



مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail
Régionale SUD

Examen de Fin de Module (Régional)

Module : Développement back-end		Variante N° : V2	
Filière : Développement digital	Année : 2023/2024	Durée : 2h30 heures	
Niveau : TS	Mode de formation : Résidentiel	Barème : .../40	

<i>Partie théorique (11 pts)</i>	<i>Note</i>
<u>Exercice 1 : (5pts)</u> Trouver la bonne réponse : 1- Quelle est la fonction utilisée pour valider les données dans les requêtes HTTP dans Laravel ? a. Validate() b. Request() c. Check() 2- Quelle est la méthode utilisée pour définir une relation "One-to-Many" dans Eloquent ? a. belongsToMany() b. hasOne() c. hasMany() 3- Quelle commande permet de créer un nouveau contrôleur dans Laravel ? a. php artisan make:controller b. php artisan new:controller c. php artisan create:controller. 4- Quel est le fichier principal de configuration des routes dans Laravel ? a. Routes.php b. web.php c. config.php 5- Comment récupérer tous les enregistrements d'une table nommée " Utilisateurs" en utilisant Eloquent ? a. Utilisateur::fetch () b. Utilisateur::all() c. Utilisateur::retrieve ()	 (1 pt) (1 pt) (1pt) (1pts) (1pts)
<u>Exercice 2 : (6 pts)</u> Q1) Quelle est la différence entre une route GET et une route POST dans Laravel ? Q2) Comment définir une relation "Many-to-Many" entre deux modèles dans Laravel ? Q3) Qu'est-ce que le routage dans Laravel?	 (2pts) (2pts) (2pts)



مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail
Régionale SUD

Partie pratique : (29 pts)

Note

Soit le modèle relationnel suivant :

Utilisateurs (Id, nom, Email, Mot_De_Passe)

Taches (Id, Titre, Description, date_De_Creation, date_De_Limit, #Utilisateur_id)

Q1) Ecrire la commande pour créer un projet laravel nommé « **Gestion_des_Taches** »

(1pts)

Soit une base de données vide avec le nom **Gestion_des_Taches**:

Q2) Ecrire la configuration des informations de connexion de base de données (dans le fichier .env)

(2pts)

Q3) Ecrire les commandes artisan correspondant :

(1pts)

➤ Pour créer model nommé « **Tache** »

(1pts)

➤ Pour créer un Contrôleur de ressource nommé « **TacheController** »

(1pts)

➤ Pour créer une migration pour la table « **Taches** »

Q4) Modifiez le fichier de migration pour définir les champs de table «**Taches**»

(2pts)

Q5) Ecrire la commande pour Exécutez les migrations

(1pts)

Q6) Ajouté une route de ressources pour le contrôleur « **TacheController** » avec un URL : «**Tache**»

(1pts)

Soit une vue avec le nom **AjouteTache.blade.php** dans le dossier /resource/views/ qui permet d'ajoute une tache à la table **Taches** :

Ajouter une Tache

Task Title:

Description:

Creation Date:

Deadline:

User ID:

(Annexe 1)

Q7) Modifier la fonction **create()** (dans le fichier **TacheController**) pour retourne la vue **AjouteTache.blade.php** (Annexe 1)

(1pts)

On Cliquant sur le bouton **Submit form** :

Q8) Modifier la fonction **store()** (dans le fichier **TacheController**) pour Ajoute les donnée de la formulaire à la table **Taches**

(2pts)



مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail
Régionale SUD

Q9) Modifier **Action**="" et **Method**="" dans le formulaire de la page ci-dessus (Annexe 1) pour router la fonction **store** ()

(2pts)

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

ID	Titre	Description	Date de création	Date limite	Utilisateur ID	Actions
1	Titre de la tâche 1	Description de la tâche 1	2024-04-16	2024-04-30	1	Éditer Supprimer
2	Titre de la tâche 2	Description de la tâche 2	2024-04-16	2024-04-30	2	Éditer Supprimer

(Annexe 2)

Q10) Modifier la fonction **index()** (dans le fichier **TacheController**) pour afficher les données de la table **Taches**. Utilisez les fonctionnalités de pagination de Laravel pour afficher 10 taches par page

(1pts)

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

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

(4pts)

Q12) Modifier la fonction **Update()** (dans le fichier **TacheController**) pour modifier les donnée de la tâche à la base de données et retourne la vue **ListTache.blade.php**

(2pts)

Q13) Modifier la fonction **destroy()** (dans le fichier **TacheController**) pour supprimer une tache et retourne la vue **ListTache.blade.php**

(2pts)

Q14) Ajoutez une vue de confirmation pour supprimer une tache.

(1pts)

Q15) Ecrire les commandes artisan correspondant de Créer une seeder pour insérer des données de test dans la table **Taches**

(1pts)

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

(3pts)