LARAVEL

LES COMMANDES

Pour créer un nouveau projet php Laravel, nous avons besoins de commandes :cd

Composer create-project laravel / nom du projet-project--prefer-dist

Apres avoir créé notre projet Laravel, on tape la commande

**Cd** → chemin d’accès

**Php artisant serve**→ pour demarrer le serveur, il permet de faire plusieurs choses :

- pour voir les différentes fonctionnalités qu’on avoir avec php artisant,

- la liste les differentes commandes qui sont associes à php artisant

Une fois php artisant est lance, il aura un message avec une adresse qui va nous permet d’acceder a notre serveur, nous devons taper cette adresse au niveau de notre navigateur

**(http : // 123.0.O.1 :800 /)**

Le **framework** est unensemble de composeur**,** un logiciel qui nouspermet de faciliter le développement, il permet de mettre en place un site dynamique

Structure de base de dossiers :

* Laravel est le dossier principal
* Les modules
* Pages
* Eser (

**App** tout ce qui est exécutable se fait au niveau du dossier app

**Bootstrap** gere la responsibilite c’est-à-dire l’utilisation d’un site ou l’application par rapport a un pip different . ca peut etre un ordinateur , un tephone potable

**Config** permet de gérer la configuration de tous les fichiers qui se trouvent dans le dossier config c’est a tout ce qui est configuration

**Database** est la base de donnee , avec Laravel , la bse de donnee se fait de manière automatique

**Laravel** est un framework phpest sous orme du model views controller (MVC)

**Resource** dans views gere les pages et les modules ( FRONT ET BACK)

tout ce qui est exécutable se fait niveau du dossier App

tout ce qui est configuration se fait au niveau de config

views : gere page et module

route :dossier très important , il le faut pour fonctionner

storage : stocker dossier vituel

.env

1. Créer un modele au niveau de app , ce modele cree de manière automatique une table qui a le meme nom que le modele mais cette table n’ira pas dans la BD directement , ca sera un ficher qui va etre ecrit au niveau de notre dossier mylaravel mais c’est a nous de l’executer apres c’est-à-dire en suivant des etapes :

* Créer le controller du modele apres on crée la table en faisant **php artisan migrate**
* **le public** c’est ce tout le monde voir , il gere que css et js c-a-d les images, les photos etc mais pas un fichier html

public correspond a notre dossier **asset ( js, css ,images ..)**

**resource**  dans ce dossier se trouve **views** qui gere tout ce qui est vus , la partie BACK et FRONT

Etape1 : Php artisan make : model etudiant-m ( fait entrer)

Etape2 : php artisan make :controler EtudiantController- - etudiant- -resource (entrer)

Allez ds App – resouce – controller

**Migration de la table cree**

Php artisan migrate

Après la migration

$table→ id() ;

$Table → timestamps();

$Table →string(‘nom’);

$Table→ string(‘nom’);

$Table→ string (‘email’);

**PARTIE1**

Pour créer une table (model)

c:\xampp\htdocs\Plaravel>php artisan make :model etudiant –m

pour créer un controller

c:\xampp\htdocs\Plaravel>php artisan make:controller EtudiantController --model etudiant

pou migrer une table on rempli les champs avant de la migrer

c:\xampp\htdocs\Plaravel>php artisan migrate

public :page ,module,formulaire

temlog :pour telecharger des templete laravel

View :FRONT ET BACk

Route :web.php pou gerer les routages

Templaite

Copi le doc asset,mettre dans public,

Ressouece -View FRONT- index.blade.php

Idex.html copi met dans index.blade.php

Cop le head

View front heade .blade .php Col le head.html

Das index.blade.php @include(« FRONT/head »)

Cpie header.html

Ainside suite

On ferme tous les balise pour aller plus vite dans le decoupe

Les commantaire on les coupe

Sections dans main .blade.php dans front

Containt.blade .php

@yield

(‘contenu’)(on fait le chiwt les cases) Inclus dans index.blade.php

Head :devant asse ‘’{{asset(‘as.. ‘)}}’’

Pour tous les asset dn le href,head(les cod css,les js ,js

Ctr+f :on ecrit .js’)}}’’ remplacer tou

Pour permettre de ne pas perdre les view dn les asset remlace ../

Routes-web.php- return view(‘FRON .main’) main notre 1 page

Main.blade.php

1ligne( @extends(‘FRON .index’),@section(‘contenu’)

Alafin du code

@endsection(pour fermer)

@if()@endif

Cmd

Ctr +c

Composer require laravel/jetstream(va nous chercher les commposant jet pour gerer installeer la partie back et on change le template par defaut conexion,auten…)(y’ il a des dossier qui son cree,le ficher execution il le cree)

php artisan jetstream:install livewire(qui cree automatiquement la parti back les chemain view())

Php artisan jetstream:install livewire --teams

Node. js gere tous technologies actuelle

composer install(pour installer tous les composers)

npm install

npm run dev

php artisan migrate

php artisan vendor:publish --tag=jetstream-views

npm run dev

si le nod js est bien installer on aura un doc dn notre projet node\_module

istalle node js,

php artisan config:clear

1. vider le cache : php artisan optimise : clear
2. vidage du compositeur-chargement automatique
3. essayer à nouveau

php artisan view:clear

php artisan migrate:reset

**PARTIE2**

Route::get('/logout',[UserController::class,'logout'])->new('logout');

Pour la partie back :les clesse dans le HTL on les change pas ;

Pour la partie BACK :

1)auth dans viw,on diplique les ficher,on copîe les code html correspondant on les cole dans les .blade.php,et on fait les changement par rapport a notre besoin,les LIKE et JS on ajoute apres href{{asset(‘BACK/…’)},aprs chaque gengement on suprime le code qui vient de hath

On remplace : href="./register">Creer un compte

Pour la derniere ligne du code

Pour l’espace de coonnecxion on remplace : <a href="./"><img src="{{asset('BACK/images/logo-full.png')}}" alt=""></a>

Pour les image aussi on m