

QCM LARAVEL

Question 1 : composer create-project laravel/laravel nom-du-projet

Question 2 : En utilisant la méthode with() ou en passant un tableau associatif à la méthode view().

Question 3 : Blade.

Question 4 : En utilisant les classes de validation ou en utilisant la méthode validate() dans le contrôleur.

Question 5 : php artisan make:model NomDuModele

Question 6 : En créant une nouvelle migration avec la commande php artisan make:migration, puis en ajoutant la colonne dans la méthode up() de la migration

Question 7 : redirect()->route()

Question 8 : En utilisant la méthode statique all() sur le modèle concerné

Question 9 : php artisan make:middleware NomDuMiddleware

Question 10 : En utilisant la méthode any() lors de la définition de la route

Question 11 : En injectant la dépendance Illuminate\Http\Request dans la méthode du contrôleur ou en utilisant la méthode request() dans le contrôleur

Question 12 : En utilisant les rate limiters de Laravel.

Question 13 : En utilisant la classe Mail de Laravel

Question 14 : php artisan serve

Question 15 : En utilisant la méthode request() dans la route ou en utilisant la classe Request dans le contrôleur

Question 16 : En utilisant la méthode foreign() dans la méthode up() de la migration

Question 17 : En utilisant les tâches planifiées (scheduled tasks) via la méthode schedule() dans le fichier app/Console/Kernel.php

Question 18 : En utilisant les form request et la méthode validated()

Question 19 : En définissant les relations dans les modèles Eloquent à l'aide des méthodes hasMany() et belongsTo()

Question 20 : En utilisant la méthode with() avec des contraintes conditionnelles

Question 21 : En définissant un accesseur dans le modèle Eloquent

Question 22 : Tinker est un REPL (Read-Eval-Print Loop) interactif pour Laravel, utile pour tester du code rapidement et interagir avec l'application

Question 23 : En utilisant la commande php artisan vendor:publish

Question 24 : En utilisant la commande php artisan key:generate

Question 25 : En utilisant la commande `php artisan make:event` pour créer un événement et `php artisan make:listener` pour créer un écouteur associé

Question 26 : En créant une classe qui implémente l'interface `Illuminate\Contracts\Cache\Store`

Question 27 : Le fichier de configuration `config/services.php`

Question 28 : En utilisant les directives `@slot` et `@component` dans les fichiers de vues Blade

Question 29 : Un trait est une collection de méthodes réutilisables dans les modèles ou les contrôleurs. Il peut être utilisé en l'incluant dans une classe via le mot-clé `use`

Question 30 : En définissant une méthode dans le modèle Eloquent avec le préfixe `get` suivi du nom de l'attribut, suivi de `Attribute`. Par exemple, `getNomAttribute()`

Question 31 : Utilisez `DB::transaction()` pour encapsuler les opérations de base de données et garantir leur succès ou leur annulation en cas d'échec

Question 32 : La méthode `encrypt()` permet de crypter les données avant de les sauvegarder dans la base de données.

Question 33 : Créez un fichier `webpack.mix.js` et utilisez les méthodes de `Mix` pour définir les tâches de compilation des assets. Ensuite, exécutez `npm run dev` ou `npm run production`

Question 34 : Utilisez `json()` sur l'objet de réponse pour retourner une réponse JSON à partir d'un contrôleur.

Question 35 : Définissez des événements dans vos modèles avec `boot()` ou `initializeModelEvent()` et associez des écouteurs d'événements dans `app/Listeners`.