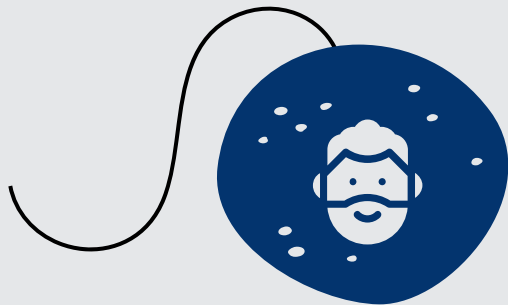




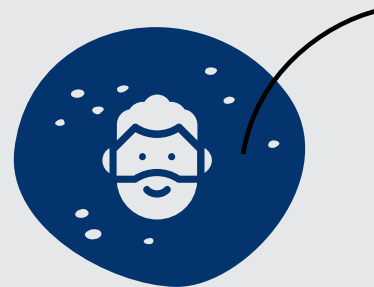
3WEBD-React SupKnowledge

Notre équipe



Dimitry Corbet

Développeur



Valentin Laîné

Développeur



Sommaire

01

Présentation du Projet

Objectifs
Description des outils utilisés

02

React

Description de la structure

03

Données et Affichage

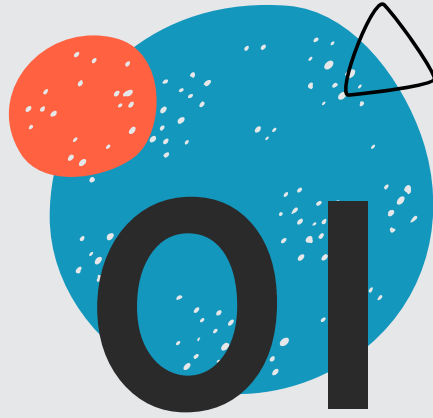
Description de la gestion
des données, et de son
affichage

04

Conclusion

Détails de ce que l'on aurait
pu ajouter / améliorer,
Démonstration





Présentation du Projet



SupKnowledge

- Créer et fournir une interface parfaite en utilisant une API de collection d'art public.
- Aider l'utilisateur à trouver des informations plus rapidement mais surtout fournir également une interface pour en savoir plus sur les collections.

Outils Utilisées

React est une bibliothèque
JavaScript pour la construction
d'interfaces utilisateur (UI)

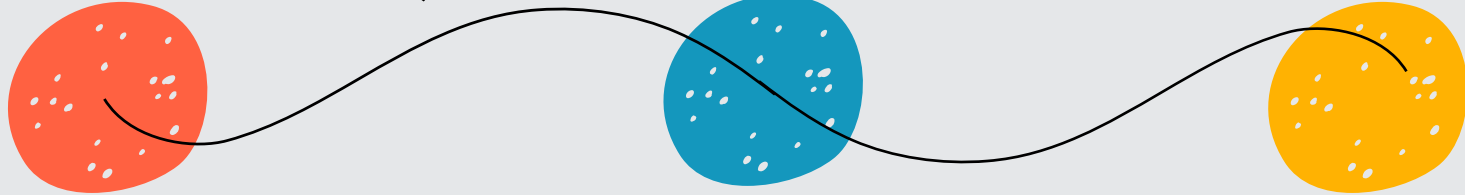
React (ReactJS)

The Metropolitan
Museum of Art
Collection API

API

Tailwind css
Framer-Motion
Ant Design

FrontEnd





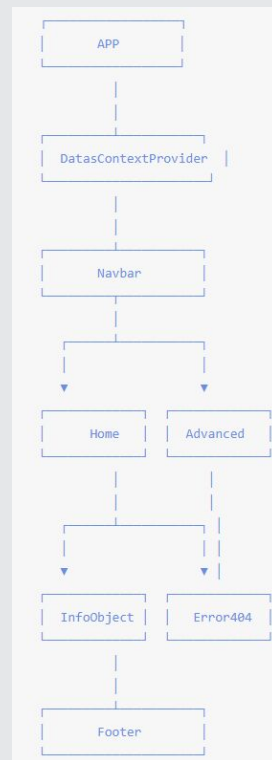
React

02

React Router

Afin d'avoir un système de page,
nous avons utilisé React Router pour
créer des routes.

```
<Router>
  <DataContextProvider>
    <Navbar />
    <Routes>
      <Route path="/" element={<Home />} />
      <Route path="/404" element={<Error404 />} />
      <Route path="*" element={<Navigate to="/404" />} />
      <Route path="/art/:objectID" element={<InfoObject />} />
      <Route path="/advanced" element={<Advanced />} />
    </Routes>
    <Footer />
  </DataContextProvider>
</Router>
```

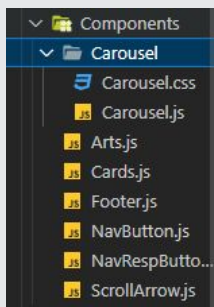




02

Component

Les Components nous ont permis de
créer des sortes de modules
réutilisables



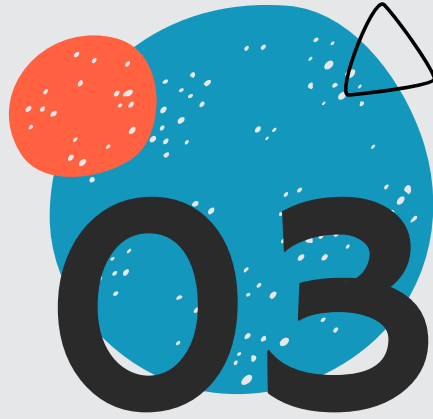
Disposition

Component

Container

Pages





Données et Affichage



03

Data Context

Pour gérer les données, nous avons utilisé un context, ainsi nous avons récupéré les informations à propos des œuvres à notre souhait.

```
// fetches only the card clicked and sends it to render
const fetchSingleArt = async (id) => {
  const resp = await fetch(
    `https://collectionapi.metmuseum.org/public/collection/v1/objects/${id}`,
    {
      method: "GET",
      cache: "force-cache",
    }
  )
  .then(async (response) => {
    const data = await response.json();
    setSingleToRender(data);
  })
  .catch(setSingleToRender([]));
};
```



03

Mise en Forme



Utilisation de quelque outils afin de mettre en forme le site, avoir un site ergonomique et beau visuellement pour l'utilisateur





Conclusion



04

Amélioration

Ce que l'on peut améliorer :

- Le temps de réaction de l'affichage des données
- Ajout de plus de filtres sur la recherche avancée
- Ajout d'un Zoom sur les images



04

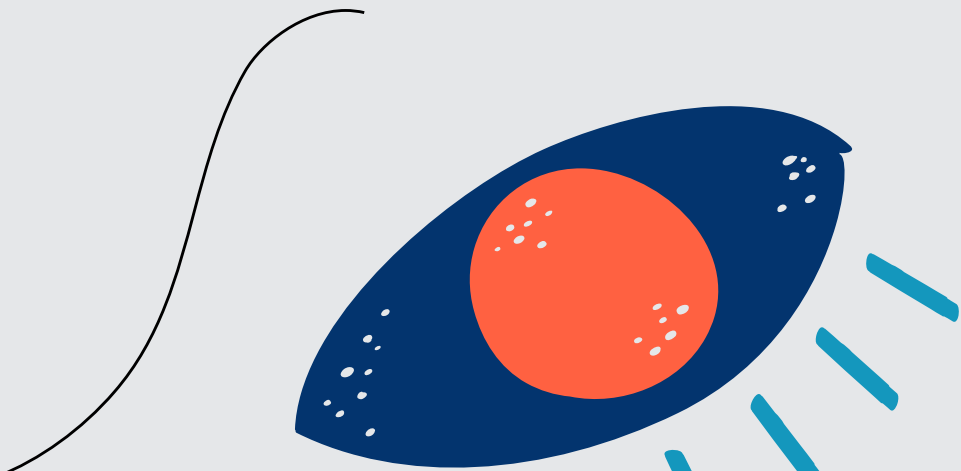
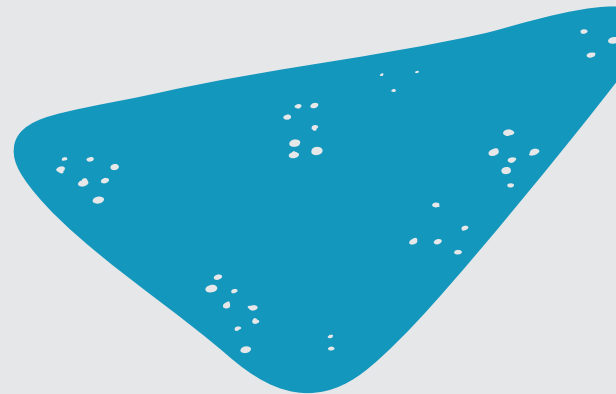
Conclusion

Malgré un fonctionnement correct de notre application nous avons rencontré des problèmes en utilisant l'API, de ce fait il nous à été impossible d'utiliser l'API pleinement.



Merci de votre attention

Avez-vous des questions ?



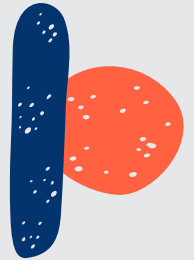
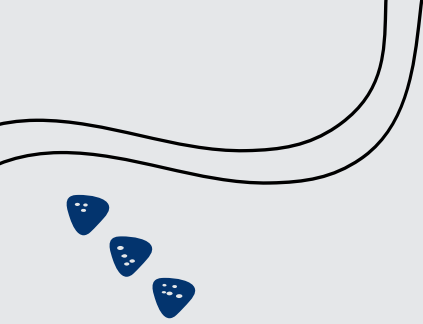
RESOURCES

- Utilisation de l'api "The Metropolitan Museum of Art Collection API"
<https://metmuseum.github.io/>
- Utilisation de l'API Wikidata [Wikidata:REST API - Wikidata](#)
- [Tailwind](#)
- [ANT Design](#)
- [Framer Motion](#)

Photos

- Image public provenant de l'api metmuseum et wikidata





Démonstration

