gives us the possibility to get a result both more faithful to our idea, and in a shorter amount of time.

But Bootstrap is a double-edged sword because even if it can be quick to set up, if the given result does not suit us or has some defects, we can quickly have to spend several hours to fix details. This is why we wanted to limit its use only to the bare essentials and focus on using CSS if our knowledge allowed us to obtain a result similar to our expectations.

Finally, we mainly used Bootstrap in the creation of our bar navigation. We wanted to display every links on big screens. But when visited on a mobile or a tablet, we wanted the navigation bar to have a hamburger menu because there was not enough space to display every links.

Here is the navbar declaration:

```
<nav class="navbar navbar-expand-lg navbar-light main-nav fixed top">
```

It creates a light navbar that will expand itself on larger screens and fix it to the top of our page.
The hamburger button is the following (without the icon):

```
<button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent">
```

It toggles the visibility of the element with the navbarSupportedContent id.
The navbar is divided in 3 parts: the left links (with the Man.ga link), the centered search bar and the right links (mostly login, register or profile links).
They all have these classes:

```
nav navbar-nav flex-fill w-100 flex-nowrap
```

They are declared as navbar navigation blocks, with a full width. By using flexbox, we ask them to expand as much as possible without wrapping.
The centered and right parts have one more class: justify-content-center and justify-content-end respectively, so they are correctly positioned.

Finally, using a framework helps when building a complex component yourself is complicated. However, it adds a huge overhead, and – in our opinion – makes the code harder to understand. That's why we mainly used pure CSS, except for this navbar.