|  |  |
| --- | --- |
|  | Année Universitaire : 2023-2024  **Applications Web Avancées** |
| **Atelier 4 : Authentification-Jetstream** | |

**Objectifs**

Le but de cet atelier est d’installer, configurer, personnaliser le package Jetstream pour créer des fonctionnalités d'authentification dans Laravel.

**Prérequis logiciel**

* Projet Laravel créé.
* Création de la base de données
* Gestion du CRUD d’un modèle
  1. **Installation par défaut**

1. Créer un nouveau projet Laravel.
2. Créer la base de données « authentification »
3. Installer jetstream : **composer require laravel/jetstream**
4. Installer Livewire : **php artisan jetstream:install livewire**
5. Installer les dépendances : **npm install**
6. Lancer la commande**: npm run dev**
7. Lister les routes
8. Exécuter les migrations
9. Afficher la page d’accueil.
10. Créer un nouvel utilisateur. Puis connectez-vous et consulter la partie profile et dashboard.
    1. **Personnalisation de l’inscription :**

On va maintenant ajouter deux champs dans le formulaire d’inscription : username et phone.

1. Mettre à jour le fichier de migration de la table User

**public function** up()  
{…

$table->string(**'username'**)->**unique**();  
$table->string(**'phone'**)->**unique**();

..}

1. Mettre à jour la base de données
2. Mettre à jour le modèle User

**protected $fillable** = [  
 **'name'**,  
 **'username'**,  
 **'phone'**,  
 **'email'**,  
 **'password'**,  
];

1. Mettre à jour la vue **ressources/auth/register.blade.php**

<**div**>  
 <**x-label for="username" value="**{{ \_\_(**'Username'**) }}**"** />  
 <**x-input id="username" class="block mt-1 w-full" type="text" name="username" :value="old('username')" required** />  
</**div**>  
<**div class="mt-4"**>  
 <**x-label for="phone" value="**{{ \_\_(**'Phone Number'**) }}**"** />  
 <**x-input id="phone" class="block mt-1 w-full" type="text" name="phone" :value="old('phone')" required** />  
</**div**>

1. Mettre à jour le fichier **App/Fortify/CreateNewUser.php**

**public function** create(**array** $input)  
{  
 ….

**'username'** => [**'required'**, **'string'**,**'max:255'**, **'unique:users'**],  
 **'phone'** => [**'required'**, **'string'**,**'max:255'**, **'unique:users'**],…

}

**return** User::*create*([  
**…**  
 **'username'** => $input[**'username'**],  
 **'phone'** => $input[**'phone'**],  
 **…**]);

1. Mettre à jour la vue de connexion **ressources/auth/login.blade.php**

<**x-label for="auth" value="**{{ \_\_(**'Email/Username/Phone Number'**) }}**"** />  
 <**x-input id="auth" class="block mt-1 w-full" type="text" name="auth" :value="old('auth')" required autofocus** />  
</**div**>

1. Mettre à jour le fichier de configuration **config/fortify.php**

**'username'** => **'email'**,  
  
**'email'** => **'email'**,

Remplacer par ceci

**'username'** => **'auth'**,

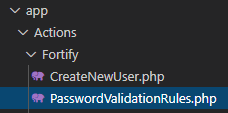
Mettre à jour lefichier **JetstreamProvider**

**use** App\Models\User;  
**use** Illuminate\Http\Request;  
**use** Illuminate\Support\Facades\Hash;  
**use** Laravel\Fortify\Fortify;

**public function** boot()  
{  
 …  
 Fortify::*authenticateUsing*(**function** (Request $request) {  
 $user = User::*where*(**'email'**, $request->**auth**)  
 ->orWhere(**'username'**, $request->**auth**)  
 ->orWhere(**'phone'**, $request->**auth**)  
 ->first();  
  
 **if** ($user &&  
 Hash::*check*($request->**password**, $user->**password**)) {  
 **return** $user;  
 }  
 });  
}

* 1. **Personnalisation des règles de mot de passe :**

Par défaut le mot de passe doit contenir juste au minimum 8 caractères. Ouvrez ce fichier d’action :

[](https://laravel.sillo.org/wp-content/uploads/2020/09/Capture-111.png)

1. Modifier les règles de validation du mot de passe, plus précisément la fonction passwordRules, pour que le mot de passe contient au minimum un Majuscule, un Minuscule, un chiffre et un caractère spécial :

**protected function** passwordRules()  
{  
 **return** [**'required'**,  
 **'string'**,  
 (**new** Password)->length(8),  
 (**new** Password)->requireNumeric(),  
 (**new** Password)->requireSpecialCharacter(),  
 (**new** Password)->requireUppercase(),   
 **'confirmed'**];

}

1. Créer maintenant un nouvel utilisateur pour tester les nouveaux paramètres de connexion.
   1. **Vérification d’email**

Par défaut Jetstream n’impose pas la vérification de l’email.

1. Configurer les détails SMTP dans le fichier .env, de la façon suivante :

MAIL\_DRIVER=smtp  
MAIL\_HOST=smtp.gmail.com  
MAIL\_PORT=587  
MAIL\_USERNAME="adressegmail"  
MAIL\_PASSWORD="motdepassed’application"  
MAIL\_ENCRYPTION=tls  
MAIL\_FROM\_ADDRESS="adressegmail"  
MAIL\_FROM\_NAME="${APP\_NAME}"

1. Activer la vérification d’email au sein du fichier **config/fortify.php**, plus précisément le tableau 'features'.

**'features'** => [  
 Features::*registration*(),  
 Features::*resetPasswords*(),  
 Features::*emailVerification*(),  
 Features::*updateProfileInformation*(),  
 Features::*updatePasswords*(),  
 Features::*twoFactorAuthentication*([  
 **'confirmPassword'** => **true**,  
 ]),  
],

1. Ajoutez que votre modèle **App\Models\User** implémente le contrat **Illuminate\Contracts\Auth\MustVerifyEmail**.
2. Ré-lister les routes.

Après avoir activé cette fonction, vous verrez toutes les routes de vérification des e-mails en action. Les itinéraires de vérification des e-mails sont de trois types :

* Route pour l’envoie un e-mail à l'utilisateur pour vérification.
* Route qui change le statut lorsque l'utilisateur clique sur l'email de vérification.
* Route qui renvoie l'e-mail de vérification.

1. Créer un nouvel utilisateur
   1. **Création d’une nouvelle rubrique au sein du Dashboard**

Nous allons créer une nouvelle section dans le tableau de bord pour gérer les éléments à faire.

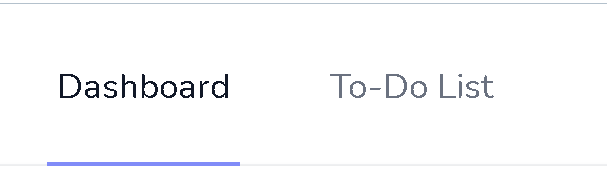
1. Copiez le contenu du fichier **dashboard.blade.php** dans une nouvelle vue **todo.blade.php**
2. Mettre à jour le fichier **resources/views/navigation-menu.blade.php** pour y inclure votre nouvelle section :

*…<!-- Navigation Links -->*……  
  
<**div class="hidden space-x-8 sm:-my-px sm:ml-10 sm:flex"**>  
 <**x-nav-link href="**{{ route(**'dashboard-todo'**) }}**" :active="request()->routeIs('dashboard-todo')"**>  
 {{ \_\_(**'To-Do List'**) }}  
 </**x-nav-link**>  
  
</**div**>

1. ajoutez une nouvelle route protégée dans **routes/web.php**.

Route::*middleware*([**'auth:sanctum'**, **'verified'**])->**get**(**'/dashboard/todo'**, **function** () {  
 **return** view(**'todo'**);  
})->name(**'dashboard-todo'**);

1. Vérifier que vous avez obtenu un nouveau lien dans le menu de navigation. Une page blanche, puisque nous n'avons pas encore personnalisé nos composants



1. Personnaliser l’affichage de la vue «todo.blade.php »

<**x-app-layout**>  
 <**x-slot name="header"**>  
 <**h2 class="font-semibold text-xl text-gray-800 leading-tight"**>  
 {{ \_\_(**'My To-Do List'**) }}  
 </**h2**>  
 </**x-slot**>  
  
 <**div**>  
 <**div class="max-w-7xl mx-auto py-10 sm:px-6 lg:px-8"**>  
 <**div class="bg-red-100 overflow-hidden shadow-xl sm:rounded-lg"**>  
 create a new ToDo  
 </**div**>  
 </**div**>  
 </**div**>  
  
 <**div class="py-12"**>  
 <**div class="max-w-7xl mx-auto sm:px-6 lg:px-8"**>  
 <**div class="bg-red-100 overflow-hidden shadow-xl sm:rounded-lg"**>  
 Affichage des ToDo  
 </**div**>  
 </**div**>  
 </**div**>  
</**x-app-layout**>

