

# Programmation WEB

## Projet 2023

**Important :** En fin du sujet l'ordre d'exécution des tâches est donné. Il est important de le suivre, car l'évaluation sera fondée sur l'exécution consécutive des tâches. Le barème préliminaire est donné en fin de feuille.

On vous propose de créer un site pour gérer les plannings dans une formation. On simplifiera le sujet en négligeant la différence entre les CMs et les TDs/TPs – il n'y aura qu'une seule catégorie – le cours qui sera suivi par tous les étudiants inscrits dans une formation.

Le site comporte 3 acteurs :

- a. Les étudiants : ils peuvent s'inscrire pour des cours de leur formation et voir leur planning personnalisé.
- b. Les enseignants : Ils sont responsables du cours et peuvent déplacer ou créer des séances (entrées) dans le planning. Ils peuvent également voir leur planning personnalisé.
- c. L'administrateur : Il fait les tâches de gestion – création/modification des cours, formations, enseignants et étudiants. Il peut également faire toutes les tâches des étudiants et des enseignants, ainsi que voir le planning pour tout le monde.

Le site doit permettre d'effectuer les opérations suivantes:

1. Pour les étudiants :
  - 1.1. Voir la liste des cours de la formation (dans laquelle l'étudiant est inscrit).
  - 1.2. Gestion des inscriptions :
    - 1.2.1. Inscription dans un cours.
    - 1.2.2. Désinscription d'un cours.
    - 1.2.3. Liste des cours auxquels l'étudiant est inscrit.
    - 1.2.4. Rechercher un cours dans la liste des cours de la formation.
  - 1.3. Affichage du planning personnalisé (uniquement les séances des cours auxquels cet étudiant est inscrit) :
    - 1.3.1. Intégral.
    - 1.3.2. Par cours.
    - 1.3.3. Par semaine.
2. Pour les enseignants :
  - 2.1. Voir la liste des cours dont on est responsable.
  - 2.2. Voir le planning personnalisé (les séances dont on est responsable) :
    - 2.2.1. Intégral.
    - 2.2.2. Par cours.
    - 2.2.3. Par semaine.
  - 2.3. Gestion du planning :
    - 2.3.1. Création d'une nouvelle séance de cours.
    - 2.3.2. Mise à jour d'une séance de cours.
    - 2.3.3. Suppression d'une séance de cours.
    - 2.3.4. Utiliser 2 vues différentes pour les opérations ci-dessus (par cours et par semaine).
3. Pour l'utilisateur (étudiant ou enseignant) :
  - 3.1. Gestion du compte :
    - 3.1.1. Création du compte.
    - 3.1.2. Changement de son mot de passe.
    - 3.1.3. Modification du nom/prénom.
4. Pour l'administrateur :
  - 4.1. Gestion des utilisateurs :
    - 4.1.1. Liste :
      - 4.1.1.1. Intégrale.
      - 4.1.1.2. Filtre par type (étudiant/enseignant).
      - 4.1.1.3. Recherche par nom/prénom/login.
    - 4.1.2. Acceptation (ou refus) d'un utilisateur qui a été auto-crée.
    - 4.1.3. Association d'un enseignant à un cours (même point que 4.2.6).
    - 4.1.4. Création d'un utilisateur.
    - 4.1.5. Modification d'un utilisateur (y compris le type).
    - 4.1.6. Suppression d'un utilisateur.

- 4.2. Gestion des cours :
  - 4.2.1. Liste.
  - 4.2.2. Recherche par intitulé.
  - 4.2.3. Création d'un cours.
  - 4.2.4. Modification d'un cours.
  - 4.2.5. Suppression d'un cours.
  - 4.2.6. Association d'un enseignant à un cours (même point que 4.1.3).
  - 4.2.7. Liste des cours par enseignant (similaire à 2.1, mais pour n'importe quel enseignant).
- 4.3. Gestion des formations :
  - 4.3.1. Liste
  - 4.3.2. Ajout d'une formation.
  - 4.3.3. Modification d'une formation.
  - 4.3.4. Suppression d'une formation.
- 4.4. Gestion des plannings :
  - 4.4.1. Tous les points du 2.3, mais pour n'importe quel enseignant.

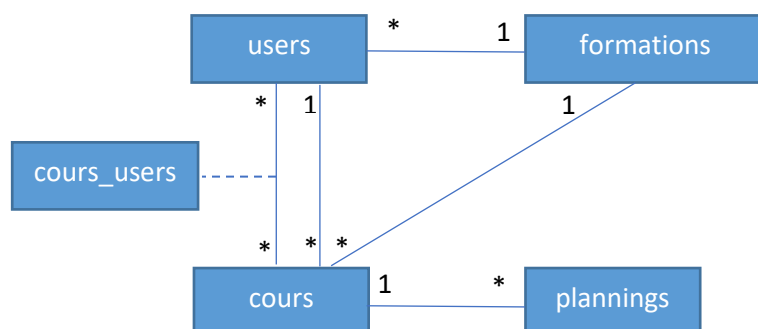
#### Notes :

- Pour simplifier le projet, il existe un seul endroit qui stocke les identifiants des acteurs (la table users). Par conséquent il faut vérifier le type pour les discriminer, afficher les bonnes pages et vérifier les droits.
- Le compte administrateur (admin:admin) est précréé dans la base initiale.
- Pour créer un compte la procédure est la suivante : l'utilisateur remplit le formulaire en indiquant entre autre sa formation (pour les enseignants et les admins – un item spécial serait utilisé). Ces informations seront inscrits dans la table users (avec le type égal à null). Ensuite, l'administrateur valide cette demande et maj le type. Donc un type=null indique que l'utilisateur est inactive et ne peut pas encore accéder au site.
- Pour un enseignant ou admin le champs formation\_id sera null.
- Un admin peut contourner la procédure ci-dessus, en créant directement un enregistrement.
- Afin de mieux tester les différents cas d'utilisation, utilisez des Seeders pour remplir la BD avec des données factices.
- La base de données suit les conventions de nommage de Laravel (excepté le cours).

#### La base de données et le schéma des relations :

- users(id,nom,prenom,login,mdp,formation\_id,type)
- formations(id,intitule,created\_at,updated\_at)
- cours(id,intitule,user\_id,formation\_id,created\_at,updated\_at)
- cours\_users(cours\_id,user\_id)
- plannings(id,cours\_id,date\_debut,date\_fin)

#### Schéma des relations :



#### Ordre des tâches :

Ordre	1	2	3	4	5	6
Nom groupe	Création utilisateurs	Ajout/liste formations/cours	Inscriptions	Planning	Admin1	Admin2
Tâches	3 4.1.2 4.1.3	4.2.1 4.2.3 4.3.1 4.3.2	1.1 1.2 2.1	1.3 2.2 2.3	4.2 4.3 4.4	4.1
Points	2	3	3	7	3	2