|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***EXAMEN DE FIN DE MODULE  Direction régionale Fès-Meknès*** | | | | | |
| *Etablissement :* | | ***IFMOTICA FES*** | | | ***Durée : 2H*** |
| *Filière :* | *Développement digital option web full stack* | | | *Année : 2022/2023* | *1A 2A* |
| *Module :* | *Développer en back-end* | | | | ***V 1*** |
| *Epreuve/Barème :* | | *Théorique /40* | *Pratique* | | |

***Questions du cours (5 pts)***

1. Quelle commande est utilisée pour démarrer le serveur Laravel ? (1pts)
2. Pourquoi Laravel utilise-t-il le moteur de template Blade ? (1pts)
3. Où se trouve le fichier de routage dans Laravel ? (1pts)
4. Comment définir des données de session dans Laravel ? (1pts)
5. Quelle classe est utilisée dans Laravel pour gérer les exceptions ? (1pts)

***Exercice (35 pts)***

Cette application web qui permettra la gestion des stages dans une formation pour un ensemble des apprentis.

La base de données « db\_apprenti » est la suivante :

***Apprenti*** (**NumApprenti**, nom, prénom, email, date naissance, adresse, ville, photo)

***Artisan*** (**CIN**, nom, prénom, adresse, tel)

***Stage*** (**#NumApprenti,#CIN**, dateDébut, dateFin)

1. Créer une nouvelle application laravel nommé gestion\_apprenti. (1pts)
2. Configurer la base de données dans le fichier .env (1pts)
3. Créer les migrations pour les tables Apprenti et stage. (1pts)
4. Editer la fonction up () pour ajouter les champs nécessaires aux tables Apprenti et stage. (2pts)
5. Créer les modèles nécessaires pour les tables Apprenti et stage en ajoutant les relations entre les tables. (1pts)
6. Donner la commande qui permet d’exécuter les migrations. (1pts)
7. en utilisant la ligne de commande, créer une factorie nommée ***ApprentiFactory***. (1pts)
8. Créer un seeder qui contient la méthode **run (). (**1pts)
9. Ajouter, dans le seeder, le code nécessaire permettant de remplir la table Apprenti par des données de test (50 enregistrements). **(**1pts)
10. Exécuter le Seeder. **(**1pts)
11. Créer un Controller ***ApprentiController***. **(**1pts)
12. Ecrire le code de la méthode index () de contrôleur ***ApprentiController*** qui renvoi la liste des apprentis avec une pagination de 10 lignes. **(**2pts)
13. Ecrire le code de la méthode store (Request $request) de contrôleur ***ApprentiController*** qui :
14. permet d’ajouter un nouvel apprenti. **(**2pts)
15. Ajouter à cette méthode les contraintes de validation des champs : **(**2pts)

* La saisie des champs obligatoires
* email valide
* nom prénom n’accepte que les lettres
* adresse accepte au moins 12 caractères et ne dépasse pas 200 caractères

1. Ecrire le code de la méthode update (Request $request) de contrôleur ***ApprentiController*** qui permet de modifier un apprenti. **(**2pts)
2. Ecrire le code de la méthode destroy (string $id) de contrôleur ***ApprentiController*** qui permet de supprimer un apprenti. **(**2pts)
3. Ajouter une méthode getApprentiByVille (string $ville) de contrôleur ***ApprentiController*** qui permet de renvoyer la liste des apprentis par la ville passer en paramètre. **(**2pts)
4. Créer les routes pour l’application dans le fichier routes/web.php . les routes devraient inclure des routes pour créer, afficher, mettre à jour et supprimer des apprentis. **(**2pts)
5. Pour Créer un stage il nécessite l’existence au moins un apprenti et un artisan dans la base de données. pour cela ajouter un middleware qui permet de vérifier si un apprenti et artisan existe, sinon interdit l’accès à la route « /stage », et rediriger vers les routes « /apprenti » et/ou « /artisan » pour la saisi des apprentis et artisans. **(**3pts)
6. Créer la vue index.blade.php qui affiche la liste des apprentis dans un tableau (ajoute les liens des boutons ajouter, supprimer, modifier). **(**3pts)
7. Pour sécuriser notre application .écrire les commandes nécessaires pour ajouter un système d’authentification de votre choix (Breeze, Jetstream,Fortify,Sanctum …).**(**3pts)