

## **RAPPORT DE PROJET**

Le premier problème que nous avons rencontré n'était en réalité pas si important. On était deux amis d'origine orientale différente sur un seul projet pour mettre en avant un seul pays oriental ; mais lequel choisir, l'Algérie ou l'Egypte ? Nous avons fait donc le choix de représenter l'Algérie, un pays si beau et si diversifié mais méconnu du monde, qui n'attendait que d'être mis en avant par sa richesse culinaire et traditionnelle. Cette première phase du projet concernait uniquement la mise en forme du site du restaurant (HTML et CSS), afin d'avoir un prototype de site web structuré et visuellement attirant.

Avant de commencer à bosser sur le code en lui-même, j'ai réalisé une maquette sur Whiteboard de chaque page web à effectuer afin d'avoir au préalable dans nos têtes une organisation d'éléments qui nous permettrait de faciliter l'écriture du code. Je me suis occupé des pages contenant les éléments les plus récurrents et complexes à structurer : la page d'accueil, la page de présentation des plats, la page administrateur, la page du profil personnel et le codage de l'entête principal du restaurant avec les onglets principaux qui permettront dans les prochaines phases de naviguer dans le site web. Adam s'est chargé des pages plus "indépendantes" c'est-à-dire les pages de livraison, commande, notation de la livraison et profil du resto (connexion ou inscription = formulaires). Tout au long de cette phase, chacun de notre côté, on a construit la charte graphique tout en définissant une palette de couleur dominante (le violet).

Il va sans dire que les pages où l'entête était présent en plus du bloc principal qui changeait selon chaque page étaient difficiles à coder. En effet, c'était une structure complexe qui nécessitait une organisation laborieuse du code HTML pour réussir à aligner et organiser les éléments correctement avec le CSS. Il nous fallait donc plus que des balises `<div>` pour ça. D'où le choix des balises répétitives `<nav>`, `<main>` et `<section>`. La balise `<nav>` m'a permis de regrouper dans un encadré tous les liens de navigation, utiles dans la phase 2. La structure du bloc qui définissait l'entête qui revenait tout le temps était mille fois plus claire et améliorait la compréhension du document. Cela m'a aussi énormément aidé pour le CSS et l'alignement des onglets, très difficile seulement avec que des `<div>` sans structure logique et un padding/margin à ajuster et tester inutilement maintes et maintes fois. Les balises `<main>` et `<section>` avaient un but similaire. `<main>` isolait le contenu principal qui devait s'adapter selon chaque fonctionnalité de la page, centrait bien les sections en évitant les manœuvres répétitives de marge/espacement dans le CSS et permettait d'éviter des conflits de nommage pour les sélecteurs avec le `<header>`. La balise `<section>` permettait de séparer efficacement les blocs dans le bloc principal du `<main>`, par exemple les différentes cartes dans la page du profil perso.

Pour la police du texte du nom du restaurant, on a découvert grâce à l'intelligence artificielle (Chat GPT, Claude) qu'on pouvait importer des polices beaucoup plus stylées pour mettre en avant les choses importantes. On a donc ajouté un lien externe dans le <head> du HTML (`<link href="https://fonts.googleapis.com/css2?family=Great+Vibes&display=swap" rel="stylesheet">`) pour pouvoir utiliser la police "Great Vibes" dans le font family dans le CSS.

Pour ce qui est du CSS, nous travaillions indépendamment, donc certains sélecteurs (body, h2, etc) rentraient en conflit et l'affichage ou les polices changeaient sans aucune raison. C'est pour ça que moi qui avait des choses qui se répétaient j'ai décidé de faire un seul fichier style.css et Adam a codé plusieurs fichiers .css pour chaque page concernée. De plus, comme on s'y attendait, en plus du problème des balises, on avait aussi énormément de mal à aligner les éléments avec le CSS : `display:inline block` ne suffisait pas pour le <header> donc on a défini des largeurs précises (en %) et utilisé `vertical-align`, le `display:flex` avec `justify-content` et `align-items` nous a permis de centrer correctement les cartes dans les pages de commande et de livraisons avec le `height:100vh` sur body directement. Il y a eu aussi un dysfonctionnement auquel on n'avait pas du tout pensé, c'était le fait que sur la page de la présentation des plats, les "petites fiches" pour chaque plat se désordonnaient complètement sur pc par exemple quand on réduit la taille de la fenêtre du navigateur, j'ai donc dû, en plus du `display:grid`, ajuster manuellement le gap en testant sur plusieurs tailles de fenêtres (et encore je ne suis pas sûr que tout est fonctionnel). Cependant, la chose dont on est fier est le `position:absolute` pour que les menus déroulants, déjà difficiles à coder, puissent ne pas disparaître bizarrement quand on réduit la taille des fenêtres. Enfin, le dernier problème important à notifier – car il nous a pris beaucoup de temps alors que la solution était simple – était le sens du flex pour les étoiles dans la page de notation ; Adam s'était en effet rendu compte que pour le hover des étoiles, le sens changeait aléatoirement et on n'avait pas de solution. Mais après échange avec Claude et recherche approfondie sur MDN Web Docs (un site qui a été le troisième membre du groupe littéralement), on a trouvé tout simplement le `flex-direction:row-reverse`.

Cette phase de mise en forme nous a permis de produire un site visuellement attrayant pour ensuite le rendre fonctionnel dans les phases suivantes (avec le PHP notamment par exemple pour récupérer une vraie liste d'utilisateurs avec une base de données pour la page admin, etc).