Je vais commencer par faire un rapide rappel de l’objectif de ce projet :

Il s’agit de développer un site « mobile first » qui répertorie les menus de différents restaurants. Nous avons 4 restaurants différents, on doit donc développer 4 pages de menus (une par restaurant donc).

Le développement doit se faire en CSS avec Sass.

Dans les livrables attendus, il est précisé qu’au clic sur la carte, l’utilisateur est redirigé vers la page du restaurant choisi – CLIQUER – ce qui est bien le cas ici. Au niveau du footer, au clic sur « Contact » un renvoi vers une adresse mail doit être effectué – CLIQUER – ce qui est le cas également. Au niveau du header des pages de menus, on doit avoir une flèche qui renvoi à la page d’accueil – CLIQUER – ce qui est le cas.

Au survol des boutons principaux, ces derniers doivent légèrement s’éclaircir – SURVOLER.

Le bouton j’aime en forme en cœur doit se remplir progressivement au survol en version desktop. EXPLICATIONS : Pour cela, j’ai superposé 2 cœurs l’un sur l’autre (un vide et un rempli) en position absolute. Comme l’objectif est partir d’un cœur transparent à un cœur rempli, j’ai mis un color :transparent au cœur plein et le linear-gradient sur le background. MAIS, comme le background est un carré qui recouvre le cœur, j’ai utilisé -webkit-background-clip: text pour que le background prenne la forme du « texte » (la forme du cœur ici). Et pour obtenir l’effet de transition du cœur qui se rempli lentement, j’ai mis une opacity de 0 au cœur puis une opacity de 1 au :hover et une transition de 500ms pour que le changement ne soit pas instantané et comme on doit obtenir l’effet de transition inverse quand la souris quitte le cœur, j’ai mis une transition de 500ms sur le :hover également.

A la page d’accueil, un loader doit apparaître – MONTRER – Il doit couvrir l’intégralité de l’écran avec les animations CSS. EXPLICATIONS : j’ai mis une z-index à 100 pour que le loader soit vraiment en « premier plan ». L’idée était que l’animation dure environ 3s au total et je voulais que pendant ces 3 sec, 3 icones différentes apparaissent successivement, l’une à la suite de l’autre, chacune avec une des 3 couleurs prédominantes du projet. J’ai donc commencé par déclarer 2 variables : une pour la durée totale de l’animation et une pour le délai d’apparition des icônes. J’ai ensuite opté pour une construction avec @for pour créer ce système de « boucle ». J’ai mis une opacity 0 à chaque icône afin qu’elles ne deviennent visibles qu’au moment de l’animation et avec une animation-delay un peu plus grande pour chacune, elles apparaissent l’une après l’autre avec une durée d’apparition égale pour chacune.

Et j’ai créé 2 keyframes : une pour l’animation du loader afin qu’il ait une opacity de 1 tout du long et qu’à la fin de l’animation, il ait un z-index négatif pour qu’il ne reste pas au milieu de la page à la fin de l’animation.

La 2nde keyframe était pour les icônes : comme je l’ai dit, je leur ai mis de base une opacity de 0 et avec cette construction, chacune obtenait une opacity de 1 lorsque c’était son tour sans se mélanger à l’icône suivante ou précédente.

Et j’ai choisi des icônes représentant des éléments de nourriture pour rester dans le thème du restaurant.

Sur les pages des menus, les plats doivent apparaître progressivement les uns à la suite des autres EXPLICATIONS : Pour ça, j’ai d’abord mis une opacity de 0 sur les plats, puis j’ai créé 2 variables : une pour la durée totale de l’animation et une pour le délai d’apparition de chaque plat. J’ai ensuite opté pour une construction avec @for pour que chaque plat apparaisse l’un après l’autre. Et j’ai fait une animation après avoir créé une keyframe où on partait donc de l’opacity 0 qui aboutissait à une opacity de 1 et également une transform :translateY(20px) à 0px pour obtenir cet effet du plat qui « arrive par le haut ». + animation-fill-mode :forwards pour que la cible retienne les valeurs définies sur de la dernière étape.

Comme l’idée est qu’à terme, le visiteur puisse sélectionner le plat qu’il souhaite, on veut faire apparaître une petite coche à droite du plat qui doit coulisser de la droite vers la gauche au survol et si l’intitulé du plat est trop long, ce dernier doit être rogné avec … EXPLICATIONS : Pour cela, j’ai d’abord créé la carte avec la checkbox visible, comme si elle était déjà en :hover. Je lui ai ensuite mis une margin-right négative pour la faire sortir du cadre. J’ai ensuite mis un overflow :hidden sur la carte pour cacher la checkbox quand elle à droite. Puis j’ai mis la margin-right à 0 au hover pour que la checkbox revienne à sa place au survol. Et enfin, j’ai mis une transform : translateX(0px) avec une transition d’1 seconde pour obtenir l’effet de coulissage.

Pour les … j’ai mis text-overflow :ellispsis sur les textes ainsi qu’un overflow :hidden et un white-space :no-wrap comme indiqué dans la Doc pour obtenir l’effet voulu.

L’autre objectif de ce projet était de travailler en « mobile first ». J’ai donc commencé à coder sur la base d’une présentation pour version mobile et une fois les pages au point, j’ai appliqué des adaptations en responsive design pour obtenir un affichage pour la version desktop. En soi, la manipulation était très similaire à ce qui avait été fait sur le projet précédent. La différence majeure a été la valeur attribuée à la min-width de @media.