

Examen de Fin de Module Régional

Module : Développement back-end	Variante N° : V1
Filière : Développement digital	Année : 2023/2024
Niveau : TS	Durée : 2h30 heures
	Barème : .../40

<i>Partie théorique (11 pts)</i>	<i>Note</i>
<p><u>Exercice 1 : (5pts)</u></p> <p>Trouver la bonne réponse :</p> <ol style="list-style-type: none"> Quel est le principe fondamental du concept "Migration" dans Laravel ? <ul style="list-style-type: none"> a. Gestion des sessions. b. Versionnement de la structure de la base de données. c. Configuration des routes. Quelle est la méthode utilisée pour définir une relation "Many-to-Many" dans Eloquent ? <ul style="list-style-type: none"> a. hasMany() b. belongsTo() c. belongsToMany() Quelle est la convention de nommage par défaut pour le contrôleur qui gère les opérations CRUD d'une entité "Departements" ? <ul style="list-style-type: none"> a. DepartementController. b. CRUDController c. DepartementCRUD Dans quel fichier est défini le modèle par défaut pour une table dans Laravel ? <ul style="list-style-type: none"> a. database/migrations b. app/Models c. config/database.php Quelle est la commande pour créer une nouvelle migration dans Laravel ? <ul style="list-style-type: none"> a. php artisan make:migration b. php artisan migration:make c. php artisan create:migration 	<p>(1 pt)</p> <p>(1 pt)</p> <p>(1pt)</p> <p>(1pts)</p> <p>(1pts)</p>
<p><u>Exercice 2 : (6 pts)</u></p> <p>Q1) Quelle est la différence entre les migrations et les seeders dans Laravel?</p> <p>Q2) Comment définir une relation "One-to-Many" entre deux modèles dans Laravel?</p> <p>Q3) Qu'est-ce que les middlewares dans Laravel et à quoi servent-ils ?</p>	<p>(2pts)</p> <p>(2pts)</p> <p>(2pts)</p>



Partie pratique (29 pts)	Note
Soit le modèle relationnel suivant : Départements (<u>Id</u> , NomDepartement) Employes (<u>Id</u> , NomEmploye, PrenomEmploye, Adresse, Telephone, #Departement_Id)	
Q1) Ecrire la commande pour créer un projet laravel nommé « gestion_des_employés »	(1pts)
Soit la base de données vide avec le nom Gestion_des_employés :	
Q2) Ecrire la configuration des informations de connexion de base de données (dans le fichier .env)	(2pts)
Q3) Ecrire les commandes artisan correspondant : ➤ Pour créer model nommé « Employe » ➤ Pour créer un Contrôleur de ressource nommé « EmployeeController » ➤ Pour créer une migration pour la table « Employes »	(1pts) (1pts) (1pts)
Q4) Modifiez le fichier de migration pour définir les champs de table « Employes »	(2pts)
Q5) Ecrire la commande pour Exécutez les migrations	(1pts)
Q6) Ajouté une route de ressources pour le contrôleur « EmployeeController » avec URL: « Employe »	(1pts)
Soit la vue avec le nom AjouteEmploye.blade.php dans le dossier /resource/views/ qui permet d'ajoute un employé à la table Employés :	
<div style="border: 1px solid black; padding: 10px;"> <h3 style="text-align: center;">Ajouter un employé</h3> <p>Nom d'employé : <input type="text"/></p> <p>Prénom d'employé : <input type="text"/></p> <p>Adresse : <input type="text"/></p> <p>Téléphone : <input type="text"/></p> <p>Departement_Id : <input type="text"/></p> <p style="text-align: left;"><button style="background-color: #007bff; color: white; padding: 5px 10px;">Valider</button></p> </div>	
(Annexe 1)	
Q7) Modifier la fonction create() (dans le fichier EmployeeController) pour retourne la vue AjouteEmploye.blade.php	(1pts)
On Cliquant sur le bouton Valider :	
Q8) Modifier la fonction store() (dans le fichier EmployeeController) pour Ajoute les donnée de la formulaire à la table Employes	(2pts)
Q9) Modifier Action="" et Method="" dans le formulaire de la page ci-dessus (Annexe 1) pour router la fonction store ()	(2pts)



Soit la vue avec le nom **ListEmployes.blade.php** dans le dossier /resource/views/ qui permet d'afficher la liste des employés :

Liste des employés						
ID	Nom	Prénom	Adresse	Téléphone	Département ID	Actions
1	Nom 1	Prenom 1	Laayoune hay laayoune N°287	02-00-00-00-00	2	Éditer Supprimer
2	Nom 2	Prenom 3	Laayoune hay laayoune N°412	02-11-11-11-11	2	Éditer Supprimer

(Annexe 2)

Q10) Modifier la fonction **index()** (dans le fichier **EmployeController**) pour afficher la liste des employés. Utilisez les fonctionnalités de pagination de Laravel pour afficher 10 employés par page

(1pts)

On Cliquant sur le bouton **Modifier** (annexe 2)

Q11) Modifier la fonction **edit()** (dans le fichier **EmployeController**) pour retourne une vue de modification d'un employé existante avec un formulaire pré-rempli avec les informations de l'employé à modifier.

(4pts)

Q12) Modifier la fonction **Update()** (dans le fichier **EmployeController**) pour modifier les donnée de l'employé à la base de données et retourne la vue **ListEmployes.blade.php**

(2pts)

Q13) Modifier la fonction **destroy()** (dans le fichier **EmployeController**) pour supprimer un employé et retourne la vue **ListEmployes.blade.php**

(2pts)

Q14) Ajoutez une vue de confirmation pour supprimer un employé

(1pts)

Q15) Ecrire les commandes artisan correspondant de Créer « seeder » pour insérer des données de test dans la table **Employes**

(1pts)

Q16) Créez un middleware pour empêcher l'accès aux utilisateurs qui ne sont pas autorisés à modifier les employes.

(3pts)