G3B-3 **2023**

Rapport de projet
Slave narrative

QUINVEROS Jordan

Sommaire **2**

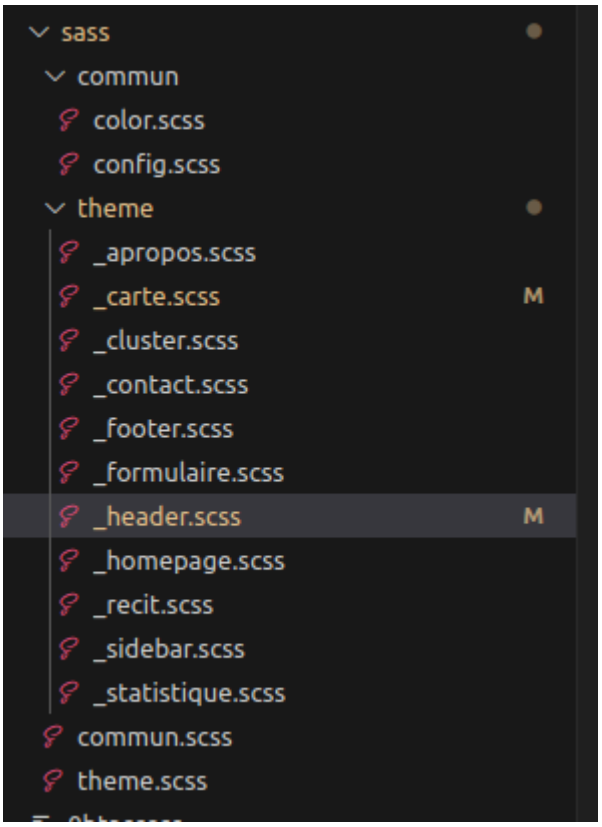
User story traitées / issues

J'ai travaillé sur une user story qui consiste à refactoriser la partie style d'un site web existant. Ma tâche était d'améliorer et de modifier le style actuel. Pour ce faire, j'ai utilisé CSS (Cascading Style Sheets), un langage informatique essentiel pour définir la présentation et le style visuel des pages web. CSS permet de séparer la structure (HTML) et le style (CSS) d'une page web, ce qui offre une approche plus efficace et flexible pour la conception de sites web.

J'ai également travaillé sur une deuxième fonctionnalité, la création de la version mobile du site. Mon travail consistait à rendre le site réactif, de sorte qu'il s'adapte aux appareils mobiles. J'ai également ajouté un menu conçu pour les téléphones, permettant ainsi une navigation plus claire et présise sur les petits écrans. C'est pourquoi j'ai dû ajouter un menu burger pour répondre aux besoins de l'expérience utilisateur sur mobile.

Avant /après

Avant le début de mon travail, le code était illisible, avec tous les styles regroupés dans un seul fichier. Il était difficile de discerner à quoi chaque élément correspondait, ce qui rendait la maintenance du code compliquée, ainsi que l'ajout de code



Representation de l'arborescence avec le SASS.
Chaque fichier contient donc les styles correspondant à sa place. Par exemple, le fichier _header.scss contient le menu de navigation.
_footer.scss les styles du footer uniquements.

Avec cette nouvelle approche de travail, il y a une amélioration de l'efficacité de la création et de la maintenance des fichiers. Pour commencer à utiliser Sass, il faut :

Installez npm en utilisant la commande suivante :

```
sudo apt-get install npm
```

Ensuite, installez Sass :

```
npm install -g sass
```

Une fois tout installé, exécutez la commande suivante pour compiler le code Sass à chaque sauvegarde :
arduino

```
npm run sass
```

En ce qui concerne la version mobile, rien n'avait été prévu pour cette fonctionnalité, quand j'ai récupéré le projet. Il m'a fallu penser à une version mobile, et la concevoir.

Pour adapter correctement chaque élément de la page à une disposition adaptée aux smartphones, j'ai dû ajouter du code SASS afin de définir des règles de mise en page

spécifiques pour les appareils mobiles. J'ai également dû créer un tout nouveau menu, car le menu existant ne permettait pas un accès efficace à l'ensemble du site.

Exemple de code commentés

```
<div id="menu_mobile_container" class="d-block d-lg-none ">
  <nav class="navbar_mobile">
    <div class="burgermobile" id="boutonburger">
          <svg id="burger" class="ham1" viewBox="0 0 100 100" width="90" >
            <path
                class="line top"
                d="m 30,33 h40 l -42,38" />
            <path
                class="line middle"
                d="m 30,50 h 40" />
            <path
                class="line bottom"
                d="m 30,67 h40 l -42,-38" />
          </svg>
      </div>
  </nav>
```

Le code que vous voyez ici inclut un SVG qui affiche une icône de burger pour la version mobile. Cependant, en combinant JavaScript (JS) et SASS, il est possible de transformer cette icône de burger en une icône de croix quand le menu est ouvert.

```
$black: #000000;
$green: #20A238;
$blue: #2E4C9B;
$white: #FFFFFF;
$light-blue: #1ba4b6;
$grey: #555;
$light-grey : #777;
$orange : #ff7800;
```

Il s'agit d'une partie d'un fichier de configuration SASS qui permet la configuration de variables de couleur. En modifiant la valeur d'une de ces variables, l'ensemble du site s'adaptera automatiquement à la nouvelle couleur.

Justification

Pour simplifier la gestion des styles, j'ai choisi d'adopter une nouvelle technologie : Sass. Sass permet d'optimiser la lisibilité du code en autorisant l'imbrication de classes. Par exemple, le CSS nécessite une classe distincte pour chaque élément auquel vous souhaitez appliquer des styles. En revanche, Sass permet de créer des structures plus complexes :

```
#id{
        .une_classe{
                .une_classe{
                        …
                }
        }
}
```

De plus, Sass permet d'utiliser des variables, principalement pour définir des couleurs. En modifiant une seule variable, il est possible de mettre à jour l'ensemble des couleurs du site. Ainsi, Sass présente de nombreux avantages par rapport au CSS, en offrant une version améliorée de ce dernier.

L'utilisation d'une version mobile est obligatoire de nos jours, car regarder internet sur son téléphone est très facile. **Dans 75 % du temps, selon Médiamétrie, c'est le mobile qui est utilisé pour aller sur Internet**. Soit cinq fois plus que sur l'ordinateur.
Il était donc primordial de faire une version mobile.

Les risques / les freins

Dès le début du projet, nous avons rencontré plusieurs problèmes liés à l'infrastructure. Au moment de notre arrivée, l'équipe travaille en local sans serveur en place. Par conséquent, nous avons sollicité le responsable informatique pour mettre en place un serveur, mais des problèmes ont surgi en cours de route. Nous avons donc perdu un temps considérable qui s'étale sur plusieurs jours. Aujourd'hui encore le serveur n'a pas réussi à être mis en place, nous nous sommes donc rabattu sur l'utilisation du localhost. Dans ce cas aussi des problèmes ont surgi au niveau de l'utisation de CodeIgniter qui est un modèle de conception MVC.
Nous avons été dans l'obligation de créer des VM linux pour utiliser le localhost avec CodeIgniter.

Conseils pour la suite

Respecter le nouveau modèle de style. Ne pas hésiter à creer de nouveaux fichiers sass pour simplifier la lecture du code au maximum. Si on avait eu plus de temps j'aurais conseillé de changer toutes l'architecture du projet. CodeIgniter n'a pas d'intérêt dans ce cas present, il est trop compliqué pour un si petit site. L'utilisation de Wordpress ou même sans acun modèle MVC aurait été plus facile.

Pour ce qui concerne la SAE en général:
Donner un serveur en même temps que le debut du projet.
Eviter de donner 3 jours avant la fin du projet un rapport à rendre qui compte pour 80% de la note.