

ohmyfood

Sass

Restaurants



La palette du goût
Ménilmontant



Bilan de projet « Ohmyfood »

https://tetyanachouteau.github.io/TC_P3_Dynamisez_page_avec_des_animations_CSS/index.html

Contexte du projet

Commande de repas en ligne de « Ohmyfood ».

Composition menu gastronomiques en ligne, desservie à New-York et Paris.

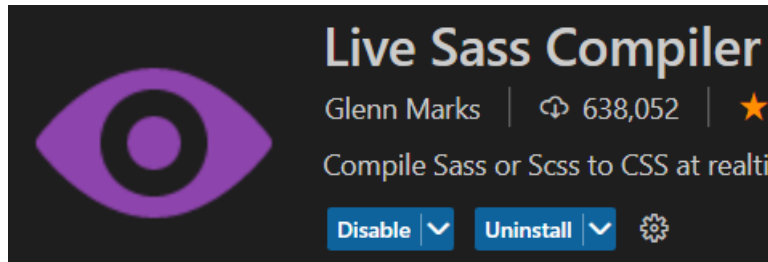
Code HTML, fait en CSS, sans JavaScript et framework sauf SASS.

Le site est responsive sur mobile, tablette, desktop.

Language SASS -> CSS, Instalation

Language Sass et sa syntaxe SCSS est réutilisable.

Il utilise des variables, fonctions, boucles, logique conditionnelle, compilation.



```
sudo gem install sass/npm install -g sass  
sass --watch scss:css
```

Exécutable:
`npm install -g sass-convert`

**SASS : @mixin, @import, @include,
opérations arithmétiques.**

**CSS : transition, @keyframe, transform,
@media query, pseudo classes,
pseudo éléments ::before, ::after**

Rendu visuel : desktop, tablette, mobile

Au clic sur la carte, l'utilisateur est redirigé vers la page du menu. 4 pages contenant chacune le menu d'un restaurant.

“loading spinner” couvre intégralité de l'écran.

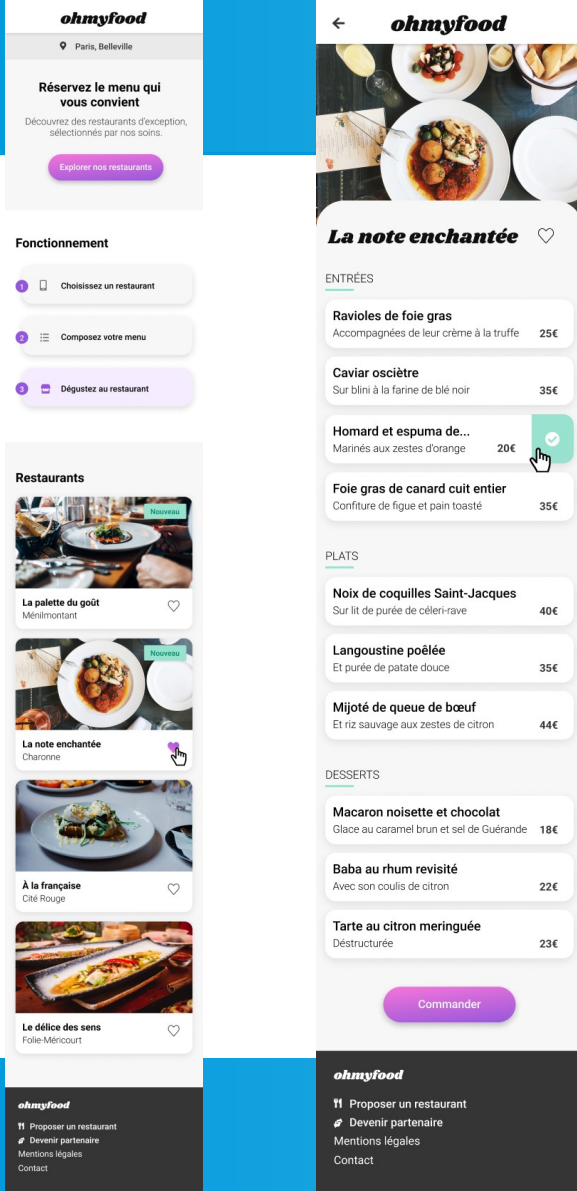
Menu : plats apparaissent progressivement avec un léger décalage dans le temps.

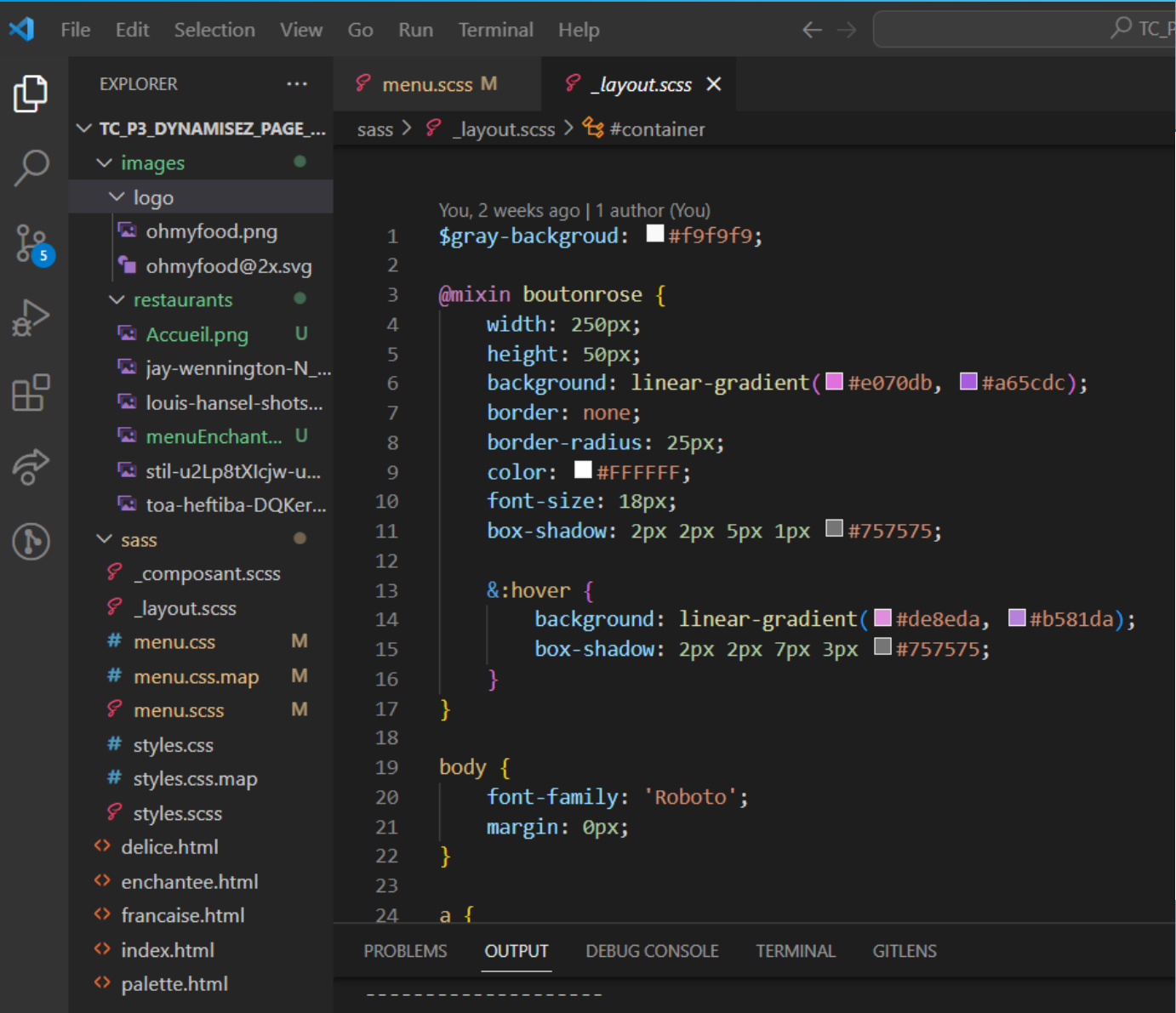
Si titre du plat est trop long, il se rogne avec « ... ».

Au survol, la couleur de fond des boutons principaux s'éclaircit. L'ombre portée est plus visible.

Bouton en forme de cœur au clic se remplit progressivement.

Coche coulisse de droite vers gauche.





Code du projet VSC

- **GitHub** : branches, commit, terminal
- **F12**
- **Copie des autres pages sur modèle de palette.html.**



<body>

<div id="container">

<header><div></div></header>

<main>

<div id="recherche"><form><button></button></form>

<article id= "fonctionnement"></article></div>

<section id="restaurant"><article><figure>x4</figure></article></section>

</main>

<footer></footer>

</div>

<div><i></i></div>

</body>

CSS : layout.scss, composant.scss

- Création menu.scss à partir styles.scss.
- Ajout styles dans **_layout.scss** avec @import 'layout'.
- Création variable scss pour fond de la div main et du menu (pour éviter de modifier l'un sans l'autre) et ajout cette variable dans styles.scss et menu.scss (**background-color css**).
- Création de **composant.scss** pour style des **cœurs et input checkbox** (pas sur même parent, donc c'est pas style en commun uniquement des définitions css en commun) @import dans les parents respectifs (style.scss : #restaurants et menu.scss : #titre).

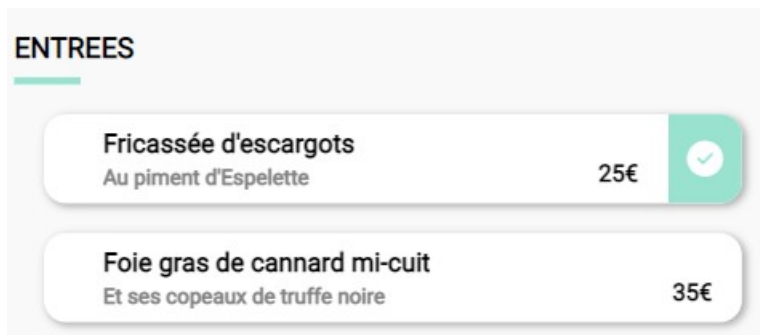
Bouton vert

Création dans li de 3 zones : titre, prix et **bouton**.

Sur hover du li on modifie propriété de son enfant bouton en right: 0 donc on le remet à ça place normale (**transition: right 1s;**).

Ensuite on fait tourner la case à cocher avec une animation sur i qui contient l'icône coche. FontAwesome.

Réutilisation de l'animation rotation (du spinner) et on la déclenche 500ms après le hover pour qu'elle se lance quand elle est visible sur le bouton (**animation: rotation 500ms linear 500ms;**).



Apparition progressive des : fadeIn

Pour faire apparaître les entrée, plat et dessert progressivement on ajout crée un @KeyFrame fadeIn qui fait varié l'opacité:

```
@keyframes fadeIn {  
  0% {  
    opacity: 0;  
  }...  
}
```

```
@for $i from 1 through 3 {  
  &:nth-child(#{ $i }) {  
    animation-delay: #{ ($i - 1) * 200 }ms;  
  }  
}
```

On applique cette animation sur les .

Pour qu'ils apparaissent les uns après les autre on utilise la propriété nth-child pour ajouter un délais si le li n'est pas le premier. On ajout 200ms à chaque fois :

```
li. nth-child().  
&:nth-child(1)
```

On fait à la place une boucle SASS.