Description of Tasks for SPRINT 2

SPRINT 2 consists of three parts to be developed, each of which is described in detail below:

1. The first part consists of including the final content that the website to be developed will have dynamically using vanilla JavaScript. To do this, this final content must be included in a JSON file that should be properly structured as if it were a database since it is assumed that this JSON file should be generated by exporting the content from the database. This JSON file including the data to be loaded on the website will be (for now) stored locally along with the rest of the HTML, CSS, and JavaScript files of the website, but it should be taken into account that, later on, it could be obtained remotely from a server. An option to simulate this could be to use a "fake server" like JSON-Server. In this file, in addition to the general content of the website, we could also include the different users who will be validated and authenticated on the validation form implemented for this purpose on the website. Another option for generating this JSON file with the website content (including users) could be to use Strapi to generate the backend in a more comfortable way and use its API to access the data using the included API. In this part of the Sprint, the potential of using templates to generate the different pages that the website will contain by dynamically loading their content from the JSON file using JavaScript will be seen. The use of JavaScript will also allow us to generate these website pages using a composition of more basic templates (for example, a reusable header or footer) that add up to generate the final template and then fill it with the appropriate content to obtain the final web page.

2. The second part involves including at least one form on one of the website's pages and performing its validation on the client-side using HTML5's native validation support. A mandatory use case for forms is one that allows registering and authenticating users so that we can, for example, use these users later within the website to show different content depending on whether the user is authenticated or not, or if authenticated, differentiate the content depending on the current role of the authenticated user.

3. The third and last part involves modifying the current templates by applying RWD (Responsive Web Design) to adapt them to the characteristics of the different devices from which the website to be developed may be accessed. To do this, at least three variants of the original design of each of the existing templates must be made. These three minimum required designs must be adapted so that their content is displayed properly on large-size devices (for example, desktop and laptop), medium-size devices (for example, tablet in landscape mode), and small-size devices (for example, mobile). For this adaptation to RWD of the current templates, floating elements, element positioning, fluid or flexible layouts, media queries, and grid systems can be used. To verify that the designs adapt correctly, we should decrease or increase the size of the browser window until reaching the different defined breakpoints and then check that the design changes appropriately and as expected.