

Dernières notes sur le projet final de la recherche d'informations et site WEB

1. Abandon du travail en binôme

En raison de la coopération pas toujours satisfaisante, on a choisi de travailler à nous-mêmes. On a gardé dans le présent site certains éléments contribués par Nabila Salah pendant le premier étape (p.ex. l'idée de faire des avis d'étudiants).

2. Responsive web design

Le site a été ajusté pour être affiché sur écrans de différentes tailles. Malheureusement, il n'était pas possible de le tester sur mobile, alors la mise en page pour les écrans/fenêtres de moins de 450 px de largeur peut s'avérer d'une qualité douteuse – afin de tester la mise en page pour les écrans/fenêtres plus petits, on a dû accroître l'affichage du page dans le navigateur, ce qui change simultanément la taille de la police.

Pour observer les effets du RWD, tout simplement changez la largeur de la fenêtre de votre navigateur. Le changement le plus important est la manière d'affichage du menu, mais vous pouvez observer aussi des changements de la pied de page et de la mise en page de certains contenus (surtout ceux avec propriété `display: flex`, alors la page d'accueil, les avis étudiants ou les données de personnes enseignantes).

3. Amélioration des feuilles de style

Par rapport au 2^e étape, les feuilles de style ont été ordonnées, surtout les éléments liés au RWD. Comme une partie majeure de la feuille de style originelle a été consacrée au slideshow de la page d'accueil, on a décidé de créer une feuille séparée pour les sous pages. Il serait plus optimal de trouver les éléments communs, les mettre dans une feuille commune et puis créer d'autres feuilles avec les éléments spécifiques pour chaque page, mais cela exigera plus de temps qu'il le vaut la peine avec un projet tant simple. Les différences entre les éléments `h1`, `h5` et `strong` pour la page d'accueil et pour les sous-pages sont intentionnelles.

4. JavaScript

Au lieu d'utiliser la bibliothèque jQueryUI, on a décidé de faire le code sur-mesure pour répondre aux besoins spécifiques pour le site qu'on a imaginé au début du projet. On croit qu'il

faut ajouter les moyens aux buts et pas à l'envers. Pour cette raison le menu du page a été créé seulement en HTML/CSS – les menus de la bibliothèque jQueryUI n'étaient pas convenable au design du site. Probablement on a pu faire plus d'éléments en JavaScript, mais on ne l'a pas trouvé nécessaire. Le code d'accordéon n'a pas été réutilisé (il y en a deux fichiers pour deux sous-pages) pour avoir de différentes propriétés d'affichage (`block` et `flex`).

5. Moteur de recherche

Dans la conception du site web, on a mentionné la possibilité de créer un simple outil de recherche. Faute de temps pour le faire proprement, on a abandonné cette idée.

6. Carte Google

En accord avec la suggestion de M. Aouinti, on a utilisé les cartes Google pour indiquer la localisation des sites universitaires. Malheureusement, les services de localisation Google sont payants. On a donc utilisé un `<iframe>` comme substitution. Si on a eu l'accès à une clé API on aura mis dans le code de la page Localisation :

```
<script  
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap&libraries=&v=weekly"defer></script>  
<script src="../../JS/map.js"></script>  
  
...  
  
<div id="map-malsherbes" class="map"></div>
```

Vous pouvez consulter le fichier `map.js` qui est inclus dans le dossier JS, mais qui n'est pas utilisé dans le projet. On a préparé seulement le code pour le site Malsherbes pour montrer une compréhension générale du code JS (les sites Sorbonne et Maison de recherche auront été fait respectivement). Malheureusement, on n'a pas pu tester le fonctionnement du code – il peut y avoir des erreurs, surtout parce que dans son tutoriel Google utilise un exemple assez ambigu avec `map:map` dans la dernière ligne¹ – on espère qu'on l'a bien interprété.

¹ Cf. <https://developers.google.com/maps/documentation/javascript/adding-a-google-map>