

Cahier des Charges - Site de Quiz

Objectif :

- Développer un site de quiz simple et fonctionnel.
- Permettre aux professeurs de créer et gérer des quiz.
- Offrir aux élèves la possibilité de passer les quiz et consulter leurs résultats.
- Se concentrer sur les fonctionnalités essentielles (MVP).
- Garantir une expérience utilisateur fluide et efficace.

Fonctionnalités

Authentification Simple

- **Page de connexion** : email + mot de passe
- **Page d'inscription** : nom, email, mot de passe, type (prof/élève)
- **Session PHP basique** avec logout

Interface Professeur

- **Créer un quizz** : titre + jusqu'à 25 questions maximum
- **Types de questions** : QCM (choix unique) ET questions à réponse libre
- **Publier le quizz** : bouton on/off
- **Voir les résultats** : liste simple des scores
- **Définir le timer** : durée personnalisable pour chaque quizz

Interface Élève

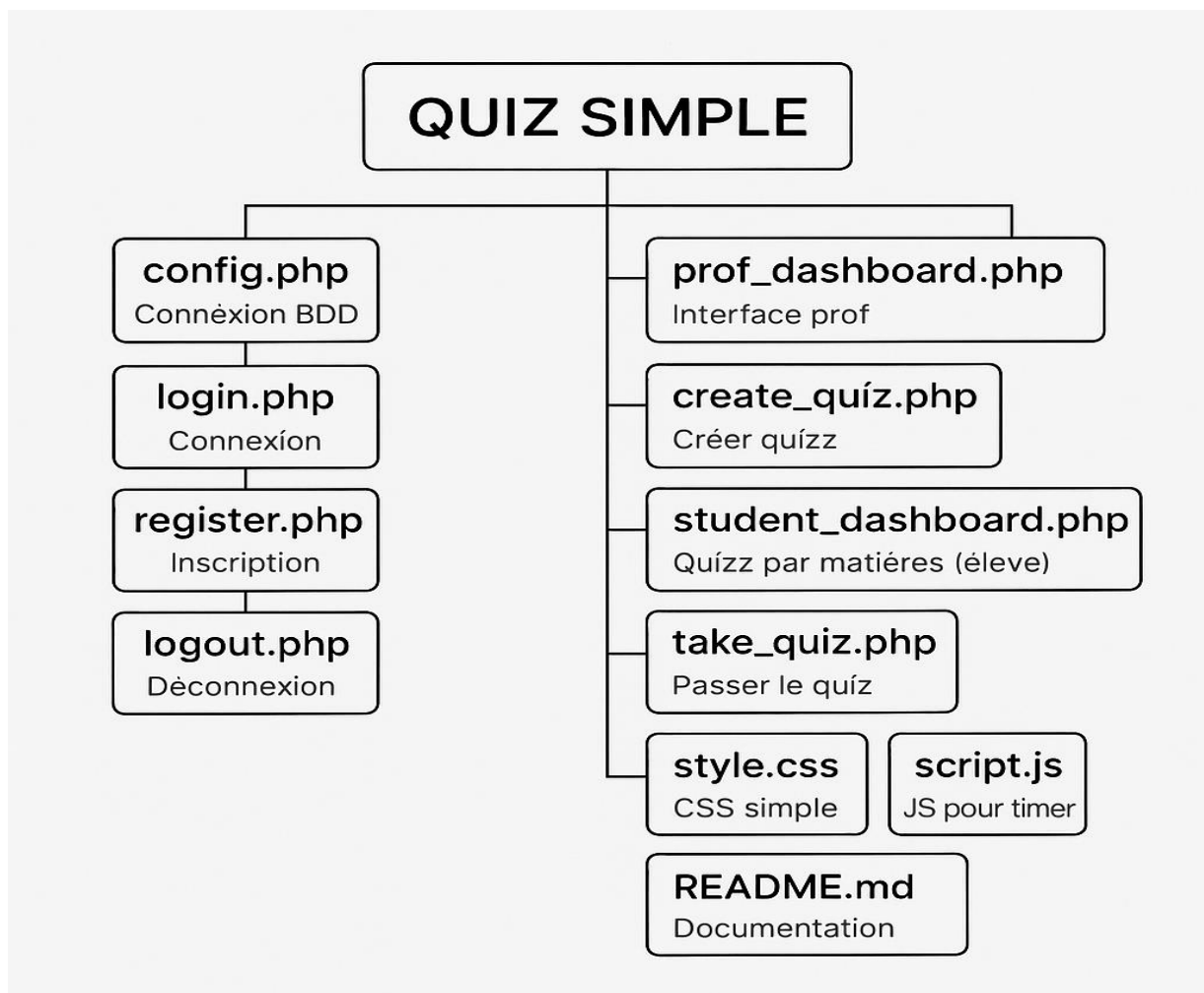
- **Voir les quizz par matière/module** : organisation par catégories
- **Sélectionner une matière** : accès aux quizz de cette matière
- **Répondre au quizz** : une question à la fois (QCM ou réponse libre)

- **Timer personnalisé** : défini par le professeur
- **Voir son score** : après soumission

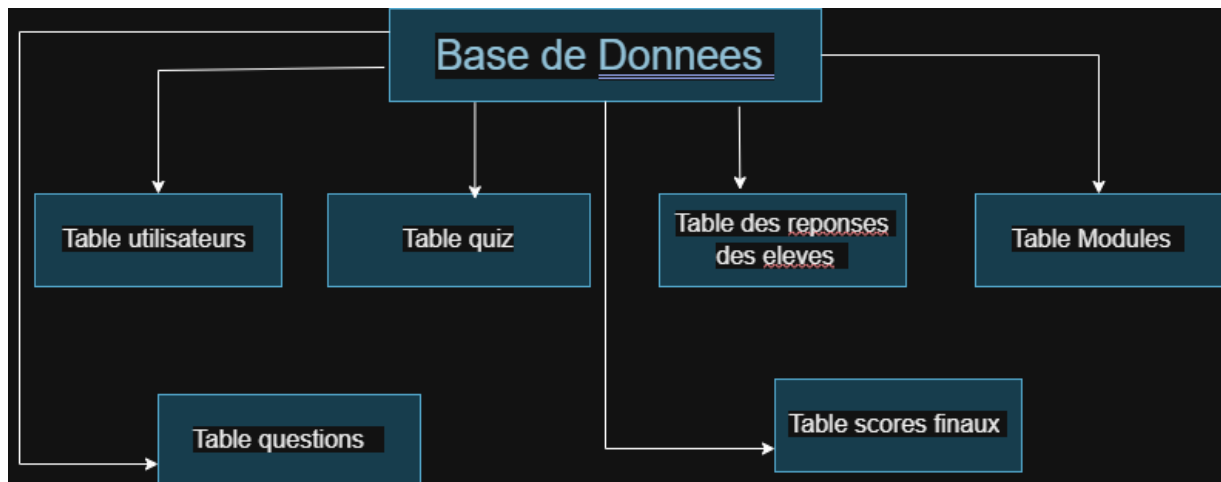
Anti-triche

- **Timer personnalisé** : défini par le professeur (1-60 minutes par quiz)
- **Soumission automatique** : à la fin du temps défini
- **Une seule tentative par élève**

Structure Simple



Base de Données :



Planning :

Semaine 1

- **Jour1-2 : Base technique:**
 - Installation XAMPP/WAMP
 - Création base de données
 - config.php (connexion BDD)
- **Jour3-4: Authentification:**
 - login.php
 - register.php
 - Système de sessions
- **Jour 5-7: Interface Professeur:**
 - prof_dashboard.php (liste des quizz)
 - create_quiz.php (formulaire avec QCM et questions libres)
 - Gestion des matières/modules

Semaine 2

- **Jour 1-2: Interface Élève:**
 - subjects.php (sélection de matière)

- student_dashboard.php (quizz par matière)
 - take_quiz.php avec timer personnalisé et types de questions
 - results.php
- **Jour 3-4: Finitions:**
 - CSS basique pour rendre présentable
 - Tests et corrections bugs
 - Documentation/README

Technologies à Utiliser

- PHP pur
- MySQL avec PDO
- HTML/CSS/JS
- Bootstrap pour le CSS

Anti-triche :

- Timer personnalisé : défini par le professeur
- Soumission automatique : quand timer = 0
- Une tentative unique : vérification en BDD
- Questions mélangées : ORDER BY RAND() en SQL

Ce qui est abandonné ou reporté à la fin:

- Système de classes/groupes
- Interface admin
- Statistiques avancées
- Mesures anti-triche sophistiquées
- Design complexe
- Responsive design parfait
- Validation côté client poussée
- Sécurité avancée (CSRF, etc.)

Objectif Final

- Un site qui fonctionne localement
- Professeurs peuvent créer des QCM ET questions libres (max 25 questions)
- Système de matières/modules pour organiser les quizz
- Élèves peuvent naviguer par matière puis choisir leurs quizz
- Timer personnalisable par le professeur
- Scores calculés automatiquement (correction manuelle pour questions libres)
- Interface basique mais propre
- Code commenté et lisible

EVERY ONE OF OUR GROUP SHOULD do :

KATIA – Back-end & Base de Données (PHP + MySQL)

Katia est responsable de **la logique du site et de la gestion des données**. Elle doit principalement s'occuper de :

Back-end (PHP)

- `config.php` : connexion à la base de données avec PDO.
- `login.php`, `register.php`, `logout.php` : gestion de l'authentification.
- `prof_dashboard.php` : affichage des quiz du professeur.
- `create_quiz.php` : traitement de la création de quiz avec QCM et questions libres.
- `take_quiz.php` : gestion du quiz côté élève, y compris le timer.
- `results.php` : calcul et affichage des scores.
- Mise en œuvre du système de sessions (connexion/déconnexion).
- Vérification anti-triche (timer, soumission automatique, tentative unique).

Base de Données

- Création des tables :
 - utilisateurs

- modules
- quiz
- questions
- réponses des élèves
- scores finaux
- Requêtes SQL pour insertion, mise à jour, récupération des données.
- ORDER BY RAND() pour mélanger les questions.

ALICIA – Front-end & Interface (HTML/CSS/JS)

Alicia est responsable de **l'apparence visuelle et de l'interaction utilisateur**. Elle doit s'occuper de :

Interface utilisateur

- register.php, login.php, logout.php : design des pages d'inscription et de connexion.
- prof_dashboard.php, create_quiz.php : design de l'espace professeur.
- subjects.php, student_dashboard.php, take_quiz.php : pages côté élève pour navigation et prise de quiz.

HTML/CSS/JS

- Utilisation de Bootstrap pour un style simple et propre.
- style.css : mise en forme des pages.
- script.js : gestion du timer pendant les quiz.
- Organisation claire et intuitive des modules/matières.
- Responsive minimal (pas de design complexe exigé)

ET A LA FIN ON DOIT S'ASSURER QUE LE FRONT-END **COMMUNIQUE BIEN AVEC LE BACK-END** et on réalise des tests ensemble pour s'assurer que tous fonctionnent bien

