

# Rapport de projet tuteuré

FRITSCH Matthieu, CARL Noé, KREBS Jérémie

<b>Liens Github.....</b>	<b>1</b>
<b>Manuel d'utilisation.....</b>	<b>1</b>
Lancement du projet backend.....	1
Lancement du script d'insertion des données.....	1
Lancement du projet backend.....	1
Lancement du projet frontend.....	1
<b>Logins utilisateurs.....</b>	<b>2</b>
<b>Tâche fonctionnelle.....</b>	<b>2</b>
<b>Tâche non fonctionnelle.....</b>	<b>2</b>
<b>Répartition des tâches.....</b>	<b>2</b>

# Liens Github

*Lien du projet back-end:*

<https://github.com/mattfritsch/steam-backend>

*Lien du projet front-end:*

<https://github.com/CarlNoe/steam-library>

## Manuel d'utilisation

Avant de suivre les étapes d'utilisation du projet, récupérer les deux projets ci-dessus. Puis lancer votre client elasticsearch, assurez vous aussi de ne pas avoir d'index **users** et **steam\_games**.

### Lancement du projet backend

Avant tout, veuillez effectuer cette commande pour initialiser le projet:

```
npm install
```

Une fois cela fait, vous pouvez vérifier dans le fichier **config/elasticsearch-client.js** si à la ligne 2 votre serveur elasticsearch se lance sur ce port, sinon changez le.

### Lancement du script d'insertion des données

Pour lancer le script d'insertion des données, placez vous dans le dossier **database** (*cd database* sur Windows) du projet backend et exécuter cette commande:

```
node getDataFromCsvToElasticsearch.js
```

Une fois le script terminé les données seront bien indexées dans elasticsearch.

### Lancement du projet backend

A la racine du projet backend, exécuter la commande ci-dessous:

```
node index.js
```

### Lancement du projet frontend

Une fois le projet backend lancé, veuillez faire cette commande à la racine du projet pour l'initialiser:

```
npm install
```

Une fois le projet initialisé, pour exécuter le projet frontend veuillez lancer cette commande à la racine du projet:

*npm run dev*

## Logins utilisateurs

Voici deux utilisateurs créés:

### Utilisateur 1:

- *username*: matthieu
- *password*: mdp123

### Utilisateur 2:

- *username*: noe
- *password*: mdp123

## Tâche fonctionnelle

Voici la liste des tâches fonctionnelles demandées pour ce projet:

- Le chargement complet des données CSV dans une base de données ElasticSearch.
- Affichage de la liste de jeux avec un scroll infini sans charger la liste complète des jeux.
- Fonction de recherche sur le nom des jeux avec la recherche de type fuzzy search.
- Recherche avec auto-complétion.
- Afficher une page contenant les détails d'un jeu.
- Tri de la liste de jeux dans l'application.
- Création de compte utilisateur.
- Connexion à un compte utilisateur.
- Possibilité d'ajouter des jeux à sa liste de favoris en tant qu'utilisateur. Ainsi que de consulter la liste des jeux favoris.

## Tâche non fonctionnelle

- Recherche d'un jeu dans la liste favoris de l'utilisateur.

## Répartition des tâches

Nous avons utilisé un trello pour suivre les tâches à faire, en cours et terminées.

Voici le lien du trello :

<https://trello.com/invite/b/FkijbBxQ/ATTIb9f3b1eb467a67f092d27b3be6b72ee08DDBD9D2/p/rojet-tutore-s6>

Pour ce qui est de la répartition des tâches, voici les tâches effectuées par chacun des membres du groupe.

Matthieu:

- Script insertion des données CSV vers Elasticsearch.
- Point d'API pour:
  - récupérer tous les jeux.
  - récupérer un jeu en fonction de son ID.
  - trier la liste des jeux en fonction d'un champ donné et d'un ordre donné.
  - rechercher un jeu dans la liste des jeux (fuzzy search).
  - récupérer le nom des jeux d'une recherche pour l'auto-complétion.
  - créer un utilisateur.
  - permettre la connexion d'un utilisateur.
  - ajouter un jeu à la liste des favoris de l'utilisateur connecté.
  - consulter la liste des favoris de l'utilisateur connecté.
- Page de consultation de la liste des jeux ajoutés en favoris.

Noé:

- Page avec scroll infini pour afficher la liste des jeux.
- Barre de recherche avec affichage des résultats et avec auto-complétion.
- Input select pour les tris avec affichage en scroll infini.
- Barre de navigation.
- Page détails d'un jeu.

Jérémy:

- Page de connexion utilisateur.
- Page d'inscription utilisateur.
- Personnalisation de la barre de navigation lorsqu'un utilisateur est connecté ou non.