

Projet fil-rouge – Application de Quiz

(Cours : Développement Web – TypeScript / Node.js / Express)

1. Description générale

L'objectif du projet est de concevoir et développer une **application web de quiz** permettant :

- aux utilisateurs de **s'inscrire et se connecter**,
- de **créer, modifier et supprimer leurs propres quiz**,
- de **parcourir et jouer aux quiz publiés par les autres utilisateurs**.

Chaque quiz est composé de :

- un **titre** et une **courte description**,
- un ensemble de **questions** (QCM par défaut, possibilité de vrai/faux ou réponses libres si développement assez rapide),
- une **correction** et un **score automatique**.

2. Fonctionnalités attendues

Front-end (interface utilisateur)

- **Inscription et authentification** (formulaire d'inscription, connexion/déconnexion, gestion de session).
- **Création, modification et suppression** d'un quiz par son auteur.
- **Ajout, édition et suppression** de questions à l'intérieur d'un quiz.
- **Navigation publique** :
 - lister les quiz disponibles,
 - rechercher par titre, par auteur ou par mot-clé,
 - jouer à un quiz existant et recevoir le score en fin de partie.

Back-end (API et logique)

- **CRUD complet** (Create, Read, Update, Delete) pour les entités principales :
 - utilisateurs,
 - quiz,
 - questions.
- Gestion de l'authentification (sessions ou token simple).
- Points d'API sécurisés (certaines routes accessibles uniquement aux utilisateurs connectés).

3. Contraintes techniques

- **Langages et outils :**
 - Développement en **TypeScript**.
 - Interface en **HTML/CSS** (libre choix de framework CSS si souhaité).
 - **Node.js + Express** pour le serveur.
- **Données :**
 - Persistance dans des **fichiers JSON** (une base par entité : utilisateurs, quiz, questions).
 - Les opérations CRUD doivent lire/écrire proprement dans ces fichiers.
- **Architecture :**
 - Respecter un **modèle MVC** (Models, Views si nécessaires, Controllers, Routes).
 - Découpage clair du code en modules.
- **Organisation du travail :**
 - Groupes de **trois étudiants** (un binôme accepté si besoin).
 - Utilisation de **Git** et **GitHub/GitLab** pour le suivi et la collaboration.
 - Maintien d'une **feuille de route** (issues, to-do list ou équivalent) et d'un journal de suivi.

4. Critères d'évaluation

- **Cahier des charges initial**
- **Respect des fonctionnalités demandées**
- **Schéma des données**
- **Qualité du code**
- **Architecture du projet**
- **Sécurité**
- **Fonctionnalités CRUD**
- **Gestion du projet et suivi**
- **Documentation**
- **Présentation finale**