



EXAMEN DE FIN DE MODULE RÉGIONAL

Au titre de l'année : 2022/2023

DIRECTION REGIONALE FES-MEKNES	
Année de formation : 2A	Épreuve : <input checked="" type="checkbox"/> Théorique <input type="checkbox"/> Pratique <input type="checkbox"/> Synthèse
Niveau de formation : TS	
Filière : Développement digital – Option web full stack	Durée : 02h30
	Barème : / 40 pts
Intitulé du module : Développement back end	Variante : V2

Exercice 1 : (4pts)

Indiquer la bonne réponse :

- Qu'est-ce que Composer dans Laravel ?
 - Un serveur web léger pour tester des applications
 - Un éditeur de code intégré
 - Un gestionnaire de dépendances pour PHP
- Quel est le rôle du Contrôleur dans l'architecture MVC de Laravel ?
 - Gérer la logique métier de l'application
 - Afficher les données de l'application à l'utilisateur
 - Stocker les données de l'application dans une base de données
- Qu'est-ce qu'un middleware dans Laravel ?
 - Une fonctionnalité pour gérer les cookies utilisateur
 - Une étape qu'on réalise avant la livraison finale d'un projet Laravel ;
 - Un mécanisme permettant de contrôler les requêtes http entrante à notre application ;
- Qu'est-ce que CMS signifie ?
 - Content Marketing System
 - Content Management System
 - Creative Media Studio

Exercice 2 : (6pts)

- Indiquer les rôles des directives blade suivantes : @csrf et @auth ? (2pts)
- Donner les commandes permettant de : (4pts)
 - Créer un nouveau projet Laravel ;
 - Créer un contrôleur de ressource ;
 - Lister toutes les routes ;
 - Annuler une migration ;

Exercice 3 : (30pts)

Un centre culturel organise et gère des événements culturels, tels que des pièces de théâtre, des concerts, des opéras, des ballets, etc

Le centre souhaite automatiser la gestion des réservations de ses adhérents pour assister aux évènements qu'il organise ;

Ainsi, on propose la base de données suivante :

types (**id**, titre, photo)

evenements (**id**, type_id, description, date, salle, nombre_max_places, created_at, updated_at)

users (**id**, nom, email, email_verified_at, password, remember_token, created_at, updated_at)

evenement_user (**evenement_id**, **user_id**, nombre_places, created_at, updated_at)

- Les champs en gras sont des clés primaires et les champs soulignés sont des clés étrangères ;
- "nombre_max_places" présente le nombre maximal des spectateurs pouvant assister à cet événement ;
- La table evenement_user est une table pivot permettant d'enregistrer les réservations des places, effectuées par les utilisateurs, pour assister à un événement donné ;
- Le champ « nombre_places » indique le nombre de places que veut réserver un utilisateur pour un événement donné ;

1. Création de la table « evenements » :

- Donner la commande pouvant générer le fichier de migration de création de la table « evenements » ; **(1pt)**
- Ecrire le code à ajouter à la méthode « up » de ce fichier, sachant que : **(4pts)**
 - Le champ « salle » ne peut contenir que l'une des valeurs suivantes : théâtre1, théâtre2, conférence1 ;
 - Le « nombre_max_places » est un entier positif et sa valeur par défaut est « 50 » ;

Dans la suite de cet exercice, on suppose que les migrations de création des différentes tables ont été bien exécuté ;

On souhaite ajouter à notre projet Laravel, la fonctionnalité de mise à jour de la table « evenements »

On suppose que les éléments ci-après sont déjà créés :

- Le modèle « Evenement »
- Le contrôleur de ressource « EvenementController » correspondant ;
- Une vue « layout.blade.php » sous le répertoire « views » (un champ @yield (« content ») a été ajouté pour que les autres vues puissent définir leur propre contenu).
- Un dossier « evenement » sous le répertoire « views »
- Les vues : index, create, edit et show sous le répertoire « views/eventement » (ces vues héritent de la vue modele)

-
- Indiquer l'instruction, à ajouter au fichier **web.php** (sous le répertoire **routes**), permettant de générer les routes de mises à jour de la table « evenements ». **(1pt)**
 - Sous le modèle « Evenement », configurer les champs concernés par l'affectation en masse (Mass Assignment) ; **(2pt)**
 - Ajouter au modèle « Evenement », la définition de la relation « evenement ⇔ type ». **(2pts)**
 - Lister tous les événements (en affichant les titres de leur type) comme le montre la capture ci-après :

Liste des événements

Nouvel événement

ID	Type	Description	Date	Salle	Nombre de places maximal	Action
1	Conférence	Conférence autour du sujet "Bienfaits du don du sang"	12/01/2023	conférence1	200	Afficher
2	Théâtre	Pièce théâtrale présentée par les élèves de l'école MOUMTAZ	28/06/2023	théâtre2	150	Afficher
3	Concert	Concert de l'artiste Maher Zain	17/08/2023	théâtre2	800	Afficher

- Donner le code de la méthode « **index** » du contrôleur « **EvenementController** » ; **(2pts)**
- Donner le code de la vue « **index.blade.php** » **(4pts)**

6. Ajouter un nouvel événement :

Nouvel événement

Type d'événement

Description

Date de l'événement

Salle Théâtre1 Théâtre2 Conférence1

Nombre maximal de places

Ajouter

Sachant que le type doit être choisi depuis une liste déroulante, affichant tous les types de la table « **types** » :

- Donner le code de la méthode « **create** » du contrôleur « **EvenementController** » ; **(2pts)**
- Donner le code de la méthode « **store** » du contrôleur « **EvenementController** », sachant que les données reçues auprès de la requête http, doivent être validées avant qu'elles soient transmises à la base de données, tel que : **(4pts)**
 - Le champ salle ne peut accepter que l'une des valeurs suivantes : théâtre1 ou théâtre2 ou conférence1
 - La description est une chaîne de caractères qui peut comporter au maximum 200 caractères ;
 - La date de l'événement doit être ultérieure à la date actuelle ;En cas de réussite, l'utilisateur doit être redirigé vers la vue index avec un message de succès ;
- Donner le code de la vue « **create** » permettant d'afficher un formulaire d'ajout d'un nouvel événement ; **(4pts)**

La page doit également traiter les éventuelles erreurs générées lors de la validation ;

7. Le bouton « Afficher » de la page « index » permettra d'afficher le détail d'un événement dans une autre vue « show » ;

Exemple du résultat souhaité :

Détails de l'événement numéro 3

Type : Concert
Date: 17/08/2023
Description : Concert de l'artiste Maher Zain

Salle: théâtre2
Nombre de places maximal: 800

Nombre de places réservées actuellement : 115

Liste des réservations

Nom de l'utilisateur	Nombre de places réservées
Souad Slaoui	3
Ahmed Alaoui	10
Association al majd	80
Bennani Taoufik	22

Sachant qu'on veut afficher, en plus des informations de l'événement, ses réservations en cours ainsi que le nombre total de places réservés actuellement comme le montre la capture ci-dessus ;

Donner le code de la méthode « **show** » du contrôleur « EvenementController » ; **(4pts)**

⇒ **Notez :**

- **On suppose que toutes relations entre les différents modèles ont été configurées ;**
- **Il n'est pas demandé de donner le code de la vue « show »**