

SAÉ 1.05 : Produire un site Web

En tant qu'intégrateurs ou développeurs juniors dans une agence Web, les étudiants doivent mener un travail de conception, d'intégration et de développement des vues d'un site Web. Le cahier des charges comprend la description des éléments statiques de l'application et un jeu de données structurées représentant un échantillon des contenus à mettre en ligne. Les étudiants doivent répondre à la question :

Comment produire efficacement un grand nombre de pages Web, valides et accessibles en automatisant autant que possible le travail d'intégration ?

Apprentissages critiques :

– AC14.01 | Exploiter de manière autonome un environnement de développement efficace et productif

Pour mettre en place un environnement de développement efficace, productif et autonome, j'ai utilisé GitHub, notamment pour la gestion de version et la simplicité à retrouver son code sur différents ordinateurs, cela m'a aussi permis de documenter chaque étape du travail grâce aux "commit". J'ai codé sur Visual Studio Code, avec l'extension Live Server, et "l'auto save" afin de travailler simplement sans avoir à enregistrer et actualiser la page web pour voir l'avancer de mon code sans devoir enregistrer et actualiser la page web pour suivre l'avancement de mon code. Tous ces éléments m'ont donc servi à être autonome et productif.

– AC14.02 | Produire des pages Web fluides incluant un balisage sémantique efficace et des interactions simples

Pour avoir une page web de qualité, j'ai fait attention à produire un code HTML respectant les balises sémantiques. J'ai utilisé des balises appropriées telles que `<header>`, `<main>`, et `<footer>` pour structurer le contenu. Les images sont accompagnées de textes alternatifs (`alt`) pour garantir leur accessibilité. Et j'ai ajouté des interactions simples (boutons et liens accessibles) pour favoriser une navigation fluide.

```
<!--Header-->
<header>

  <!--Titre-->
  <div class="titre_header" id="titre">
    <a href="index.html#titre_header">
      
    </a>
  </div>

  <!--Menu-->
  <div class="menu">
    <div><a href="#qui">Qui suis-je ?</a></div>
    <div><a href="#parcours">Parcours scolaire</a></div>
    <div><a href="#fred">Guillaumat</a></div>
  </div>

  <!--Bouton SAE à droite dans le header-->
  <div class="bouton-header"><a href="sommaire.html">SAE</a></div>
</header>

<!--Contenu principal-->
<main>
```

Extrait de mon code, avec des balises sémantiques

– AC14.03 | Générer des pages Web à partir de données structurées

Pour générer des pages Web à partir de données structurées, j'ai utilisé la base de données "data.js" fournie par Mr Graef. J'ai donc écrit des scripts afin d'utiliser les données de la manière la plus dynamique possible, en évitant les répétitions manuelles des tâches. Cela m'a permis de produire et de faire un script pour remplacer le code de 10 pages différentes. J'ai utilisé 2 script, en JavaScript, un pour afficher le sommaire, présentant le numéro, le titre, et la ou les compétences de chaque SAE. Le second script sert à afficher le détail d'une SAE, avec le numéro, le titre, la description, la ou les compétences, les apprentissages critiques et les ressources correspondantes à la SAE.

```
//Génère une liste des Compétences associées à la SAE
let Liste_compétences = "";
for (let compétence in sae.compétences) {
  Liste_compétences += `<div>${sae.compétences[compétence]}</div>`;
}
document.querySelector(".competences").innerHTML = Liste_compétences;
```

Extrait du script "detailsae.js", plus précisément pour la partie permettant d'afficher les compétences liés à la SAE correspondante.

– AC14.04 | Mettre en ligne une application Web en utilisant une solution d'hébergement standard

Le site a été hébergé sur un serveur web dédié à l'Intranet de l'IUT à l'aide de l'application WinSCP. Cette solution permet un transfert sécurisé des fichiers via le protocole FTPS. Pour commencer, j'ai configuré la connexion au serveur en utilisant les paramètres fournis, comme le nom d'hôte et le port (21), et en utilisant mes identifiants UHA pour m'identifier. Une fois connecté,, j'ai sélectionné les fichiers HTML, CSS et JS de mon site depuis mon arborescence locale et les ai transférés sur le serveur. Enfin, j'ai vérifié le bon fonctionnement du site en testant son URL dédiée. Cette démarche m'a permis de comprendre les étapes nécessaires pour mettre en ligne un site Web sur un serveur local tout en respectant les bonnes pratiques de sécurité et d'organisation.