

**Date: 014/04/2025 - 18/04/2025**

## SPRINT 4 BRIEF 2

### **Brief 2 : Application de Quiz avec API Trivia et suivi de progression**

#### **Contexte professionnel**

404.js souhaite développer une **application de quiz interactive** permettant aux utilisateurs de tester leurs connaissances grâce à l'**API Trivia**. L'objectif est de **suivre la progression des joueurs** via une barre de progression et de garantir une expérience utilisateur fluide et personnalisée en **enregistrant le nom du joueur en local**.

L'application devra être développée avec **React et TypeScript**, en assurant une **gestion optimisée des états** et une **interface responsive et fluide**.

---

#### **Fonctionnalités attendues**

- **Page d'accueil avec saisie du nom** (obligatoire pour commencer le quiz).
- **Stockage du nom en localStorage** pour permettre à l'utilisateur de reprendre sa partie.
- **Affichage des questions** provenant de l'**API Trivia**.
- **Navigation entre les questions** et validation des réponses.

- **Suivi de la progression** avec une **barre de progression** dynamique.
  - **Affichage des résultats** à la fin du quiz.
  - **Interface moderne et responsive** avec **Tailwind CSS**.
- 

### Architecture technique

- **React avec TypeScript** pour un typage strict et une meilleure maintenabilité.
  - **Gestion de l'état** avec `useState` et `useEffect`.
  - **Stockage local** (`localStorage`) pour sauvegarder le joueur et sa progression.
  - **Consommation de l'API Trivia** via `fetch` ou `axios`.
  - **Composants réutilisables** pour les questions, réponses et la progression.
  - **React Router** pour gérer la navigation entre les écrans.
- 

### Modalités pédagogiques

- **Travail** : individuel.
- **Durée** : 5 jours (7/04/2025 - 11/04/2025).

## Modalités d'évaluation

Une durée de 30 min organisée comme suit :

- Présenter une défense publique de son travail.
  - Démontrer le contenu et les fonctionnalités de l'application
  - Code Review et questions
  - Mise en situation
- 

## Livrables

1. **Code source** (GitHub).
  2. **Maquette de l'application** (Figma ou Adobe XD).
  3. **Lien de planification** (Trello, Jira, etc.)
1. **Présentation du projet** (Canva, PowerPoint).
- 

## Critères de performance

- ✓ **Nom obligatoire** avant d'accéder au quiz.
  - ✓ **Stockage et récupération du joueur** via localStorage.
  - ✓ **Affichage dynamique des questions avec API Trivia.**
  - ✓ **Barre de progression mise à jour en temps réel.**
  - ✓ **Navigation fluide et interface responsive** avec Tailwind CSS.
  - ✓ **Code propre, bien organisé et typé avec TypeScript.**
-

## Ressources utiles

- 📌 Trivia API → [https://opentdb.com/api\\_config.php](https://opentdb.com/api_config.php)
- 📌 React JS → <https://react.dev/>
- 📌 TypeScript → <https://www.typescriptlang.org/>
- 📌 React Router → <https://reactrouter.com/>
- 📌 Tailwind CSS → <https://tailwindcss.com/>
- 📌 Axios → <https://axios-http.com/>

**les compétences techniques visées : C1N2, C2N2, C3N2, C4N2.**

**les compétences transversales visées : CT1N2, CT4N2,  
CT6N2,CT8N2.**