



**Filière : DDOWFS**

**Niveau : TS**

**Durée : 2h30**

**Epreuve : Théorique (variante 1)**

**Barème : /40**

Le but de cet exercice est de créer une application de gestion de **livres** pour une bibliothèque en ligne !  
L'application sera utilisée par une bibliothèque en ligne pour gérer son catalogue de **livres**.

Vous serez amené à créer un système permettant d'ajouter, de modifier et de supprimer des **livres**, ainsi que de les organiser par **catégories**.

De plus, vous devrez mettre en place des fonctionnalités de validation des données pour assurer la cohérence des informations saisies par les utilisateurs. Vous aurez également l'opportunité d'explorer l'utilisation des middlewares pour sécuriser l'accès aux fonctionnalités de gestion des **livres**.

Voici un schéma simplifié des tables "**Livre**" et "**Catégorie**" :

**Table "Livre" :**

- id (clé primaire)
- titre (varchar)
- pages (decimal)
- description (texte)
- image (varchar)
- categorie\_id (clé étrangère vers la table "**Catégorie**")

**Table "Catégorie" :**

- id (clé primaire)
- nom (varchar)
- description (texte)

La table "**Livre**" est liée à la table "**Catégorie**" par le biais de la colonne "categorie\_id", qui est une clé étrangère faisant référence à l'ID de la **catégorie** correspondante dans la table "**Catégorie**".

**Travail à faire :**

1. Écrire la commande artisan pour créer les modèles "**Livre**" et "**Catégorie**".
2. Donner les lignes de code à ajouter au modèle "**Livre**" afin que tous les attributs puissent être affectés lors de la création ou de la mise à jour de plusieurs instances du modèle en même temps. (Indication : utiliser la propriété "**fillable**").
  - i. Donner le code de la méthode "**livres**" à ajouter au modèle "**Catégorie**".  
Grâce à cette méthode, nous pouvons accéder à la collection des "**livres**" associés à une "**catégorie**" en accédant à la propriété "**livres**".

3. Donner le code de la méthode **"up"** du fichier de migration correspondant au modèle **"Livre"**.  
(Ajouter aussi les colonnes **"created\_at"** et **"updated\_at"**).
4. Écrire la commande artisan pour créer le contrôleur **"LivreController"**.

Dans la classe **"LivreController"**, créer les méthodes suivantes :

5. **"index"** qui renvoie la vue **"index"** avec la liste des livres.
6. **"create"** qui renvoie la vue **"create"** avec le formulaire pour ajouter un livre.
7. **"store"** pour sauvegarder les données du formulaire.
8. **"edit"** qui renvoie la vue **"edit"** avec les informations d'un livre spécifique.
9. **"update"** pour confirmer la mise à jour d'un livre.
10. **"destroy"** pour supprimer un livre.
11. Créer les routes pour toutes les méthodes ci-dessus du contrôleur **"LivreController"**.
12. Écrire le code de la vue **"index"** qui permet :
  - a. D'étendre le layout **"layouts.app"**.
  - b. D'afficher tous les **livres** dans un tableau dans la section blade : **"content"**.
13. Écrire le code de la colonne **"Actions"** contenant les boutons **"ajouter"**, **"éditer"** et **"supprimer"** permettant d'appeler les routes correspondantes.
14. Ajouter des règles de validation pour le formulaire de création de **livre** :
  - a. Le champ **"titre"** doit être requis et avoir une longueur maximale de 255 caractères.
  - b. Le champ **"pages"** doit être requis et être un nombre entier positif.
  - c. Le champ **"description"** doit être requis.
  - d. Le champ **"catégorie"** doit être requis et correspondre à une **catégorie** existante.
15. Écrire la commande artisan qui crée le middleware **"AuthMiddleware"** qui vérifie si l'utilisateur est authentifié avant d'accéder aux fonctionnalités de gestion des **livres**.
16. Appliquer le middleware **"AuthMiddleware"** sur les routes du contrôleur **"LivreController"**.
17. Ajouter une fonctionnalité supplémentaire : la possibilité de télécharger une image pour chaque **livre**.
  - a. Écrire la commande artisan permettant d'ajouter une migration pour ajouter une colonne **"image"** à la table des **livres**.
  - b. Donner le code des fonctions **"up"** et **"down"** de la migration. Sachant que le champ **image** peut être null.
  - c. Écrire le code pour gérer le téléchargement de l'image dans la méthode **"store"** du contrôleur **"LivreController"**.
  - d. Écrire le code pour afficher l'image associée à chaque livre dans la colonne **"image"** du tableau de la question 13 de la vue **"index"**.

**Barème :**

Question	Points
1	2
2	2
3	2
4	2
5	2
6	2
7	2
8	2

Question	Points
9	2
10	2
11	2
12	2
13.a	2
13.b	2
14	2
15.a	1

Question	Points
15.b	1
15.c	1
15.d	1
16	1
17	1
18.a	1
18.b	1
18.c	1
18.d	1