Sorbonne Université

Rapport du projet de Programmation Concurrente Réactive, Répartie et Réticulaire

Julien Fang 28605540

Jean-Marc Zhuang 28605493

26 mai 2024

Table des matières

1 Introduction

Dans le cadre de ce projet, nous avons décidé de développer un quiz sur le thème des animés. L'application se distingue par sa variété de quiz, permettant ainsi aux utilisateurs de tester et d'affiner leurs connaissances. Il existe 4 types de quiz, un quiz avec des questions qui tourne autour du synopsis, un autre autour du genre, un autre en fonction d'une image, et pour finir le quiz général est un mixte des 3 quiz évoqués. De plus, les utilisateurs ont la possibilité d'épingler des animés lors du quiz pour pouvoir garder en mémoire l'animé qui a su capter la curiosité de l'utilisateur.

2 Présentation Générale

2.1 Choix de l'API et son utilisation

L'API choisie est : https ://kitsu.docs.apiary.io/. Cette API offre une vaste base de données d'animés, y compris les plus récents. Elle permet de récupérer diverses informations, telles que le nom de l'animé, un synopsis, les votes de ranking, son rang, etc. Pour récupérer les informations sur un animé, il suffit d'avoir son nom ou son ID. Il est également possible de filtrer les résultats pour affiner la recherche.

Cette API a été principalement utilisée lors de la génération d'un quiz, les informations telle que la date de sortie, le synopsis, le titre, ainsi que l'url pour afficher l'image de l'animé sont au coeur de nos quiz. Mais aussi pour afficher les animés populaire du moment ainsi que les derniers animés en date.

2.2 Les fonctionnalités

- Faire un quiz :
— L'utilisateur se connecte à la plateforme
— Il selectionne un des quatres quiz disponibles
— Il répond aux 10 questions du quiz
— A la fin, il est renvoyer sur la page d'accueil et la note de son quiz est ajouté dans ses statistiques.

- Reprendre un quiz non terminé :
— L'utilisateur se connecte à la plateforme
— Le serveur lui indique que le quiz général n'est pas terminé.
— L'utilisateur clique sur "Quiz Général", et il est envoyé à la question où il s'est arreté.

- Visualisation :
— Une liste des animés tendances est affiché pour l'utilisateur
— Une liste des dernières sorties est affiché pour l'utilisateur
— Dans chaque quiz lancés, l'utilisateur peut voir le tableau des scores

3 Base de données

User :
— id :string
— pseudo :string
— connexion :Object
  — login :string
  — password :string
  — connected :boolean

Collection pictureQuiz - synopsisQuiz - genreQuiz - generalQuiz
— id :string
— idjoueur :string
— mark :int
— finished :boolean
— number_question :int
— quizcollection :array(10) of Object
  — question :string
  — bonnereponse :string
  — anime :string
  — image :string
  — repjoueur :string
  — valide :boolean
  — reponsepossible :array(4) of string

Collection statsUser :
— id :string
— idjoueur :string
— nbquizgeneral :int
— notetotalgeneral :int
— nbquizgenre :int
— notetotalgenre :int
— nbquizsynopsis :int
— notetotalsynopsis :int
— nbquizpicture :int

— notetotalpicture :int

Collection Pin :
— id :string
— idjoueur :string
— slugs (liste de nom d'animé) :array of strings

4 Serveur

4.1 Description des requêtes

Le serveur retourne une réponse JSON contenant son id, sa date de création, un synopsis, ses titres (dans differents pays), sa notation, etc ...Cela dépend de la requete.
La réponse JSON renvoyée est assez conséquente, il est préférable de voir par soi-même en mettant l'URL sur le navigateur.

Les différentes requêtes utilisées :

1. Récupérer les informations concernant un animé a l'aide de son ID :
   https ://kitsu.io/api/edge/anime/**{id}**
   Par exemple en mettant :// https://kitsu.io/api/edge/anime/1 , l'animé devrait etre : Cowboy Bebop.
2. Récupérer les informations des animés en fonction du genre :
   https ://kitsu.io/api/edge/anime ?filter[genres]=**{genre}**
   Par exemple : https://kitsu.io/api/edge/anime?filter[genres]=action,fantasy Le serveur renvoie les 10 premiers animés en respectant les genres. Et pour récupérer la suite des animés du même genre, s'il existe, il faut récupérer le lien indiqué dans "next" à la fin de la reponse JSON.
3. Récupérer les informations d'un animé avec son titre :
   https ://kitsu.io/api/edge/anime ?filter[text]=**{titre}**
   Par exemple : https://kitsu.io/api/edge/anime?filter[text]=myheroacademia
   Il est également possible de le retrouver dans une autre langue si aucune faut d'orthographe n'est présente : https://kitsu.io/api/edge/anime?filter[text]=éĂšæŠČãĄőåůĺäžž (correspond a "Attack on titan")
4. Récupérer les informations d'un animé en fonction de sa note et du genre :
   https ://kitsu.io/api/edge/anime ?sort=**-averageRating**filter[genres]=**{genre}**
   Par exemple : https://kitsu.io/api/edge/anime?sort=-averageRating&filter[genres]=action,comedy Avec -averageRating permettant de prendre dans l'ordre décroissant.

5. Récupérer le top 10 tendances des animés :
   https://kitsu.io/api/edge/trending/anime

4.2 Ajax

— **GetTrendAnimeHandler :**
- Methode : GET
- URL : '/anime/trend'
- Description : Obtenir les animés tendances.

— **GetRecentAnimeHandler :**
- Methode : GET
- URL : '/anime/recent'
- Description : Obtenir les derniers animés sortis.

— **PostGeneralQuizHandler :**
- Methode : POST
- URL : '/anime/generalQuiz'
- Description : Créer un quiz général.

— **PostGenreQuizHandler :**
- Methode : POST
- URL : '/anime/genreQuiz'
- Description : Créer un quiz où l'utilisateur doit deviner le genre des animés.

— **PictureQuizHandler :**
- Methode : POST
- URL : '/anime/pictureQuiz'
- Description : Créer un quiz où l'utilisateur doit répondre aux questions à l'aide d'une image de l'animé.

— **SynopsisQuizHandler :**
- Methode : POST
- URL : '/anime/synopsisQuiz'
- Description : Créer un quiz où l'utilisateur doit répondre aux questions à l'aide du synopsis de l'animé.

— **UnfinishedQuiz :**
- Methode : GET
- URL : '/quiz/unfinishedQuiz'
- Description : Permet de connaître les quiz non terminés.

— **CreatePin :**
- Methode : POST
- URL : '/pin/create'
- Description : Epingler un animé durant un quiz.

— **DeletePin :**
- Methode : DELETE
- URL : '/pin/delete'
- Description : Désépingler un animé.

— **GetAllPin :**
- Methode : GET
- URL : '/pin/getAllPin'
- Description : Obtenir tous les animés épinglés.

5 Client

5.1 Choix des technologies

— **Bootstrap :** Cette bibliothèque a été utilisé pour arranger la page c'est à dire, placer les éléments en fonction du grid-system que cette bibliothèque propose, ce qui rend cette bibliothèque assez pratique pour cette usage.

— **React :** Cette bibliothèque est utilisé pour rendre la page reactive, le fait de recharger la page à chaque fois qu'une information change est pratique dans l'usage d'un quiz, par exemple passer à la prochaine question, voir les utilisateurs scores et les nombres de parties lancés de chaque utilisateurs qui ont terminé le quiz.

— **Go (Golang) :** Ce langage a été choisi pour le développement côté serveur en raison de sa performance. Go permet de créer une API RESTful rapide et capable de traiter un grand nombre de requêtes simultanément.

— **MongoDB :** Cette base de données NoSQL a été sélectionnée pour sa flexibilité et sa capacité à gérer des structures de données variées et dynamiques. Elle offre également des performances élevées pour les opérations de lecture et d'écriture, et permet de gérer efficacement de grandes quantités de données.

5.2 Plan

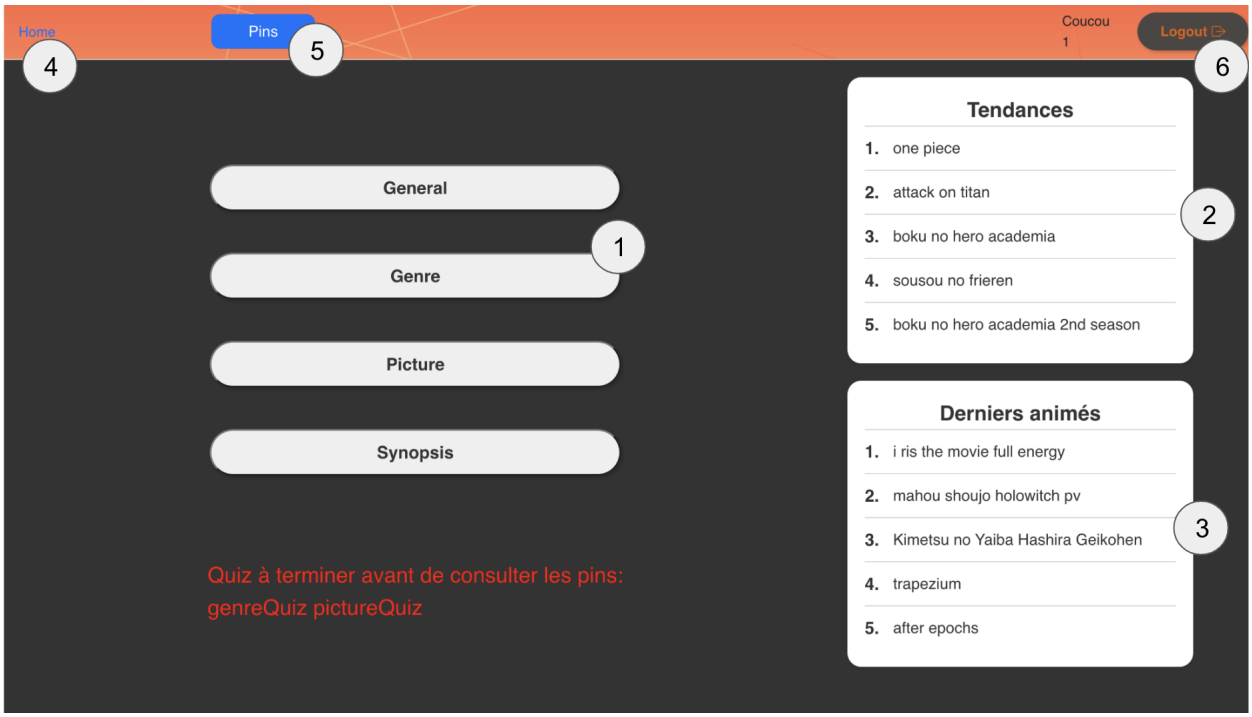

FIGURE 1 – Menu

(1) Selection du quiz à faire, s'il existe un quiz déjà existant l'utilisateur continuera là où il s'était arrêté sinon il devra attendre le temps de générer le quiz. Dans les 2 cas l'utilisateur sera aménée à la page des quiz et évaluer ses connaissances.

(2) Les animés populaires, le client fait appel au service se trouvant à l'url http ://localhost :8000//anime/trend et récupère les 5 premiers animés en tendances et les affiches sous forme de tableau.

(3) Pareil que les tendances mais avec les dernières sorties à l'url http ://localhost :8000/anime/recent.

(4) Le bouton Home se trouve dans la navbar qui est dans toutes les pages sauf quand l'utilisateur n'est pas connecté. Donc le bouton Home mène au menu. En étant déjà dans le menu, en cliquant sur ce bouton, l'utilisateur ne fait que recharger la page.

(5) Le boutons contenant tous les animés épingler de l'utilisateur, en cliquant dessus une fenêtre apparaîtra avec la liste de tous les animés épinglés, et bien sûr si

l'utilisateur décide de supprimer un élément de la liste, il le peut en cliquant sur le bouton supprimer

(6) Le bouton de déconnexion supprime tous les cookies et l'identifiant de l'utilisateur dans le stockage local, ce qui permettra à l'utilisateur de se déconnecter sans laisser de trace.

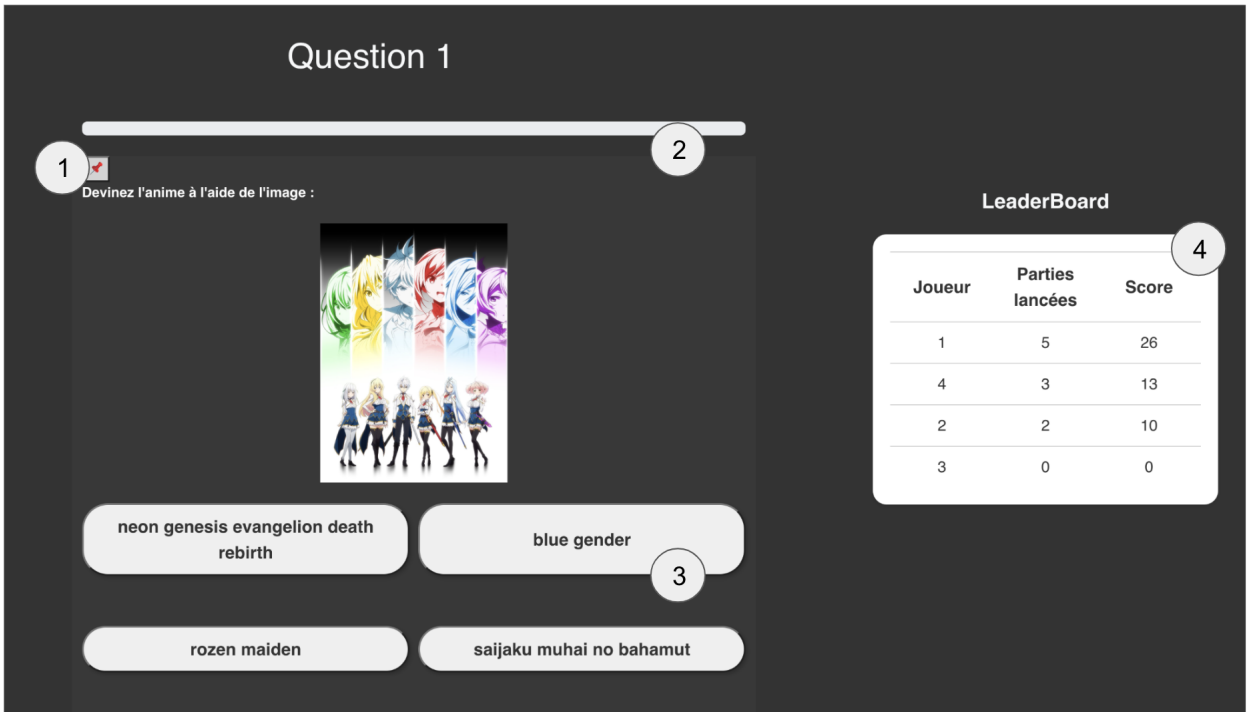

FIGURE 2 – Quiz

(1) Le boutons qui permet d'épingler un animé

(2) La barre de progression, pour que l'utilisateur sache combien de question il lui reste avant la fin

(3) Les réponses au quiz, en cliquant sur une des réponses, un appel est fait au serveur pour vérifier s'il la réponse de l'utilisateur est la bonne réponse

(4) Le Leaderboard qui est mise à jour qu'à la fin du quiz, malheureusement l'utilisateur doit relancer le même quiz pour voir où il se situe après l'ancien quiz

6 Conclusion

La réalisation de ce projet web de quiz sur les animés, utilisant une API en React et Go avec une base de données MongoDB, a été une expérience extrêmement enrichissante. C'était notre première fois que nous intégrions une API dans un site web, ce qui nous a permis de développer nos compétences en développement web de manière significative.

React nous a permis de développer une interface utilisateur réactive et intuitive, tandis que Go a assuré un traitement rapide et efficace des requêtes grâce à sa gestion native de la concurrence. MongoDB a offert une flexibilité et une rapidité d'accès aux données cruciales pour notre application.

Malgré les défis liés à l'intégration des technologies et à la gestion des états complexes, nous avons su surmonter ces obstacles grâce à des ajustements minutieux et des tests rigoureux. La sécurité des données a été garantie par la mise en place de mesures spécifiques, telles que l'utilisation de tokens d'authentification.

En résumé, ce projet nous a permis de gagner une expérience inestimable et de poser des fondations robustes pour de futurs développements, tant en termes d'ajout de nouvelles fonctionnalités que d'améliorations techniques.