TD LARAVEL

Afin de vous familiariser avec le framework Laravel , nous allons réaliser en cours une micro application où les utilisateurs pourront poster de simple messages.

PARTIE 1 – INSCRIPTION / LOGIN

Par chance Laravel nous mâche le travail en ce qui concerne la partie Inscription et connexion.

Lancer la commande *php artisan migrate* pour créer la table users déjà préétablie par Laravel (cf. dossier database/migration).

Puis lancer la commande *php artisan make :auth pour créer les templates dans le dossier ressources/views.*

Nous aurons ainsi un système clé d’inscription / connexion clé en main :

**Lien vers la page d’inscription :** localhost/platform/public/index.php/register

**Lien vers la page de connexion :** localhost/platform/public/index.php/login

PARTIE 2 – La classe Post (Model)

Importer la table posts de la base de données  « projet » dans la base de données « platform ».

Dans le dossier app/models créer le fichier Post.php :

1. **<?php**
3. **namespace** App\Models;
5. **use** Illuminate\Database\Eloquent\Model as Eloquent;
7. **class** Post **extends** Eloquent
8. {
10. **protected** $with = ['author'];  // A chaque fois qu'on récupérera un post on joindra les infos de son auteur
12. **public** **function** author()
13. {
15. return $this->belongsTo('App\User');  // Laravel fera une jointure de table pour récupérer les infos de l'auteur associé à la clé étrangère user\_id
17. }

}

PARTIE 4 – Les templates (Vue)

1. Le template global « layout.blade.php »

Il s’agit du template parent dont tous les autres templates de notre projet hériteront.

Créer le fichier layout.blade.php à la racine de dossier ressourses/views et insérez y ce code :

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Plateforme d’annonce</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.6/css/bootstrap.min.css">

<body>

<nav><a href="{{ url('/login') }}">Login</a> <a href="{{ url('/register') }}">Inscription</a></nav>

<div class="raw">   
 <div class="col-md-4">   
 **@yield('sidebar')**

</div>

<div class="col-md-8">   
 @if (count($errors) > 0)

<div class="alert alert-danger">

<ul>

@foreach ($errors->all() as $error)

<li>{{ $error }}</li>

@endforeach

</ul>

</div>

@endif  
 **@yield('content')**

</div>

</div>

</body>

</html>

1. Le template enfant « create.blade.php»

Créer un dossier post dans le dossier ressources/views puis ajoutez-y le fichier create.blade.php avec ce code :

*@extends('layout')*

*@section('sidebar')*

*Contenu divers et varié…*

*@endsection*

*@section('content')*

*{!! Form::open(['url' => 'project/create']) !!}*

*{{ csrf\_field() }}*

*{!! Form::textarea('content') !!}*

*{!! Form::submit('Envoyer !') !!}*

*{!! Form::close() !!}*

*@endsection*

< !—fin fichier create.blade.php -->

Pour en savoir plus sur la classe Form : <https://laravelcollective.com/docs/5.2/html>

PARTIE 4 – PostController

1. Créer dans le dossier app/Http/Controllers le fichier PostControllers.php :

<?php namespace App\Http\Controllers ;

use Illuminate\Routing\Controller;

class PostController extends Controller {

}

1. Charger la classe Post

<?php namespace App\Http\Controllers ;

use Illuminate\Routing\Controller;  
use App\Models\Post;

class PostController extends Controller {

}

1. Charger la classe Request qui nous permettra de récupérer les données de formulaire :

<?php namespace App\Http\Controllers ;

use Illuminate\Routing\Controller;  
 use App\Models\Post;  
 use Illuminate\Http\Request;

class PostController extends Controller {

}

En savoir plus sur l’objet Request : <https://laravel.com/docs/5.2/requests>

1. Charger la classe Auth qui nous permettra de récupérer les informations de l’utilisateur connecté, et donc de l’auteur du post :

<?php namespace App\Http\Controllers ;

use Illuminate\Routing\Controller;  
 use App\Models\Post;  
 use Illuminate\Http\Request;  
 use Auth ;

class PostController extends Controller {

}

1. Ajouter la méthode create qui se chargera uniquement d’afficher le formulaire de création de post:

<?php namespace App\Http\Controllers ;

use Illuminate\Routing\Controller;  
use Illuminate\Http\Request;  
use App\Models\Post;   
use Auth;

class PostController extends Controller {  
  
 protected function validator(array $data) {  
 //  
 }

public function create()

{

return view('post/create') ; // on charge le template create.blade.php stocké dans le dossier ressources/views/post/

}

}

6) Ajouter la méthode store qui se chargera uniquement d’enregistrer le post envoyé en base de données :

<?php namespace App\Http\Controllers ;

use Illuminate\Routing\Controller;  
 use Illuminate\Http\Request;  
 use App\Models\Post;  
 use Auth ;

class PostController extends Controller {

public function create()

{

return view('post/create') ;

}

public function store(Request $request)

{  
 $post = new Post ;  
 $post->content = $request->input('content') ;  
 $post->user\_id = Auth ::Id() ;  
 $post->save() ;

$request->session()->flash('success', 'Message publié avec succès ! ') ; // On ajoute un message flash qui sera supprimé une fois affiché.  
 redirect()->to('/post/create') ;

}

}