

# NOTE DE STAGE 2021

## CHEZ COFLINE

### SOMMAIRE

Lundi 31 mai 2021 .....	2
Missions : .....	2
Réalisations : .....	2
Mardi 01 mai 2021 .....	5
Missions : .....	5
Réalisations : .....	6
Mercredi 02 mai 2021 .....	6
Missions : .....	6
Réalisations : .....	6
Jeudi 03 mai 2021 .....	10
Missions : .....	10
Réalisations : .....	10
Vendredi 04 mai 2021 .....	11
Missions : .....	11
Réalisations : .....	11

**\*Par soucis de confidentialité ce rapport ne contiendra pas de données sensibles.**

**\*Pré-stage notre maitre de stage nous a conseillé de regarder les vidéos de Nord Coders (YouTube) sur Laravel 8 afin d'avoir une première vision sur comment fonctionne ce Framework.**

LUNDI 31 MAI 2021

## MISSIONS :

1. Implémenter des tests unitaires en rapport avec notre travail précédent afin de vérifier notre travail en particulier les relations *views-controllers-routes* et le passage de données. Travail de manière totalement autonome pour aujourd'hui (représentatif d'un cas réaliste dans le monde du travail).

## REALISATIONS :

1.
  - a. Sources

- Documentation Laravel <sup>123</sup>
- [Laravel.sillo.org](https://laravel.sillo.org) – Les test <sup>4</sup>
- [Kaherecode.com](https://kaherecode.com) – Comment démarrer le TDD (Test Driven Development) <sup>5</sup>
- [Code.tutsplus.com](https://code.tutsplus.com) – Testing Laravel Controllers <sup>6</sup>

Ce qui en ressort :

- ✓ Utilisation de la commande : `php artisan make:test UserTest (--unit)` . Il existe deux types de tests unitaires sous **Laravel 8** les **feature** et les **unit**, Les plus utilisés sont les **feature** qui sont plus versatiles.
- ✓ Utilisation `php config:cache`, `php route:clear` avant chaque test et `php artisan test` pour lancer les tests
- ✓ **Mocking Objects** pour vérifier ce que le **controller** reçoit par une fonction et retourne dans une variable.
- ✓ Les fonctions d'assertions<sup>7</sup> :
  - `assertSuccessful()`,
  - `assertOk()`,
  - `assertRedirect()`,
  - `assertDatabaseHas('table', $data)`,
  - `assertDatabaseMissing('table', $data)`,
  - `assertViewHas()`,
  - `assertViewIs() ...`

---

<sup>1</sup> <https://laravel.com/docs/8.x/testing>

<sup>2</sup> <https://laravel.com/docs/8.x/http-tests>

<sup>3</sup> <https://laravel.com/docs/8.x/mocking>

<sup>4</sup> <https://laravel.sillo.org/cours-laravel-8-les-tests/>

<sup>5</sup> <https://www.kaherecode.com/tutorial/comment-demarrer-le-test-driven-development-avec-laravel>

<sup>6</sup> <https://code.tutsplus.com/tutorials/testing-laravel-controllers--net-31456>

<sup>7</sup> <https://laravel.com/docs/8.x/http-tests#available-assertions>

## b. Essai d'applications des informations regroupées

A ce moment-là, seuls les tests de *index* et *create* semblent fonctionner.

**\*Le code affiché est un peu plus avancé que celui à ce moment-là.**

```
ests > Feature > AddressTest.php > AddressTest > testEdit
1  <?php
2
3  namespace Tests\Feature;
4
5  use Illuminate\Foundation\Testing\RefreshDatabase;
6  use Illuminate\Foundation\Testing\WithFaker;
7  use Tests\TestCase;
8
9  class AddressTest extends TestCase
10 {
11     //use RefreshDatabase;
12
13     private $address;
14
15     protected function setUp(): void
16     {
17         parent::setUp(); // TODO: Change the autogenerated stub
18     }
19
20     public function testIndex()
21     {
22         // Création Mock
23         $this->mock(Address::class,function ($mock)
24         {
25             $mock->shouldReceive('all')->andReturn('addresses');
26         });
27
28         //Action
29         $response = $this->get(route('addresses.index'));
30
31         //Assertions
32         $response->assertOk();
33         $response->assertViewHas('addresses');
34     }
35
36     public function testCreate()
37     {
38         //Action
39         $response = $this->get(route('addresses.create'));
40
41         //Assertions
42         $response->assertOk();
43         $response->assertViewIs('addresses.create');
44     }
45
46     public function testStore()
47     {
48         $data = [
49             'address' => 'test adresse 1',
50             'address2' => 'test adresse 2',
51             'city' => 'Toulon',
52             'zip_code' => '83000'
53         ];
54
55         //Action
56         $response = $this->postJson(route('addresses.store'), $data);
57         dd($response);
58         //Assertions
59         //$response->assertRedirect(route('addresses.store'));
60         //$this->assertDatabaseHas('addresses',$data);
61     }
62 }
```

```

5 public function testEdit()
6 {
7     // // Création Mock
8     // $this->mock(Address::class,function ($mock)
9     // {
10     //     $mock->shouldReceive('findOrFail')->andReturn('address');
11     // });
12     $data = [
13         'id' => $this->address->id,
14         'address' => $this->address->address,
15         'address2' => $this->address->address2,
16         'city' => $this->address->city,
17         'zip_code' => $this->address->zip_code
18     ];
19
20     //Action
21     $response = $this->get(route('addresses.edit'),$address->id);
22
23     //Assertions
24     $response->assertSuccessful();
25     $response->assertViewHas('address','id');
26 }
27
28 public function testUpdate()
29 {
30     $response = $this->get('update');
31     $response->assertStatus(200);
32 }
33
34 public function testDestroy()
35 {
36     $data = [
37         'id' => $this->address->id,
38         'address' => $this->address->address,
39         'address2' => $this->address->address2,
40         'city' => $this->address->city,
41         'zip_code' => $this->address->zip_code
42     ];
43     $response = $this->postJson(route('addresses.delete',['id' => $this->address->id]));
44     $this->assertDatabaseMissing('address',$data);
45 }
46
47 public function testShow()
48 {
49     // // Création Mock
50     // $this->mock(Address::class,function ($mock)
51     // {
52     //     $mock->shouldReceive('findOrFail')->andReturn('address');
53     // });
54
55     $data = [
56         'id' => $this->address->id,
57         'address' => $this->address->address,
58         'address2' => $this->address->address2,
59         'city' => $this->address->city,
60         'zip_code' => $this->address->zip_code
61     ];
62
63     //Action
64     $response = $this->get(route('addresses.show', ['id' => $this->address->id]));
65
66     //Assertions
67     $response->assertSuccessful();
68     $response->assertViewHas('address');
69     $response->assertViewIs('addresses.show');
70 }
71 }

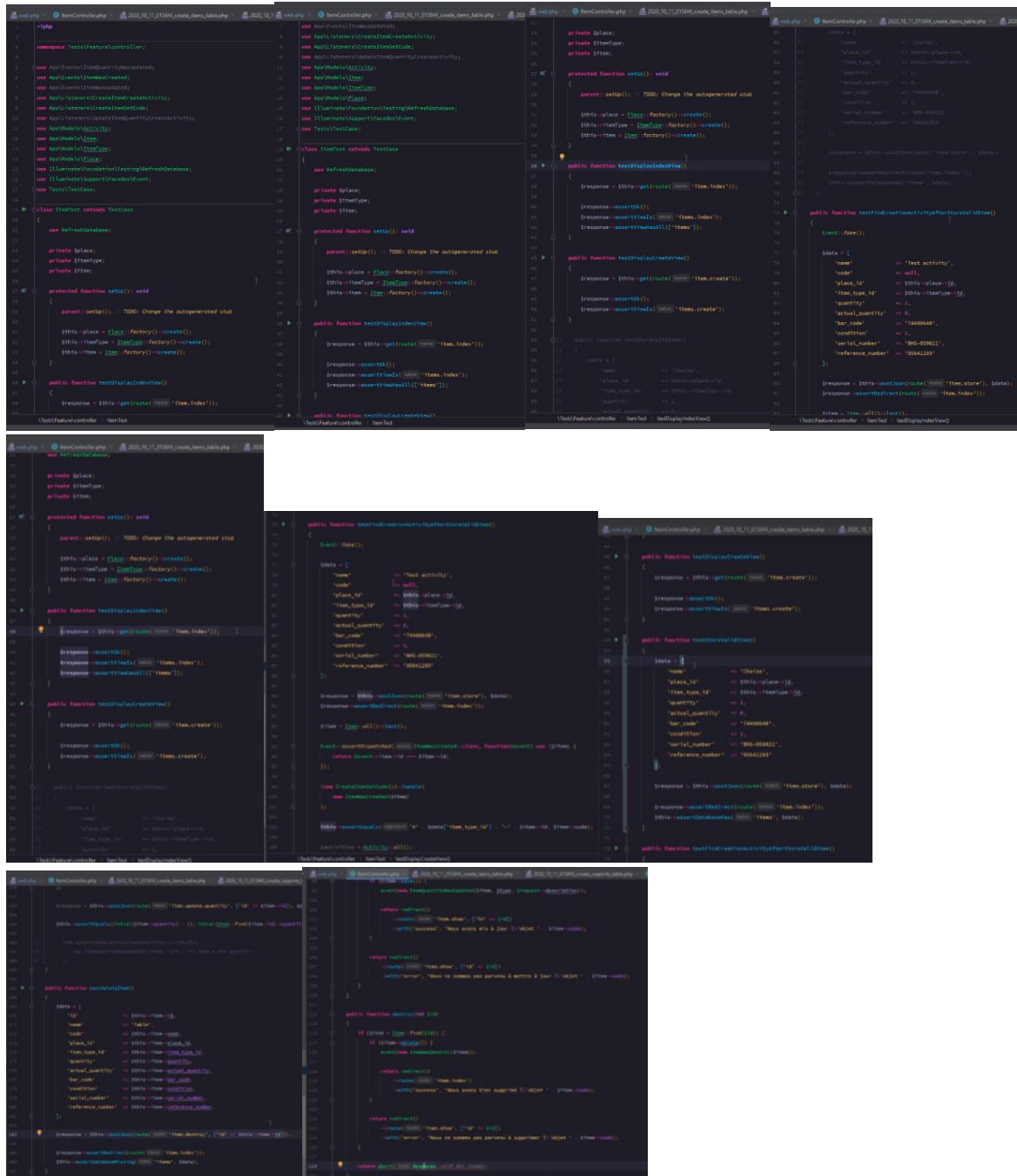
```

MARDI 01 MAI 2021

## MISSIONS :

1. Après diffusion d'un exemple concret de tests unitaires sur un autre projet, continuer l'implémentation des tests.

**\*Screen de l'exemple. Pas ma production.**



## REALISATIONS :

### 1. Erreurs rencontrées et solutions admises.

- Une des erreurs rencontrées était le **CSRF Token Mismatch**, dû aux propriétés des *Token* d'authentification de Laravel. Mon maitre de stage, n'ayant pas rencontré cette erreur, m'a aiguillée vers un lien GitHub où il y avait une solution<sup>8</sup>. De même, j'ai cherché d'autre source afin de m'aider dans la résolution de ce bug<sup>9</sup><sup>10</sup><sup>11</sup>. Mais malheureusement, même après avoir installé *Nuxt*, *NuxtJS*, *Axios*, lancé *Nginx* sur *Laragon* et le mettre à jour ... il était impossible de trouver le fichier *nuxt.config.js* dans l'entièreté du projet. J'ai dû abandonner cette piste.

Après mon partage d'expérience avec les autres stagiaires il paraissait plus évident d'utiliser le « *use WithoutMiddleware* » en début de classe, qui permettait de contourner cette erreur.

- **ErrorException : Trying to get property 'id' of non-object**, cette erreur était due au fait que je n'utilisais pas les *factories*. Notre maitre de stage ne nous avait pas demandé de les implémenter mais cela semblait obligatoire pour réussir à faire fonctionner les tests.

Ce sont les erreurs principales que j'ai rencontrées, il y en a eu d'autres mineures provenant des assertions (code de retour 500 etc...) et je n'ai pas eu le temps d'effectuer tous les correctifs sur mon code ce jour-là.

## MERCREDI 02 MAI 2021

## MISSIONS :

### 1. Finir l'implémentation des tests unitaires.

## REALISATIONS :

### 1.

Après avoir rencontré autant d'erreurs la veille et trouvé les solutions j'ai pu tout mettre à exécution et finir l'implémentation des *factories* ainsi que des tests unitaires.

<sup>8</sup> <https://github.com/nuxt-community/auth-module/issues/744>

<sup>9</sup> <https://dev.to/mrnaif2018/how-to-make-nuxt-auth-working-with-jwt-a-definitive-guide-9he>

<sup>10</sup> <https://www.youtube.com/watch?v=ZegpNcnH3go>

<sup>11</sup> <https://fr.nuxtjs.org/docs/2.x/features/nuxt-components/>

## a. Factories

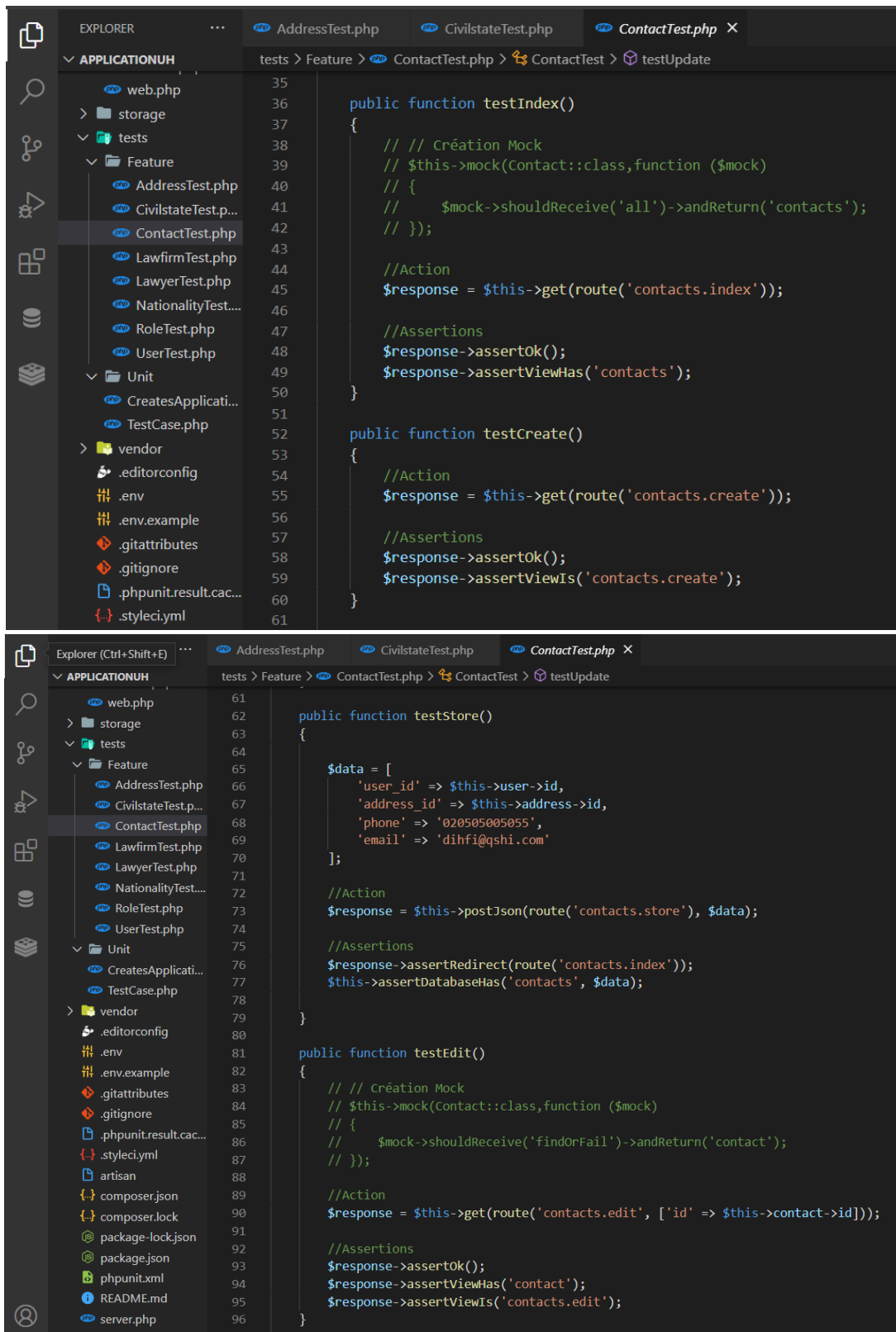
```

1  <?php
2
3  /** @var \Illuminate\Database\Eloquent\Factory $factory */
4
5  use App\Models\Contact;
6  use App\Models\User;
7  use App\Models\Address;
8  use Faker\Generator as Faker;
9
10 $factory->define(Contact::class, function (Faker $faker) {
11     return [
12         'user_id' => function() {
13             return factory(User::class)->create()->id;
14         },
15         'address_id' => function() {
16             return factory(Address::class)->create()->id;
17         },
18         'phone' => $faker->phoneNumber,
19         'email' => $faker->email
20     ];
21 });
  
```

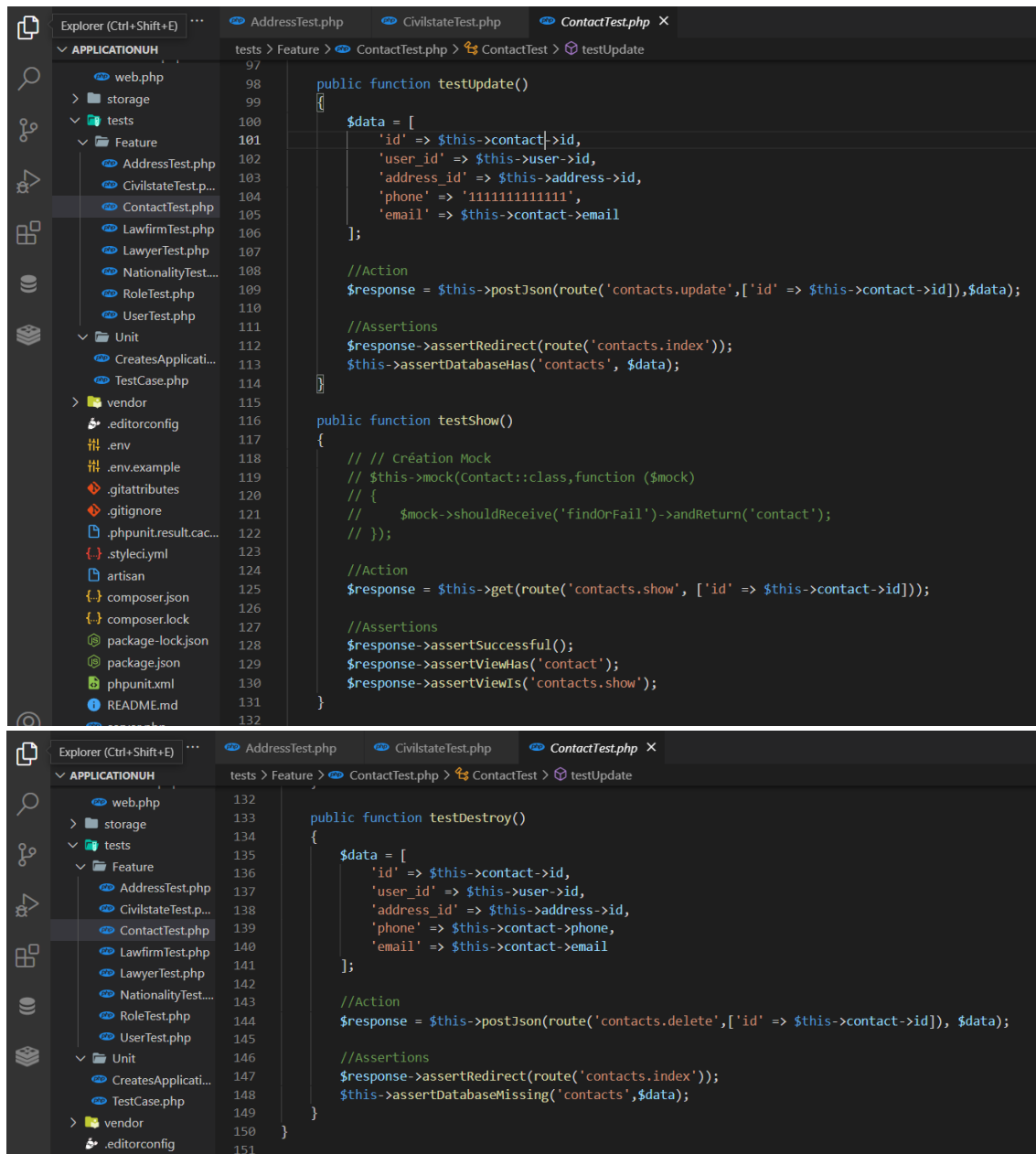
## b. Tests Unitaires

```

1  <?php
2
3  namespace Tests\Feature;
4
5  use Illuminate\Foundation\Testing\RefreshDatabase;
6  use Illuminate\Foundation\Testing\WithFaker;
7  use Illuminate\Foundation\Testing\WithoutMiddleware;
8  //use Database\Factories\ContactFactory;
9  use App\Models\Contact;
10 use App\Models\User;
11 use App\Models\Address;
12 use Illuminate\Http\Request;
13 use Illuminate\Foundation\Testing\DatabaseMigrations;
14 use Illuminate\Foundation\Testing\DatabaseTransactions;
15 use Tests\TestCase;
16
17 class ContactTest extends TestCase
18 {
19     use RefreshDatabase;
20     use WithoutMiddleware;
21
22     private $user;
23     private $address;
24     private $contact;
25
26     protected function setUp(): void
27     {
28         parent::setUp(); // TODO: Change the autogenerated stub
29
30         $this->user = factory(User::class)->create();
31         $this->address = factory(Address::class)->create();
32         $this->contact = factory(Contact::class)->create();
33     }
34 }
  
```







JEUDI 03 MAI 2021

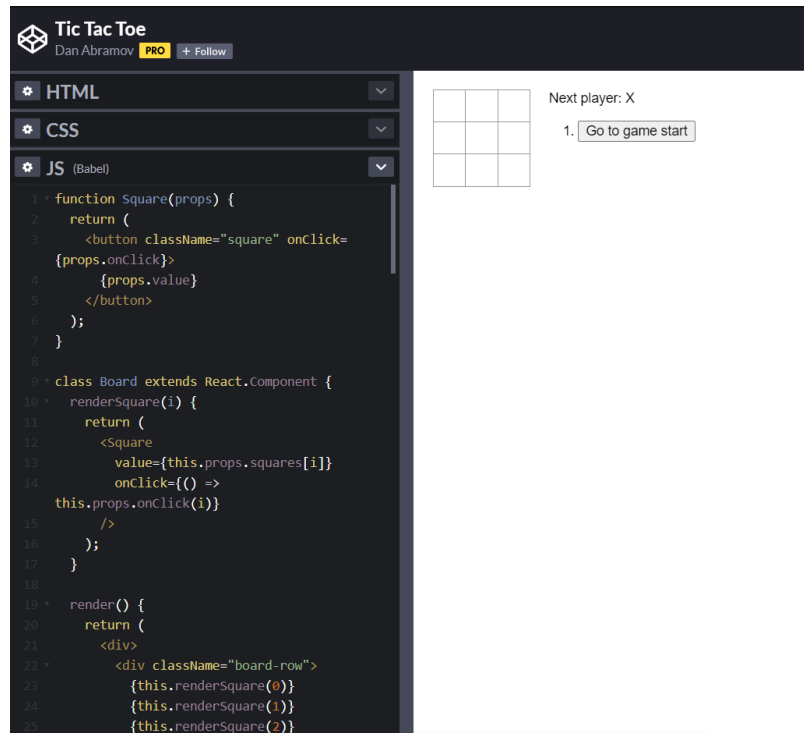
## MISSIONS :

1. Introduction à **ReactJS**. Initiation par différent biais en vue de prochaines missions à venir.

## REALISATIONS :

## 1. Sources :

- Tutoriel d'introduction sur le site de **ReactJs**<sup>12</sup>.  
*« React est une bibliothèque JavaScript déclarative, efficace et flexible pour construire des interfaces utilisateurs (UI). Elle vous permet de composer des UI complexes à partir de petits morceaux de code isolés appelés 'composants' ».*



- Documentation **ReactJs**<sup>13</sup>.
- Rappel sur le **JavaScript**<sup>14</sup>.
- Exemple d'application avec les vidéos de **Nord Coders**<sup>15</sup>.
- **React Developer Tools**, un add-on de navigateur pour avoir accès à une meilleur visibilité lors de l'inspection du code.
- Installation de **NodeJS** pour avoir accès au fonctionnalités **ReactJS**<sup>16</sup>.
- D'après les vidéos de **Nord Coders**, il nous faudra utiliser **TailwindCSS**<sup>17</sup> avec Laravel.

<sup>12</sup> <https://fr.reactjs.org/tutorial/tutorial.html#setup-for-the-tutorial>

<sup>13</sup> <https://fr.reactjs.org/docs/hello-world.html>

<sup>14</sup> [https://developer.mozilla.org/fr/docs/Web/JavaScript/A\\_re-introduction\\_to\\_JavaScript](https://developer.mozilla.org/fr/docs/Web/JavaScript/A_re-introduction_to_JavaScript)

<sup>15</sup> [https://www.youtube.com/watch?v=zNEck1k3\\_zY&list=PLeeuvNW2FHVjVHC8LTbqAvGe9I23sl0Bj](https://www.youtube.com/watch?v=zNEck1k3_zY&list=PLeeuvNW2FHVjVHC8LTbqAvGe9I23sl0Bj)

<sup>16</sup> <https://nodejs.org/en/>

<sup>17</sup> <https://tailwindcss.com/>

VENDREDI 04 MAI 2021

MISSIONS :

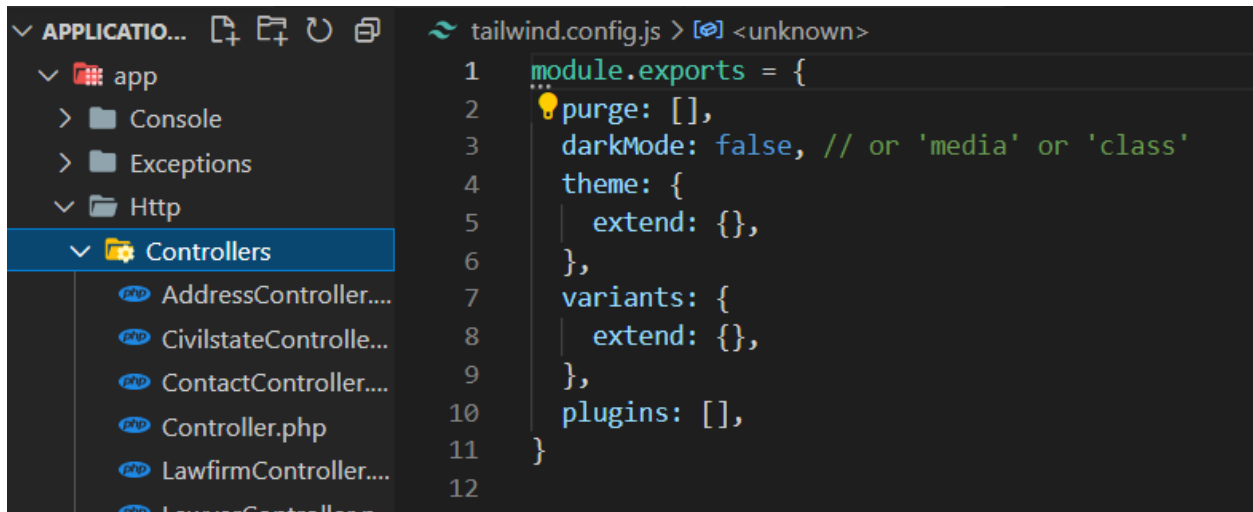
REALISATIONS :

1. Continuer l'initiation et si possible essayer de l'appliquer sur le projet.

1 `npm install`

`npm install -D tailwindcss@latest postcss@latest autoprefixer@latest`

`npx tailwindcss init`



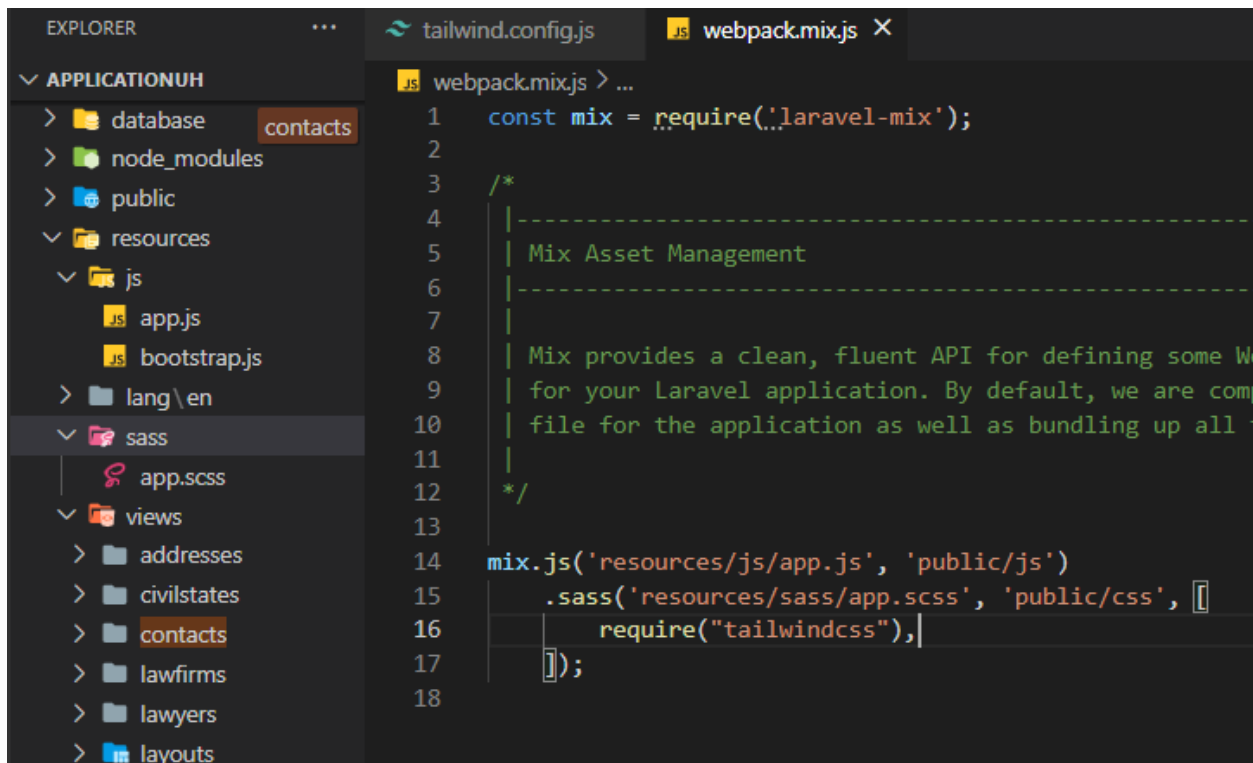
<sup>18</sup> <https://tailwindcss.com/docs/guides/laravel>

- ✓ Configuration du 'purge' pour l'optimisation.

```

1  module.exports = {
2    purge: [
3      './ressources/views/addresses/create.blade.php',
4      './ressources/views/addresses/edit.blade.php',
5      './ressources/views/addresses/index.blade.php',
6      './ressources/views/addresses/show.blade.php',
7
8      './ressources/views/civilstate/create.blade.php',
9      './ressources/views/civilstate/edit.blade.php',
10     './ressources/views/civilstate/index.blade.php',
11     './ressources/views/civilstate/show.blade.php',
12
13     './ressources/views/contacts/create.blade.php',
14     './ressources/views/contacts/edit.blade.php',
15     './ressources/views/contacts/index.blade.php',
16     './ressources/views/contacts/show.blade.php',
17
18     './ressources/views/lawfirms/create.blade.php',
19     './ressources/views/lawfirms/edit.blade.php',
20     './ressources/views/lawfirms/index.blade.php',
21     './ressources/views/lawfirms/show.blade.php',
22
23     './ressources/views/lawyers/create.blade.php',
24     './ressources/views/lawyers/edit.blade.php',
25     './ressources/views/lawyers/index.blade.php',
26     './ressources/views/lawyers/show.blade.php',
27
28     './ressources/views/nationalities/create.blade.php',
29     './ressources/views/nationalities/edit.blade.php',
30     './ressources/views/nationalities/index.blade.php',
31     './ressources/views/nationalities/show.blade.php',
32
33     './ressources/views/roles/create.blade.php',
34     './ressources/views/roles/edit.blade.php',
35     './ressources/views/roles/index.blade.php',
36     './ressources/views/roles/show.blade.php',
37
38     './ressources/views/users/create.blade.php',
39     './ressources/views/users/edit.blade.php',
40     './ressources/views/users/index.blade.php',
41     './ressources/views/users/show.blade.php',
42
43     './ressources/js/app.js'
44   ],

```

✓ Configuration de **Tailwind** avec **Laravel Mix**

```
1 const mix = require('laravel-mix');
2
3
4
5
6
7
8
9
10
11
12
13
14 mix.js('resources/js/app.js', 'public/js')
15   .sass('resources/sass/app.scss', 'public/css', [
16     require('tailwindcss'),
17
18
```

C.

## En Cours...