

Planetarium VueJS

Julie Chartier L3 SDN

1. Le projet

L'API

The Solar System OpenData : <https://api.le-systeme-solaire.net/en/>

Documentation : <https://api.le-systeme-solaire.net/swagger/>

/!\ Attention au nombre de requêtes effectuées avec le projet car cela m'est déjà arrivé de me faire bloquer par l'hébergeur du site en développant. Si cela arrive, veuillez utiliser une autre connexion internet ou un VPN.

L'idée

Après une recherche trop longue d'une API à exploiter, j'ai eu l'envie de sélectionner une API avec laquelle je pourrai essayer de représenter le système solaire et de pouvoir afficher des informations simples sur des corps célestes.

La difficulté de ce projet est que l'API ne donne pas d'informations sur la couleur, l'apparence ou une éventuelle image de la planète. J'ai donc dû trouver un moyen de les représenter pour ne pas seulement afficher une liste avec des informations surtout chiffrées.

La maquette

Pour réaliser mon idée avec la meilleure efficacité possible, j'ai réalisé une maquette avec l'outil Figma car j'aime beaucoup développer des choses en sachant déjà à quoi mon travail doit ressembler. Cela m'aide beaucoup.

Voici le lien de la maquette Figma :

<https://www.figma.com/file/jx5YBehXKeMsTuruGNOtFD/Planetarium?node-id=0%3A1&t=wb43yB8d10Y9hBkJ-1>

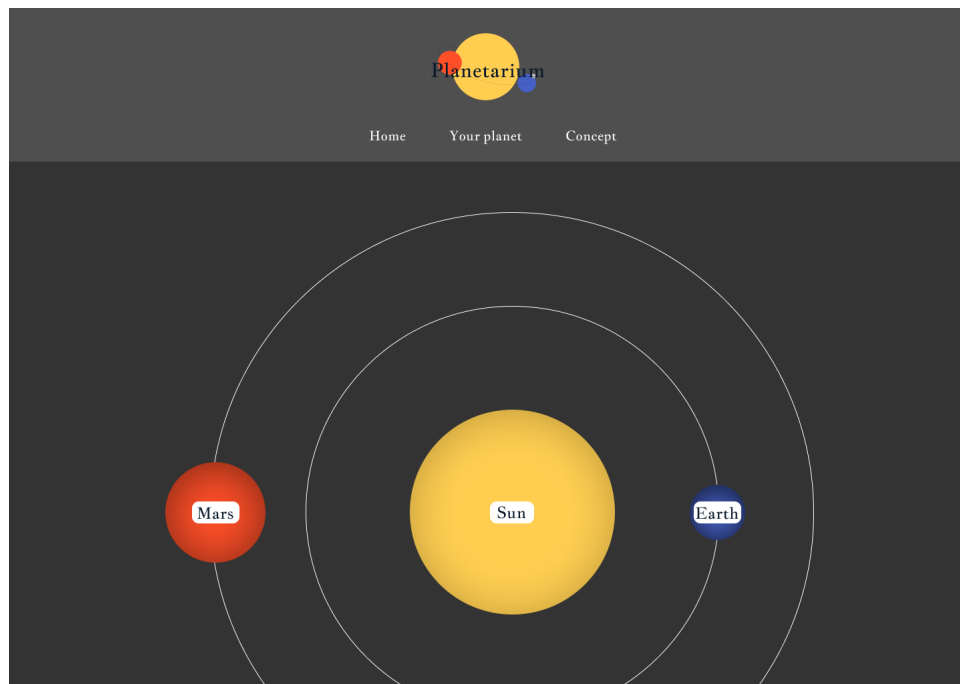
Cette maquette a été réalisée assez rapidement donc n'est pas tout à fait complète. Cependant, cela me suffisait pour me représenter les différents views et composants dont j'allais avoir besoin.

Le logo a été réalisé par mes soins sur le même outils.

Le cahier des charges

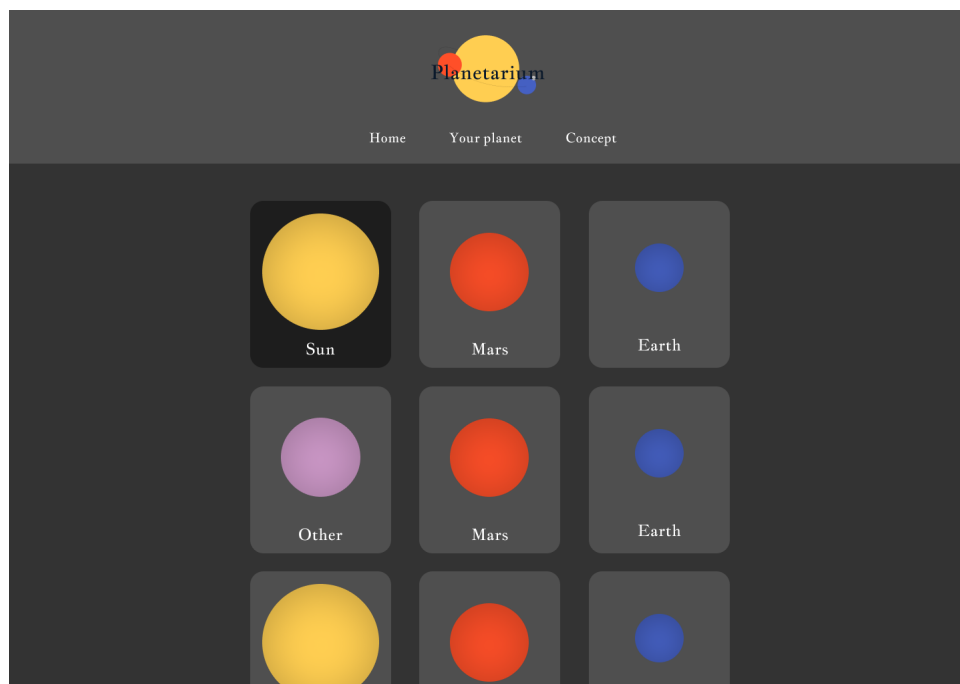
Ainsi, mon projet se découpait de la manière suivante.

Page d'accueil



La page Home devait afficher les planètes du système solaire avec des positions cohérentes avec la réalité et adaptée au format de page web. Sur la maquette, les positions observées sont fictives.

Liste des astres

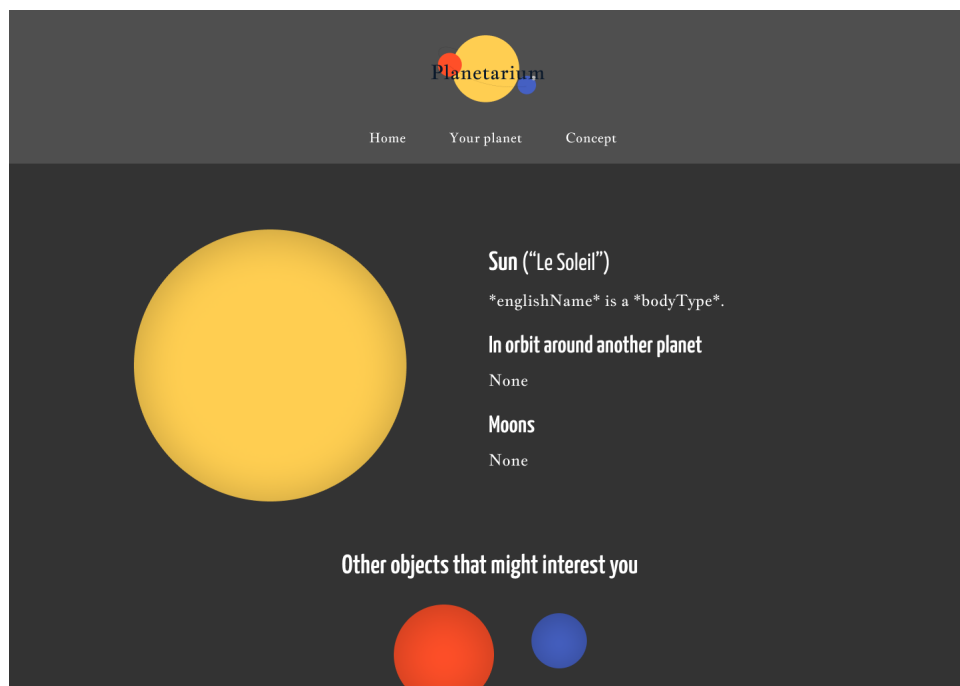


La page “Your planet” devait lister les cards de tous les objets recensés par l’API avec un span pour illustrer de la bonne manière le corps. Mon objectif était de définir des valeurs bien particulières pour les couleurs et apparences des planètes du système solaire.

Voici le raisonnement :

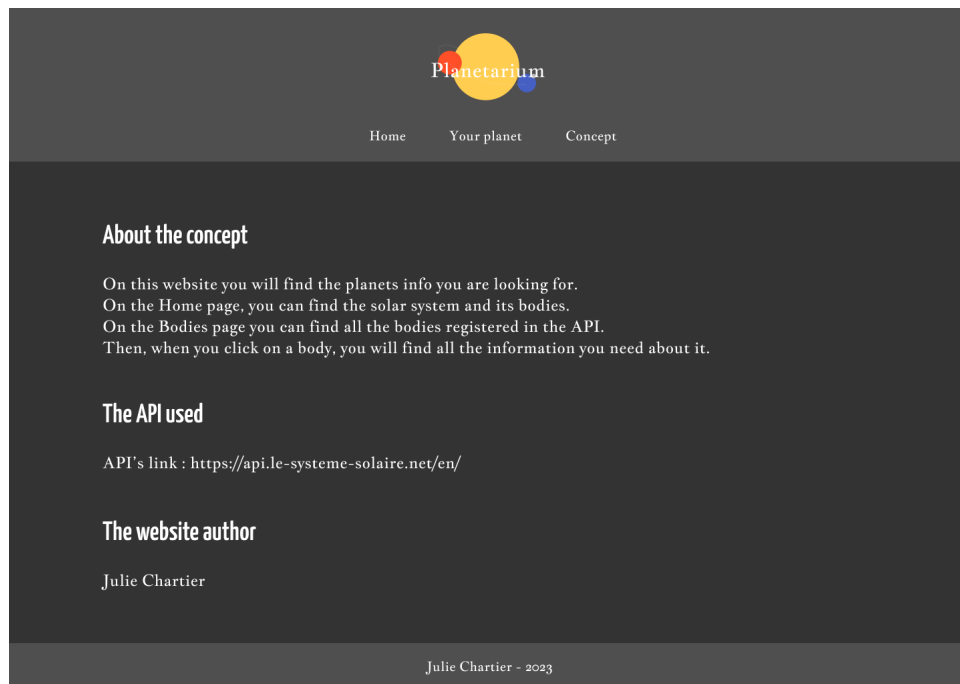
- Si l’astre fait partie de [sun , mercury , earth , jupiter , uranus , venus , mars , saturn , neptune]
 - Alors on affiche le css défini pour chacun de ces astres
- Sinon
 - Alors on affiche le css générique défini pour le type d’astre correspondant (lune, planète, etc)

Un astre



Lorsque l'utilisateur clique sur un astre alors il peut voir sa page détaillée avec le nom anglais et français, les planètes autour desquelles il est en orbite et ses lunes. Enfin, il y a des liens vers d'autres astres qui pourraient l'intéresser.

Concept



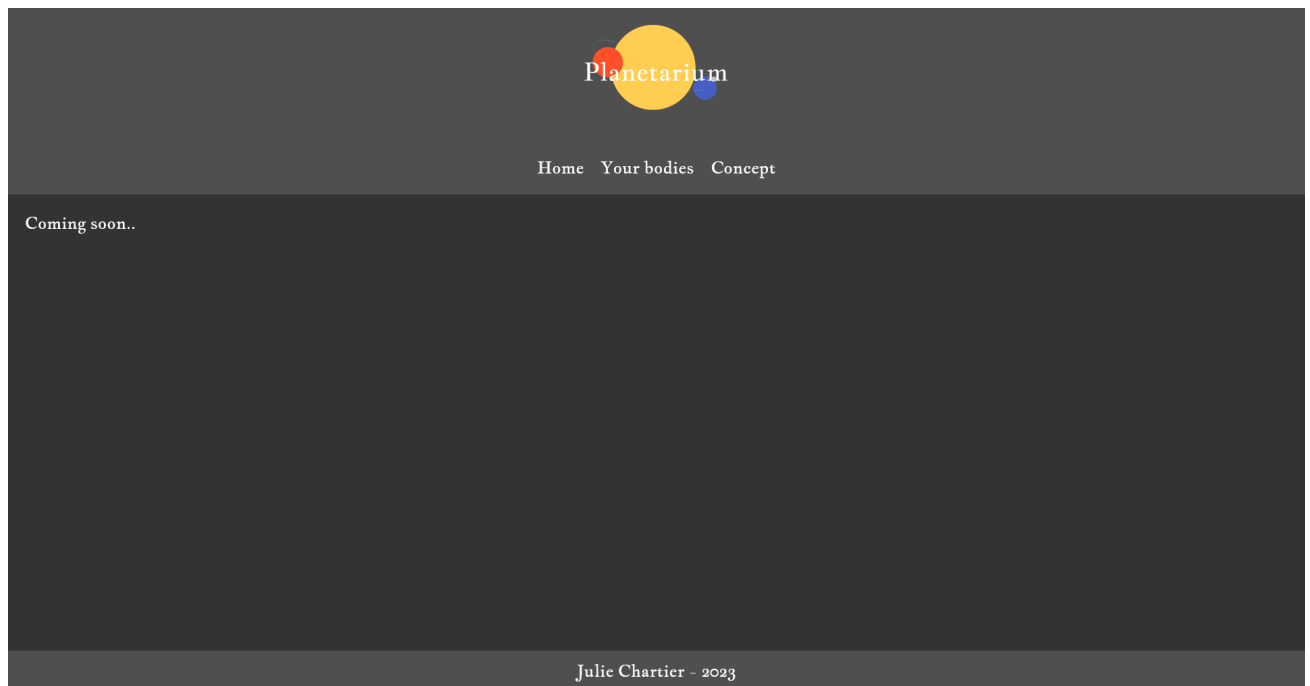
Et pour finir, la page Concept est une petite page comprenant seulement du texte pour expliquer le projet et son contexte, citer l'API et les crédits.

2. Ce qui a été réalisé

En définitive, tout le projet n'a pas pu être réalisé comme je le souhaitais. Je vais détailler dans cette partie le travail réel effectué en VueJS.

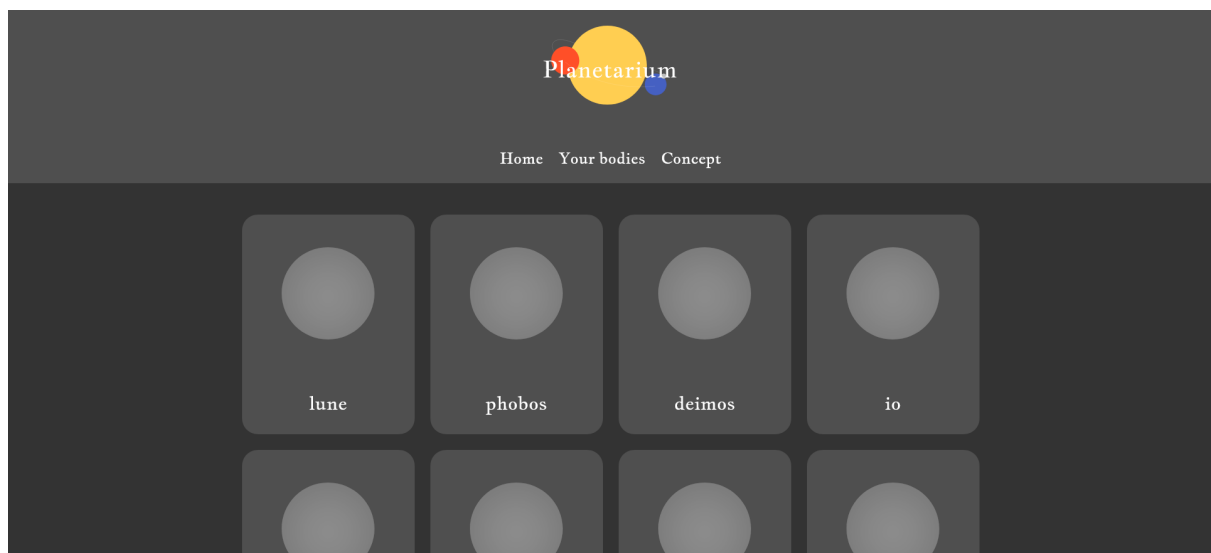
Page d'accueil

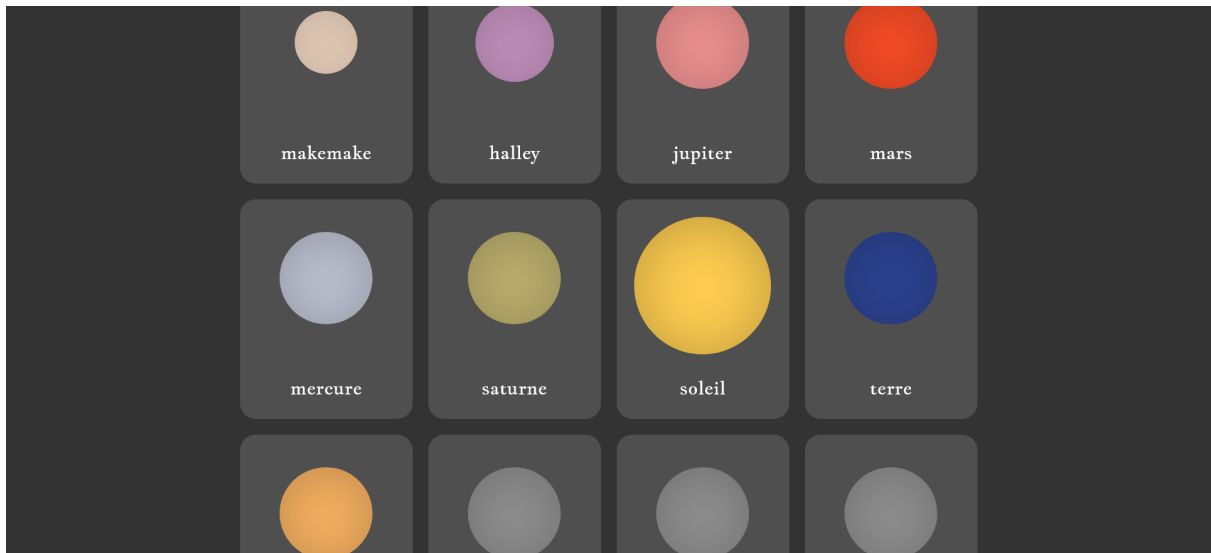
Dans la page d'accueil, il n'y a rien pour le moment à part les composants communs à toutes les pages du site. C'est-à-dire que les composants que sont la navbar (avec le logo et les différents router links vers les autres pages) et le footer (avec mon nom et l'année) sont présents.



Liste des astres

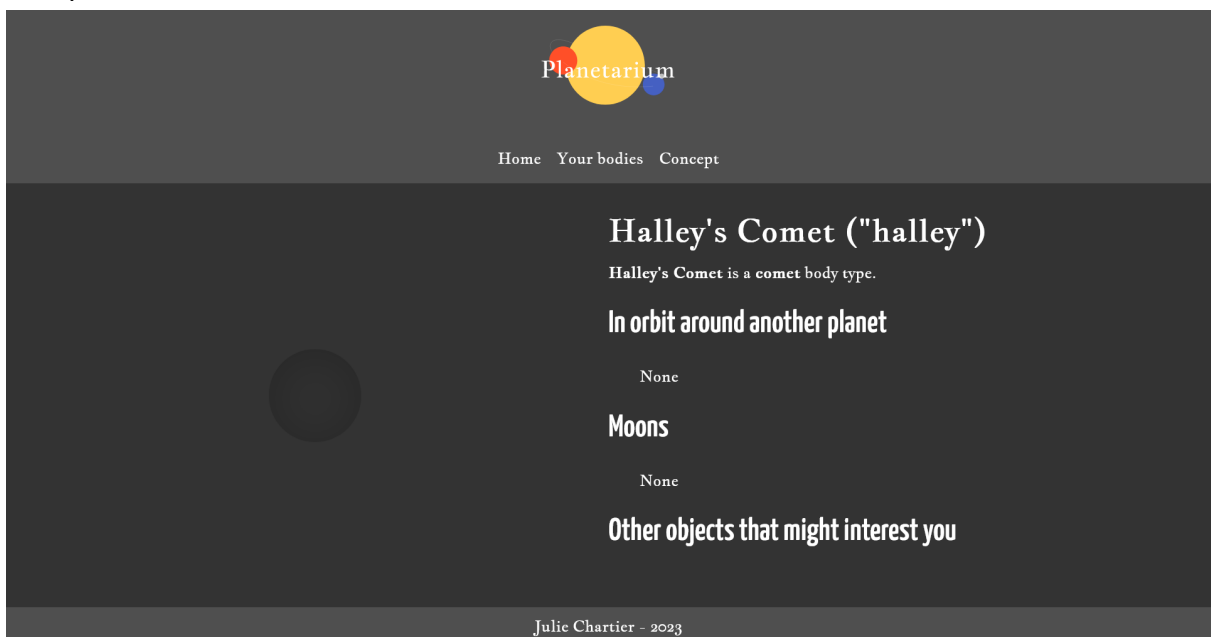
Dans la page qui liste les astres, j'ai réussi à faire ce que je souhaitais car la view est composée de composants qui correspondent aux cards sur lesquelles nous pouvons cliquer et accéder à davantage d'informations. Je pense avoir réussi à afficher les bonnes illustrations mais d'une manière détournée par rapport à ce que j'avais imaginé à l'origine.





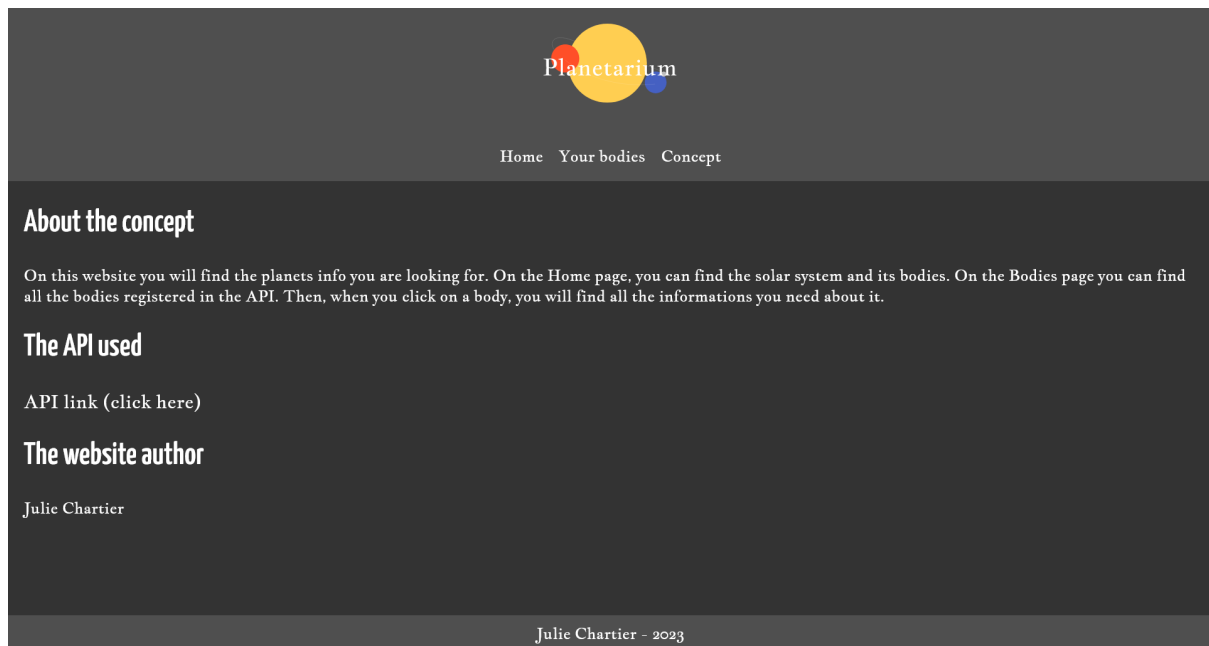
Un astre

Dans la page qui détaille un astre nous retrouvons le contenu comme cela était prévu mais il manque les finitions de la mise en forme.



Concept

Même chose que pour la page représentant un astre individuellement, le contenu est présent mais la mise en page pourrait être améliorée.



3. Points de blocage

Sélection d'une API

La sélection d'une API pour mon projet a été un vrai frein pour moi. Le fait d'être libre de son sujet m'a souvent fait changer et longtemps hésiter.

Tout d'abord j'avais envie d'utiliser l'API d'OpenFoodFacts mais n'ai pas su vraiment quoi en faire donc j'ai préféré changer pour quelque chose de plus concret avec un objectif.

Dans un second temps, j'ai trouvé une idée très précise pour répondre à l'un de mes besoins. Celle-ci était de trouver une API sur les plantes avec diverses informations d'entretien mais surtout avoir la possibilité de savoir si la plante était compatible avec un chat ou un animal de compagnie en intérieur. En effet, ayant un chat et une passion grandissante pour les plantes d'intérieur, j'aurais aimé pouvoir développer un outil me permettant en un coup d'œil de savoir si cette dernière est toxique pour mon animal. Finalement, je n'ai pas trouvé d'API gratuite répondant à mon besoin.

Enfin, je me suis rabattue sur une API avec des données un peu plus "simples" pour pouvoir faire un travail complet, celle que je vous ai présenté en début de cette documentation.

Affichage de l'image de la planète

Un point de blocage a été d'afficher les bonnes illustrations correspondant à l'astre dont il est question dans chaque component. Finalement, je m'en suis bien sortie mais je pense qu'il faudrait que je trouve un moyen de l'automatiser un petit peu plus en fonction de ce que l'on peut trouver dans la feuille css directement.

Page d'accueil et CSS

Un autre point de blocage a été l'affichage particulier de la page d'accueil avec les positions relatives des planètes du système solaire. Cela s'est avéré plus compliqué que prévu et j'ai donc préféré prioriser sur la liste des astres afin de pouvoir produire un résultat plus concret et rapide. Malheureusement, je n'ai pas eu le temps de me pencher sur la page principale.

4. Installation du projet

Ressources github

Vous pouvez trouver le projet Github via le lien suivant :

https://github.com/juchrt/planetarium_project

Installer le projet sur sa propre machine

1. Créer un dossier "planetarium_project" sur son environnement local
2. Copier ou cloner le projet dans ce dossier
3. Ouvrir Cmder
4. Exécuter : `npm run dev`
Cela permet de déployer le projet en local pour voir le résultat
5. Si Axios n'est pas installé sur votre machine, exécutez :
`npm install --save axios vue-axios`

5. Avis sur le cours

J'ai apprécié le cours de ce semestre et surtout le fait que vous avez proposé votre aide malgré votre charge de travail actuelle. Voici les axes d'améliorations auxquels je peux penser (mais qui sont tout de même minimes compte tenu des conditions dans lesquels vous avez dû préparer ce cours) :

- Réorganiser légèrement le diaporama de manière à guider un petit peu plus les étapes qui pourraient se suivre
- Ajouter des petits exercices pour nous faire pratiquer et que vous puissiez passer dans les rangs pour nous aider

Je pense que le cours était un peu rapide par moment mais pas tout le temps.