



Cahier des Charges



Projet : Plateforme d'examen en ligne

NB :

- ✓ Dans ce projet, il existe des notions que n'avez pas vu dans le cours et qui consiste une évaluation de recherche et d'autonomie chez l'étudiant.
- ✓ Le projet doit être réalisé en binôme ou trinôme
- ✓ Le projet doit être rendu sous forme de vidéo explicative de 5 minutes au maximum et du lien Github où le projet est hébergé.
- ✓ Le projet Github doit être partagé avec votre professeur
- ✓ Travaillez en collaboration sur Github (Votre professeur pourra voir les détails d'accès à vos comptes et vérifier si quelqu'un ne contribue pas avec ses membres de groupe.
- ✓ Ce projet est compté sur 25% de la note finale du module.
- ✓ A la fin de réalisation, rendez-vous sur le formulaire suivant pour déposer votre projet : <https://forms.gle/XNfD2WC5innrpZQy6>
- ✓ Le projet à rendre avant le 10 mai 2025

1. Contexte

Le projet consiste à développer une application web permettant aux enseignants de créer et de gérer des examens en ligne, et aux étudiants de s'y inscrire et d'y participer. Ce projet doit permettre aux étudiants de mettre en pratique les notions de développement web, de base de données, de gestion d'authentification et de collaboration via Git.

2. Objectifs pédagogiques

- Appliquer le développement full-stack.
- Maîtriser la gestion d'authentification (sessions, cookies, JWT).
- Gérer les formulaires complexes avec pièces jointes.
- Implémenter des fonctionnalités dynamiques avec timers et score automatique.
- Travailler en autonomie avec **GitHub Desktop** et respecter les bonnes pratiques de versioning.

3. Fonctionnalités à développer

3.1. Formulaire d'inscription (étudiants et enseignants)

- Champs obligatoires : email, nom, prénom, date de naissance, sexe, établissement, filière.
- Le formulaire est identique pour enseignants et étudiants (type d'utilisateur sélectionné).

- **Commit GitHub requis après finalisation** de ce formulaire (commit: ajout formulaire inscription).

3.2. Formulaire de connexion

- Authentification avec vérification de l'email et du mot de passe.
- Utilisation des **sessions et cookies** ou **JWT** pour maintenir la session utilisateur.
- **Commit requis après test de la connexion/déconnexion** (commit: ajout login + session/JWT).

3.3. Espace enseignant

Création d'examen :

- Titre
- Description
- Public ciblé (ex: 2e année MIP, S4, groupe A)
- Génération automatique d'un **lien unique d'accès**
- **Commit requis après création de l'interface + génération lien** (commit: interface création examen)

Ajout de questions :

- Deux types : **question directe** et **QCM**
- Interface pour choisir le type de question
- Ajout d'un **énoncé** avec possibilité de joindre **image, audio, ou vidéo**
- Pour les **questions directes** :
 - Saisie de la réponse
 - Choix d'un **taux de tolérance** (ex : 10%) pour erreurs de frappe ou casse
- Pour les **QCM** :
 - Saisie des options
 - Sélection des bonnes réponses par case à cocher
- Affectation d'une **note** pour chaque question
- Définition d'une **durée (en secondes)** pour chaque question
- **Fonctionnalités de modification et suppression** des questions
- **Commit requis après chaque fonctionnalité terminée** :
 - (commit: ajout QCM)
 - (commit: ajout question directe avec tolérance)
 - (commit: ajout gestion durée + note)

3.4. Espace étudiant

- Accès à la plateforme via le lien partagé
 - Obligation de s'inscrire et de se connecter
 - Avant le démarrage de l'examen, une demande d'**activation de la géolocalisation** est affichée et les coordonnées sont enregistrées
 - Les questions s'affichent **une par une** avec le timer prédéfini
 - À la fin de l'examen, un **score sur 100** est affiché
 - **Commit requis après chaque étape** :
 - (commit: affichage questions enchaînées avec timer)
 - (commit: enregistrement géolocalisation)
 - (commit: calcul score et affichage)
-

4. Contraintes techniques

- Base de données relationnelle ou NoSQL au choix (selon technologies maîtrisées)
- Authentification sécurisée
- UI claire et responsive
- Respect du modèle MVC ou architecture modulaire
- **Utilisation de GitHub Desktop obligatoire :**
 - Un **commit + push** après chaque fonctionnalité.
 - Le **nom de la fonctionnalité** doit être mentionné clairement dans le message du commit.