

Projet tutoré S6 2023

Encyclopédie de jeux vidéos

L'objectif de ce projet était de réaliser une encyclopédie de jeux vidéo avec des fonctions de recherche et de tri avancées. Parmi les interactions principales, nous devons entre autres réaliser une fonction de recherche avec une pagination des résultats et une gestion de bibliothèque basée sur cette encyclopédie. Par la suite, nous devons ajouter d'autres fonctions telles que la création de compte avec gestion de librairie.

Ce projet a été effectué par Adrien DESTREMONT , Yann HIPPERT , Clément STANCATO et Jade ROUBLOT.

- **Répartition des tâches**

Au sein de ce projet, Adrien DESTREMONT & Jade ROUBLOT se sont occupés du frontend et Yann HIPPERT & Clément STANCATO se sont occupés du backend. La répartition des tâches s'est effectuée à l'aide d'un Trello ci joint.

[Projet tut S6 | Trello](#)

- **Manuel d'installation**

Installation de docker:

Docker peut être télécharger sur:
<https://www.docker.com/>

Dans le root du projet:

```
docker compose -f ./backend/docker/docker-compose.yaml up
```

Une erreur qu'on a eu sur windows était:

Error: [1]: max virtual memory areas vm.max_map_count [65530] is too low, increase to at least [262144]

La solution trouvé sur stackoverflow a fonctionné:

<https://stackoverflow.com/questions/69214301/using-docker-desktop-for-windows-how-can-sysctl-parameters-be-configured-to-per>

Pour lancé les containers docker, s'ils ne le sont pas après `docker up`, utilisez la commande:

```
docker compose -f ./backend/docker/docker-compose.yaml start
```

Vous pouvez contrôler si le serveur à bien démarrer en allant sur:

http://localhost:5601/app/dev_tools

Mise en place du jeu de données:

Naviguez dans le directory datamodule:

```
cd ./datamodule
```

Installez les packages

```
npm install
```

Lancez le script pour ajouter le jeu de données

```
npm run app
```

Lancez le serveur backend:

Naviguez dans le directory backend:

```
cd ../backend
```

Installez les packages

```
npm install
```

Démarrez le serveur backend

```
npm run dev
```

Si le serveur a bien été démarré, la documentation de l'API est accessible sur l'URL suivant:

<http://localhost:9090/doc>

Lancez le frontend/serveur react:

Naviguez dans le directory frontend:

```
cd ../frontend
```

Installez les packages

```
npm i
```

Démarrez le serveur

```
npm run start
```

Normalement ça va ouvrir un nouveau tab, sinon le front devrait être accessible par l'URL: <http://localhost:3000>

- **Logins Utilisateur**

Aucun utilisateur n'est enregistré lors du démarrage de l'application.

- **Fonctionnalité**

De multiples fonctionnalités sont présentes, voici donc une liste non exhaustive des fonctionnalités existantes.

- **Frontend**

- Affichage d'une page d'accueil responsive
- Afficher de la liste de jeux sous forme de carte ou de tableau
- Interface de recherche / filtrage sur la page d'accueil
- Page détail de jeux responsive
- Page de login/register responsive
- Page d'enregistrement de nouveau utilisateur responsive
- Barre de navigation

- **Backend**

- Recherche de jeux par nom avec un fuzzy search
- Recherche de jeux par Développeur
- Recherche de jeux par Publishers
- Recherche de jeux par Platforms
- Recherche de jeux par Catégories
- Recherche de jeux par Genres
- Recherche de jeux par date
- Recherche de jeux par âge requis
- Recherche des jeux de l'utilisateur
- Création d'un utilisateur
- Connection utilisateur
- Ajout d'un jeux à sa library

· **Fonctionnalité manquant / pas fini**

Voici ci-joint une liste des fonctionnalités manquantes ou qui n'ont pas pu être abordées.

- ...
- ...

· **Liens git**

Ci joint le lien du github du projet.

[adriendst/Projet-S6: Projet S6 I \(github.com\)](https://github.com/adriendst/Projet-S6: Projet S6 I (github.com))

Ci joint le lien du github du typage du backend.

<https://github.com/yannHippert/LP-Steam-Wiki-Types>