

Date : 03 février 23h59 sur mon mail hugo.knorr@gmail.com

Rendu : application web fonctionnelle + dépôt Git accessible par mon compte hugo.knorr@gmail.com

Technologies : React

Objectif du projet

Vous devez développer une application de **quiz interactif** en React à partir de données récupérées depuis une API externe.

Ce projet a pour objectif d'évaluer votre compréhension des fondamentaux de React :

- gestion du state
 - effets secondaires (`useEffect`)
 - appels API asynchrones
 - logique de rendu conditionnel
 - découpage logique de l'application
-

Source des questions (obligatoire)

Les questions doivent être récupérées depuis l'API [Open Trivia Database](#).

- API publique
- gratuite
- sans clé d'authentification

Exemple d'endpoint

```
https://opentdb.com/api.php?amount=10&type=multiple
```

Chaque question fournie par l'API contient :

- une catégorie
- une difficulté
- une question
- une réponse correcte
- plusieurs réponses incorrectes

Les données reçues doivent être **traitées et adaptées** à l'application (notamment le mélange des réponses).

Fonctionnalités attendues

1. Écran de démarrage

L'application doit proposer :

- un bouton « Commencer le quiz »
- un choix du nombre de questions (ex : 5 ou 10)
- un choix de catégorie
- un choix de difficulté

Ces choix doivent influencer l'appel à l'API.

2. Récupération des questions

- Les questions sont récupérées **au lancement du quiz**
 - Un indicateur de chargement doit être affiché pendant la récupération des données
 - Les erreurs de récupération doivent être gérées (message simple suffisant)
-

3. Déroulement du quiz

- Les questions sont affichées **une par une**
 - Chaque question affiche :
 - l'énoncé
 - **4 réponses possibles** (1 correcte + 3 incorrectes)
 - L'ordre des réponses doit être **mélangé**
 - Une seule réponse peut être sélectionnée
 - Une fois une réponse choisie, l'utilisateur peut passer à la question suivante
-

4. Gestion du score

- Le score est calculé en comparant la réponse sélectionnée à la réponse correcte fournie par l'API
- Le score est mis à jour après chaque question

- Une indication de progression est affichée
(ex : Question 2 / 10)
-

5. Timer par question

- Chaque question dispose d'un temps limité (ex : 15 secondes)
- Le temps restant est affiché
- Lorsque le temps est écoulé :
 - la question est considérée comme incorrecte
 - la question suivante s'affiche automatiquement

Le timer doit être correctement nettoyé afin d'éviter toute accumulation ou accélération.

6. Écran de fin

À la fin du quiz, l'application doit afficher :

- le score final
 - un message adapté au score obtenu
 - un bouton permettant de relancer un nouveau quiz
-