

# Note de stage 2022

## ⌘UH Group⌘

### Semaine n° 3

## ⌘Objectifs

---

**Contexte** : Après avoir vu avec mon maître de stage pour le système de parrainage, il m’a demandé de présenter un schéma de base de données afin que je puisse illustrer et expliquer mes choix pour ce système. Une fois cela fait il m’a demandé de la faire évoluer et de rester sur cette solution. En plus de ce système de parrainage l’application UH Lawyers propose un système de paiement pour ses clients en passant par l’API Stripe.

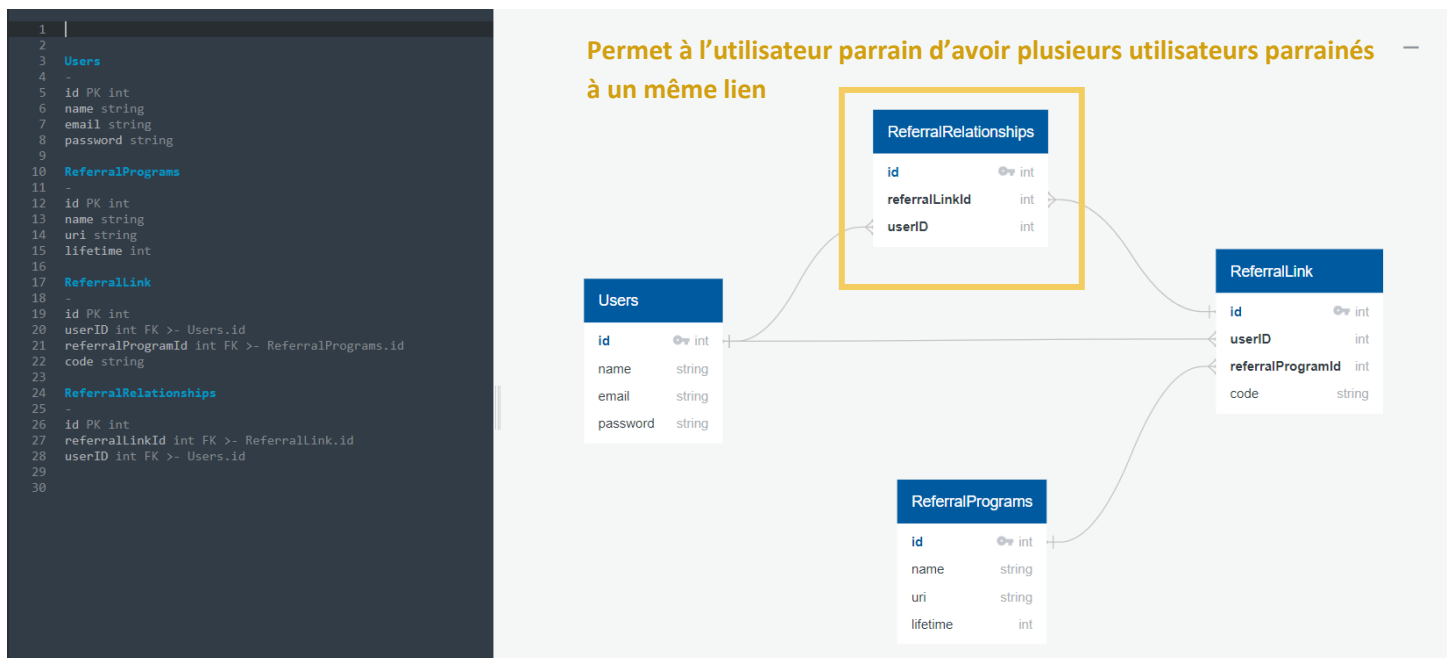
1. Réaliser le schéma de base de données du système de parrainage.
2. Rendre ce système fonctionnel.
3. Se renseigner sur l’API Stripe et réaliser un système de paiement.

⌘Ensemble des liens consultés⌘

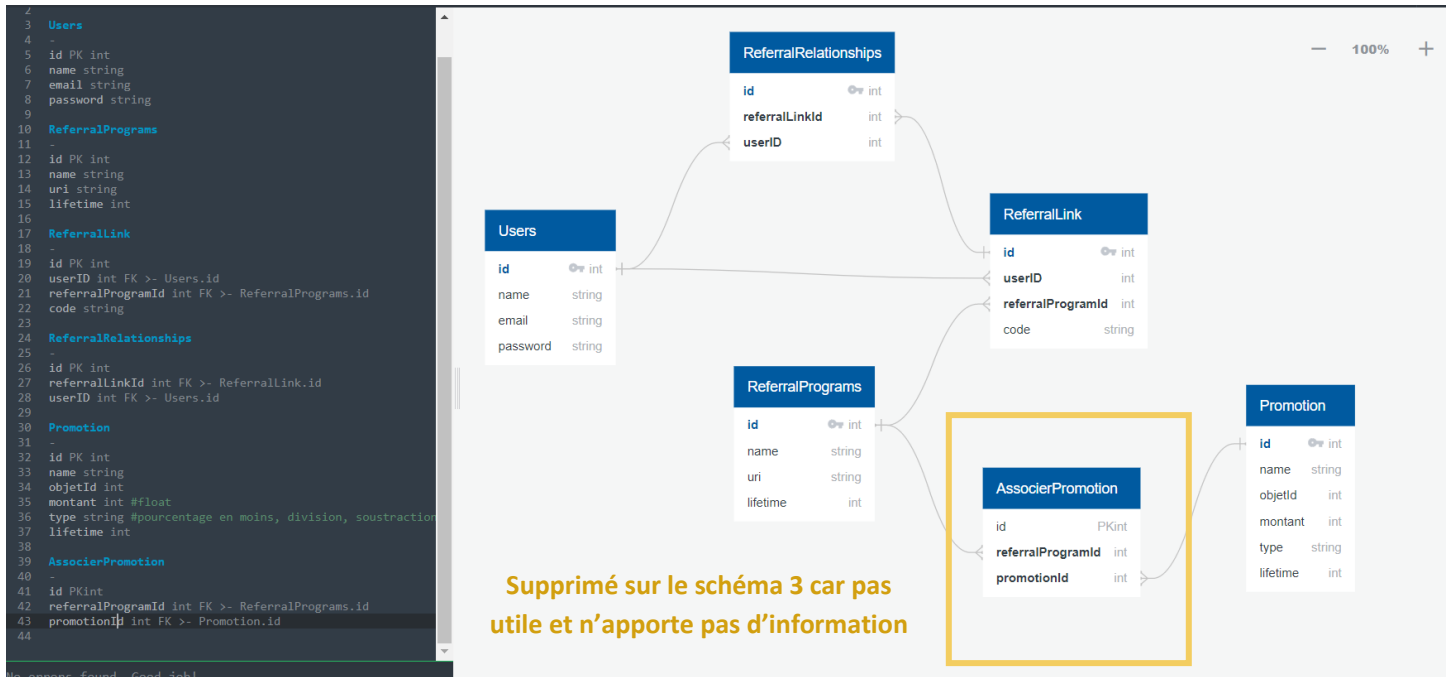
# Réalisations

Pour le premier objectif, j'ai réalisé les différentes étapes du schéma de la base de données du système de parrainage avec le site web [QuickDBD](#). Le premier schéma représente uniquement le système de parrainage, ce système est tiré du [tutoriel](#) que j'ai choisi de suivre car j'appréciais énormément le fait qu'il ne modifie que très peu le code déjà existant. Ensuite le deuxième schéma modélise l'ajout de tables liées à des promotions que l'application pourrait proposer, le lien de parrainage se ferait entre l'utilisateur qui souhaite profiter de la promotion et un compte administrateur. Enfin le troisième schéma, que mon maître de stage n'a pas encore validé, contient les corrections apportées au deuxième schéma et l'ajout des tables en lien avec les « produits » proposés par l'application.

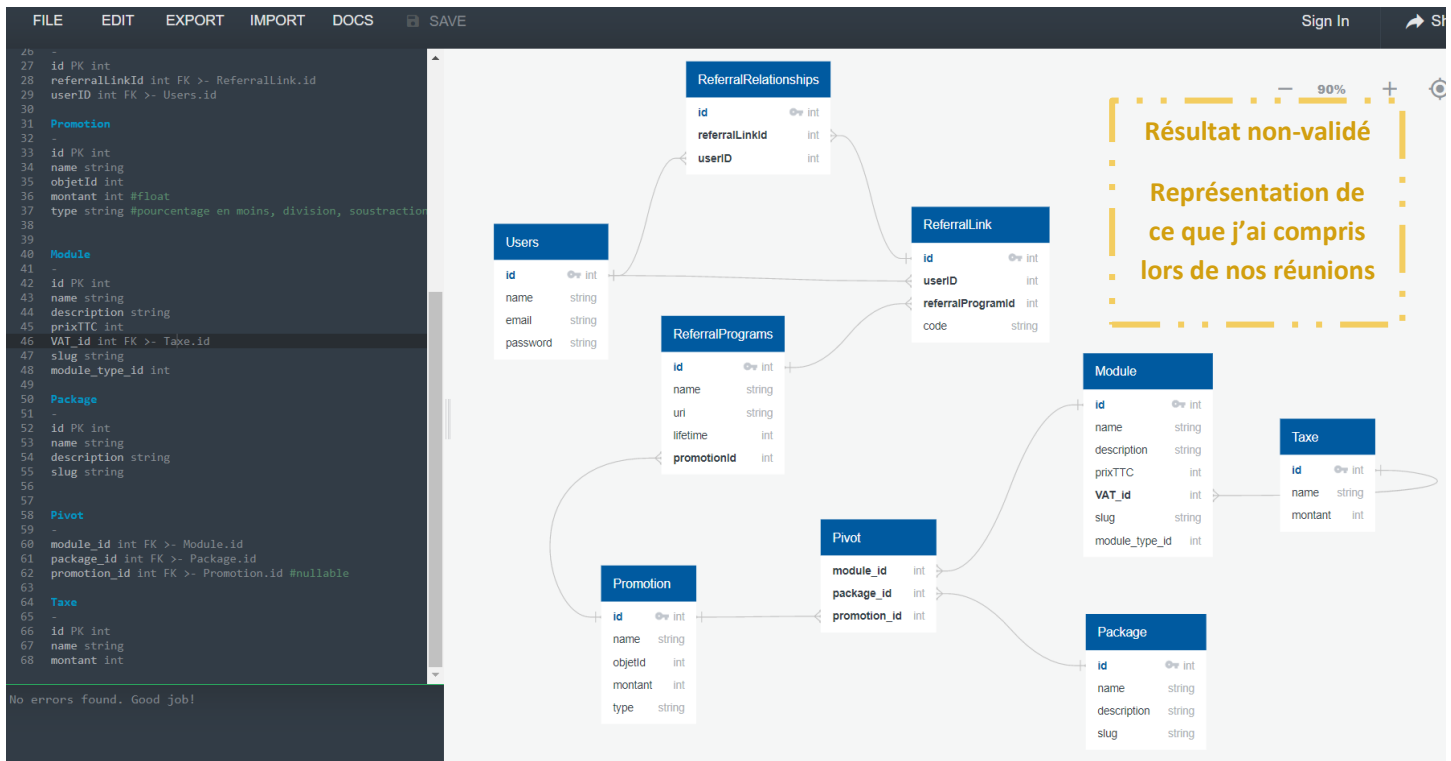
## ● Schéma #1 :



## ● Schéma #2 :



## ● Schéma #3 :



Ensuite en ce qui concerne le deuxième objectif j'ai repris le tutoriel que je suivais de zéro. Etant plus reposée, j'ai pu rapidement me rendre compte de mon erreur : la manière dont je récupérais le lien n'allait pas du tout. Il fallait que je récupère la donnée par le model de l'utilisateur avec la fonction `getReferrals()` et non pas par le model `ReferralLink` directement :


- Code modifié dans la vue du profile :

```

60 </div>
61
62 <!-- Email -->
63 <div class="col-span-6 sm:col-span-4">
64   <x-jet-label for="email" value="{{ __('Email') }}" />
65   <x-jet-input id="email" type="email" class="mt-1 block w-full" wire:model.defer="state.email" />
66   <x-jet-input-error for="email" class="mt-2" />
67 </div>
68
69 <!-- Referral Link -->
70 <div class="col-span-6 sm:col-span-4">
71   <x-jet-label for="referral-link" value="{{ __('Referral Link') }}" />
72   @forelse(auth()->user()->getReferrals() as $referral)
73     <x-jet-input id="referral-link" type="text" class="mt-1 block w-full" value="{{ $referral->link }}" /> {{-- value="{{ $referralLink }}" --}}
74   @empty
75     <x-jet-input id="referral-link" type="text" class="mt-1 block w-full" value="No referrals" />
76   @endforelse
77   <x-jet-input-error for="referral-link" class="mt-2" />
78 </div>
79 </x-slot>
80
81 <x-slot name="actions">
82   <x-jet-action-message class="mr-3" on="saved">
83     {{ __('Saved.') }}
84   </x-jet-action-message>

```

- Rendu visuel :


Dashboard   Users   Payments
Valentine Schalckens ▾

### Profile

#### Profile Information

Update your account's profile information and email address.

Name

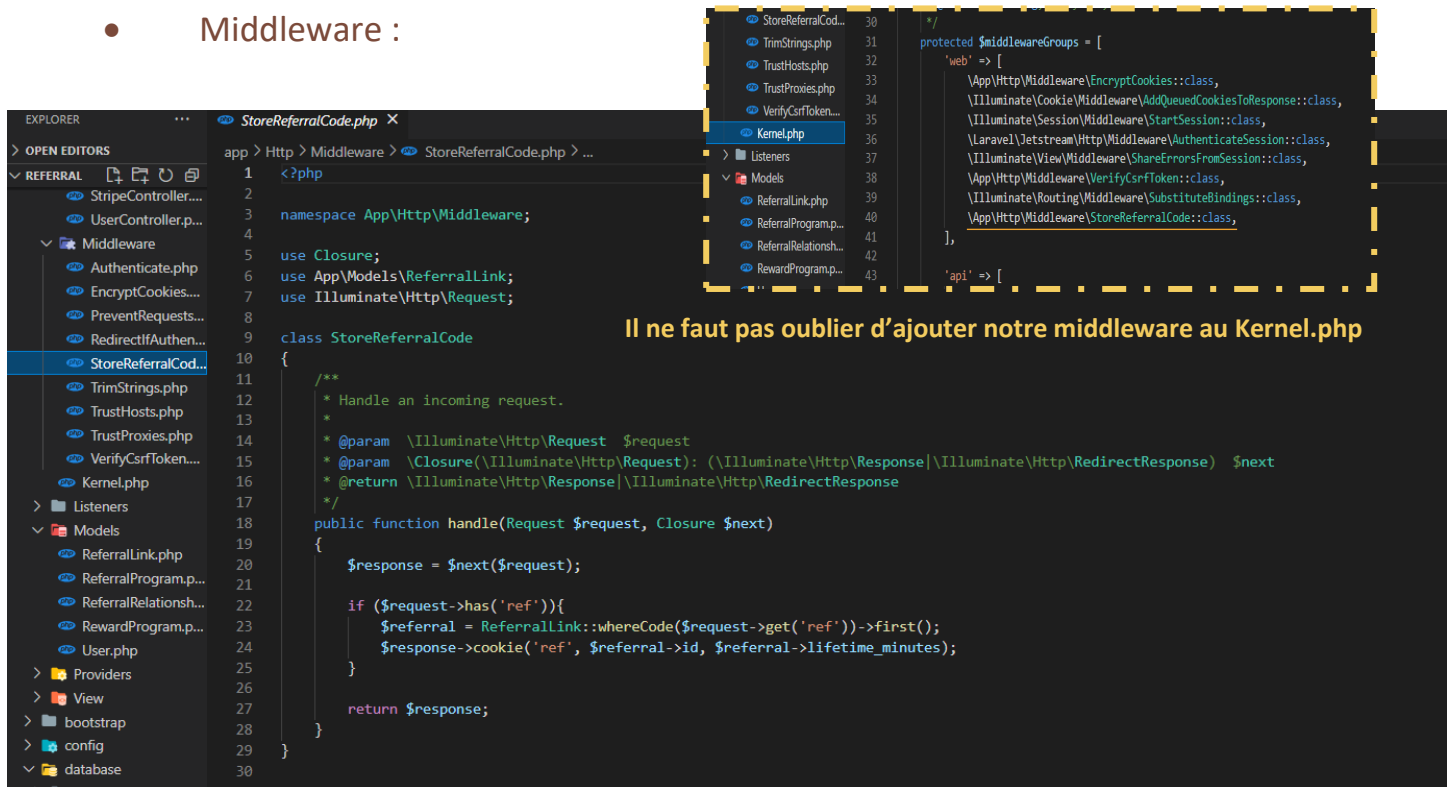
Email

Referral Link

SAVE

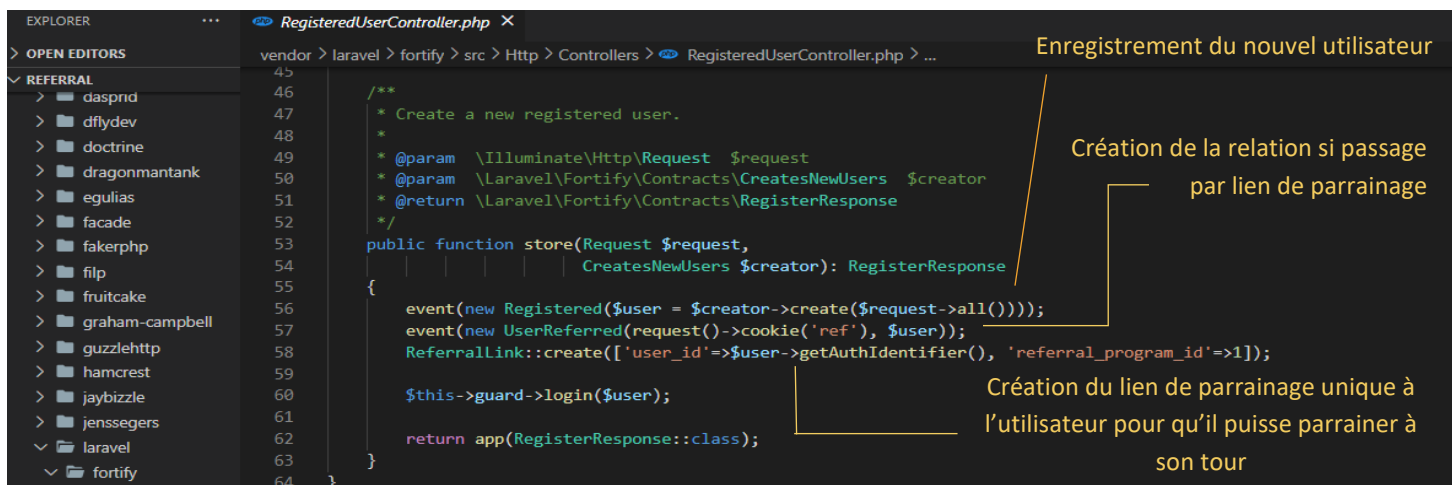
Maintenant que je pouvais récupérer et afficher le lien de parrainage de l'utilisateur, j'ai pu continuer le tutoriel. J'ai dû créer un Middleware pour enregistrer la relation parrain-parrainé entre l'utilisateur qui s'enregistre en passant par le lien de parrainage et l'utilisateur qui le fournit. Pour pouvoir garder les informations liées au lien (ici l'id du parrain et la durée de vie du lien), elles sont stockées dans un cookie pour pouvoir les récupérer d'une page à l'autre.

- Middleware :



Il ne faut pas oublier d'ajouter notre middleware au Kernel.php

- RegisteredUserController (controller appelé lors de l'enregistrement d'un nouvel utilisateur) :













Enregistrement du nouvel utilisateur

Création de la relation si passage par lien de parrainage

Création du lien de parrainage unique à l'utilisateur pour qu'il puisse parrainer à son tour






- Résultat en base de données :

	referral			id	name	email	email_verified_at	password
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	1	Valentine Schalkens	v.schalckens@gmail.com	NULL	\$2y\$10\$OM
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	2	Marc Dupont	nahum.sith@gmail.com	NULL	\$2y\$10\$1p6

☐ Tout cocher    Avec la sélection :  Éditer     Copier     Supprimer     Exporter

☐ Tout afficher    Nombre de lignes :     Filtrer les lignes:     Trier par clé :

**Opérations sur les résultats de la requête**

 Imprimer     Copier dans le presse-papiers     Exporter     Afficher le graphique     Créer une vue

referral

Nouvelle table

failed\_jobs

migrations

password\_resets

personal\_access\_tokens

referral\_links

referral\_programs

referral\_relationships

reward\_programs

sessions

id

name

uri

lifetime\_minutes

created\_at

updated\_at

<input type="checkbox"/>	Éditer	Copier	Supprimer	1	Sign-up Bonus	register	10080	2022-01-18 09:55:24	2022-01-18 09:55:24
--------------------------	--------	--------	-----------	---	---------------	----------	-------	---------------------	---------------------

Tout cocher

Avec la sélection :

Éditer

Copier

Supprimer

Exporter

☐ Tout afficher

Nombre de lignes : 

25

Filter les lignes:

Chercher dans cette table

Opérations sur les résultats de la requête

Imprimer

Copier dans le presse-papiers

Exporter

Afficher le graphique

Créer une vue

referral

- Nouvelle table
- failed\_jobs
- migrations
- password\_resets
- personal\_access\_tokens
- referral\_links
- referral\_programs
- referral\_relationships
- reward\_programs
- sessions

↔

id

user\_id

referral\_program\_id

code

<input type="checkbox"/>	Éditer	Copier	Supprimer	1	1	1	c620ca54-7843-11ec-9837-2c4d54c8df
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	2	1	edcdd3e8-7844-11ec-b06b-2c4d54c8df

☐ Tout cocher
 Avec la sélection :
 Éditer
 Copier
 Supprimer
 Exporter

☐ Tout afficher
 Nombre de lignes : 
 Filtrer les lignes: 
 Trier par clé :

Opérations sur les résultats de la requête

Imprimer
 Copier dans le presse-papiers
 Exporter
 Afficher le graphique
 Créer une vue

referral

Nouvelle table

failed\_jobs

migrations

password\_resets

personal\_access\_tokens

referral\_links

referral\_programs

**referral\_relationships**

reward\_programs

↔

id

referral\_link\_id

user\_id

created\_at

updated\_at

<input type="checkbox"/>	Éditer	Copier	Supprimer	1	1	2	2022-01-18 09:56:38	2022-01-18 09:56:38
--------------------------	--------	--------	-----------	---	---	---	---------------------	---------------------

⬆

☐ Tout cocher

Avec la sélection :

Éditer

Copier

Supprimer

Exporter

☐ Tout afficher

Nombre de lignes : 

25

Filtrer les lignes:

Chercher dans cette table

Opérations sur les résultats de la requête

Imprimer

Copier dans le presse-papiers

Exporter

Afficher le graphique

Créer une vue

Par la suite, n'ayant pas de nouvelle de la part de mon maître de stage sur mes prochains objectifs, j'ai effectué des recherches sur ce qu'était un slug et comment l'optimiser. Mon maître de stage avait évoqué ce terme lors de la création des schémas de base de données. Il m'en avait fait une brève explication et j'ai donc décidé de faire mes recherches :

\* Définition :

- C'est une **partie unique de l'URL** pour chaque page
- Wordpress est à l'origine du mot slug

\* Utilité :

- **Permet d'améliorer le référencement web**
- Permet donc l'**identification de la page**
- Format facilement lisible pour les utilisateurs et les moteurs de recherche
- Utile au classement effectué par les moteurs de recherche
- Aide les utilisateurs à se faire une idée du contenu de la page
- Procure un **avantage de référencement et d'expérience utilisateur**
- Les moteurs de recherche essaient de faire correspondre la recherche de l'utilisateur et le slug

\* Optimisation :

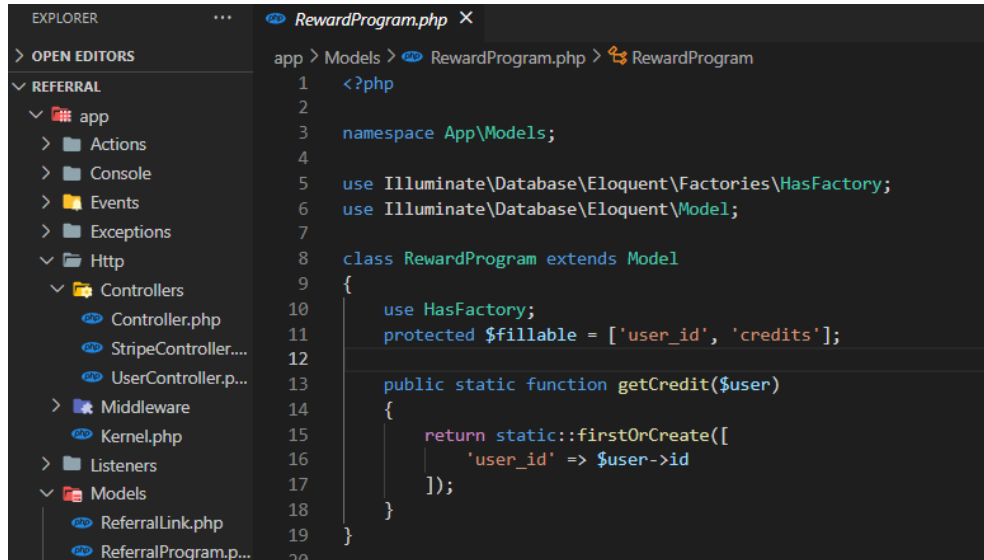
- Eviter le contenu dupliqué
- Inclure le (ou les) mot-clé(s) cible(s)
- Supprimer les mots qui n'ajoutent rien (pas assez précis pour mot clés, trop courant, pas utile à la compréhension ...) ex : mot de liaison
- Utiliser un tiret pour séparer les mots
- Limiter au maximum le nombre de mot
- Utiliser que des caractères minuscules

\* Optimisation en dehors de l'URL qui aide au référencement :

- 301 : Rediriger les URL brisées
- Ajouter un favicon
- Utiliser des URL canoniques
- Bloquer les URL incorrectes
- Référencer aux MDR quelles sont les pages adaptées aux mobiles
- Favoriser les contenus les plus importants le plus proche de la racine

Après ces recherches je suis revenue sur mon projet et j'ai essayé d'implémenter les récompenses par le biais de crédits et je voulais dans l'onglet « Users » de mon application test afficher le(s) parrain(s), le(s) parrainé(s) et les crédits obtenus. Je n'ai pas pu faire cet affichage car j'ai eu des nouvelles de mon maître de stage et donc ma prochaine mission.

- Model :

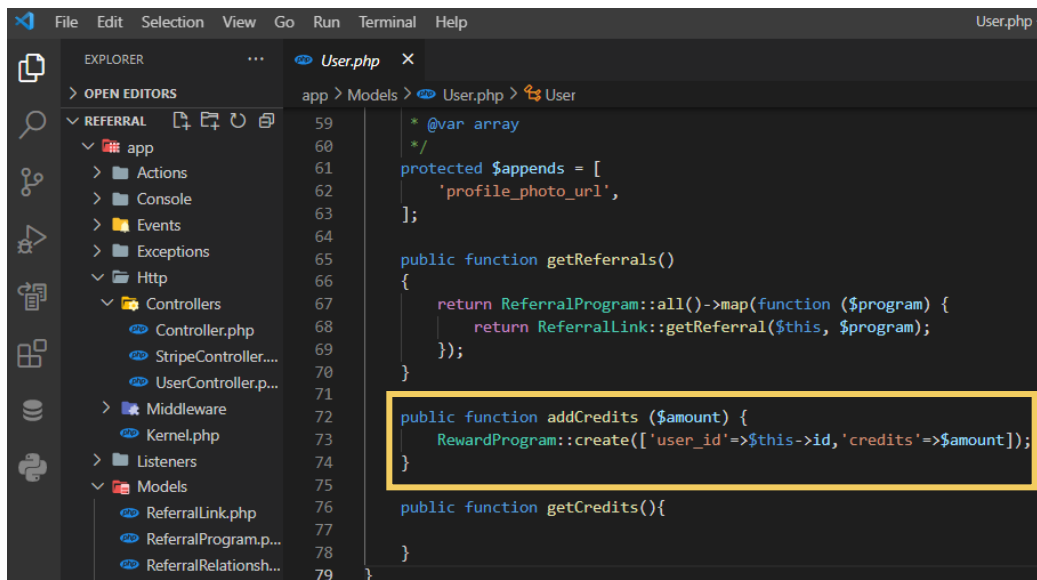


```

1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class RewardProgram extends Model
9  {
10     use HasFactory;
11     protected $fillable = ['user_id', 'credits'];
12
13     public static function getCredit($user)
14     {
15         return static::firstOrCreate([
16             'user_id' => $user->id
17         ]);
18     }
19 }

```

- Fonction ajoutée au model User pour attribuer les crédits :



```

59  * @var array
60  */
61  protected $appends = [
62      'profile_photo_url',
63  ];
64
65  public function getReferrals()
66  {
67      return ReferralProgram::all()->map(function ($program) {
68          return ReferralLink::getReferral($this, $program);
69      });
70  }
71
72  public function addCredits ($amount) {
73      RewardProgram::create(['user_id'=>$this->id,'credits'=>$amount]);
74  }
75
76  public function getCredits(){
77  }
78
79 }

```



- Ajout des crédits aux utilisateurs lors de la création de la relation parrain-parrainé dans le « Listener » de cet évènement :

```

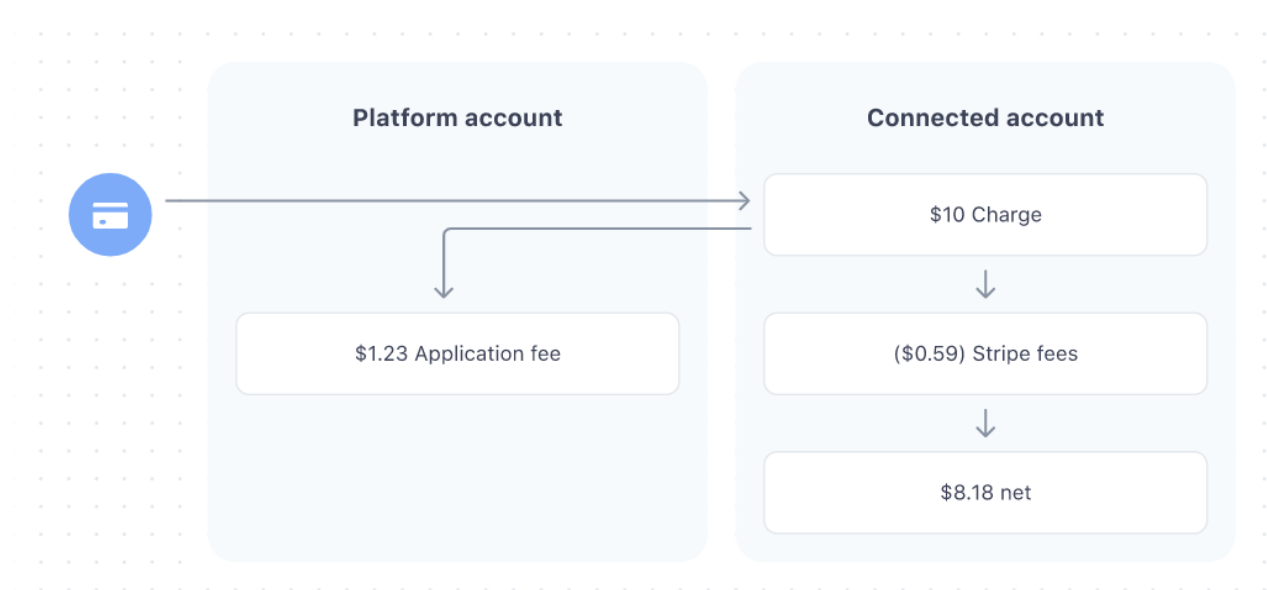
1 <?php
2
3 namespace App\Listeners;
4
5 use App\Events\UserReferred;
6 use App\Models\ReferralLink;
7 use App\Models\ReferralRelationship;
8 use Illuminate\Queue\InteractsWithQueue;
9 use Illuminate\Contracts\Queue\ShouldQueue;
10
11 class RewardUser
12 {
13     /**
14      * Create the event listener.
15      *
16      * @return void
17      */
18     public function __construct()
19     {
20         //
21     }
22
23     /**
24      * Handle the event.
25      *
26      * @param \App\Events\UserReferred $event
27      * @return void
28      */
29     public function handle(UserReferred $event)
30     {
31         $referral = ReferralLink::find($event->referralId);
32         if (!is_null($referral)) {
33             ReferralRelationship::create(['referral_link_id' => $referral->id, 'user_id' => $event->user->id]);
34
35             // Example
36             if ($referral->program->name === 'Sign-up Bonus') {
37                 // User who was sharing link
38                 $provider = $referral->user;
39                 $provider->addCredits(15);
40                 // User who used the link
41                 $user = $event->user;
42                 $user->addCredits(20);
43             }
44         }
45     }
46 }
47
48

```

- Résultat en base de données :

id	user_id	credits	created_at	updated_at
1	1	15	2022-01-18 09:56:38	2022-01-18 09:56:38
2	2	20	2022-01-18 09:56:38	2022-01-18 09:56:38

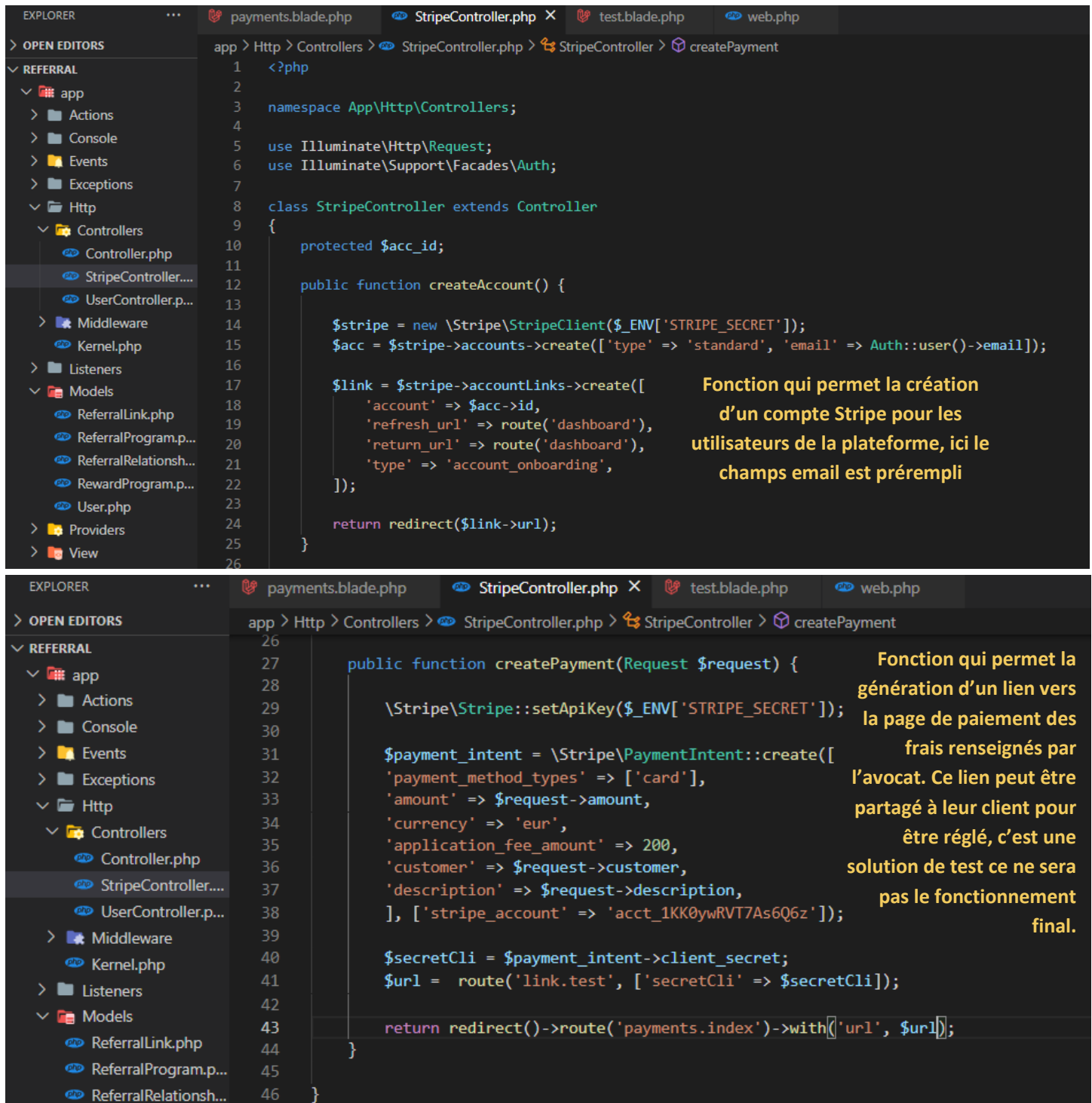
Enfin pour cette semaine je suis passé sur mon troisième objectif. Mon maitre de stage m’a prévenu que celui-ci ne serait pas facile et c’est le cas. Je me suis pas mal perdue dans la documentation conséquente de Stripe, et j’ai dû de ce fait faire pas mal de fois le point avec mon maitre de stage pour être sûre de partir dans la bonne direction (j’ai aussi parcouru énormément de liens et de vidéos car j’étais pas mal perdue, voir lien 1<sup>ère</sup> page). Le but étant, par l’API de Stripe, de se servir du service Stripe Connect nous permettant de faire le pont entre les avocats et leurs clients au niveau des paiements sans apparaître sur les factures. Voici le schéma de cette idée que l’on peut trouver sur le site de [Stripe](https://stripe.com/fr/connect) :



Pour mettre en place ce système, il fallait tout d’abord créer le compte de notre plateforme mais aussi faire en sorte que nos clients puissent se créer un compte depuis notre plateforme qui les relierait au compte de la plateforme par Stripe Connect. Puis il fallait mettre en place pour les avocats la création d’une « facture » à payer pour leur client et un moyen pour leur client de payer sur notre plateforme. J’ai pu suivre un tutoriel Stripe : « [Enable other businesses to accept payments directly](#) » qui correspondait aux attentes de mon maitre de stage.

## Code

- Controller Stripe :



```

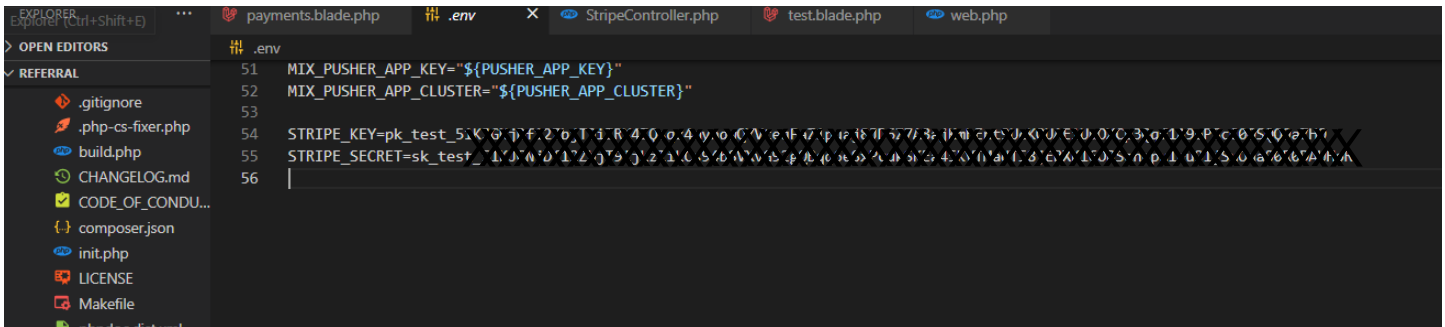
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\Auth;
7
8  class StripeController extends Controller
9  {
10     protected $acc_id;
11
12     public function createAccount() {
13
14         $stripe = new \Stripe\StripeClient($_ENV['STRIPE_SECRET']);
15         $acc = $stripe->accounts->create(['type' => 'standard', 'email' => Auth::user()->email]);
16
17         $link = $stripe->accountLinks->create([
18             'account' => $acc->id,
19             'refresh_url' => route('dashboard'),
20             'return_url' => route('dashboard'),
21             'type' => 'account_onboarding',
22         ]);
23
24         return redirect($link->url);
25     }
26
27     public function createPayment(Request $request) {
28
29         \Stripe\Stripe::setApiKey($_ENV['STRIPE_SECRET']);
30
31         $payment_intent = \Stripe\PaymentIntent::create([
32             'payment_method_types' => ['card'],
33             'amount' => $request->amount,
34             'currency' => 'eur',
35             'application_fee_amount' => 200,
36             'customer' => $request->customer,
37             'description' => $request->description,
38             ['stripe_account' => 'acct_1KK0yRVT7As6Q6z']);
39
40         $secretCli = $payment_intent->client_secret;
41         $url = route('link.test', ['secretCli' => $secretCli]);
42
43         return redirect()->route('payments.index')->with(['url', $url]);
44     }
45 }
46

```

**Fonction qui permet la création d'un compte Stripe pour les utilisateurs de la plateforme, ici le champs email est prérempli**

**Fonction qui permet la génération d'un lien vers la page de paiement des frais renseignés par l'avocat. Ce lien peut être partagé à leur client pour être réglé, c'est une solution de test ce ne sera pas le fonctionnement final.**

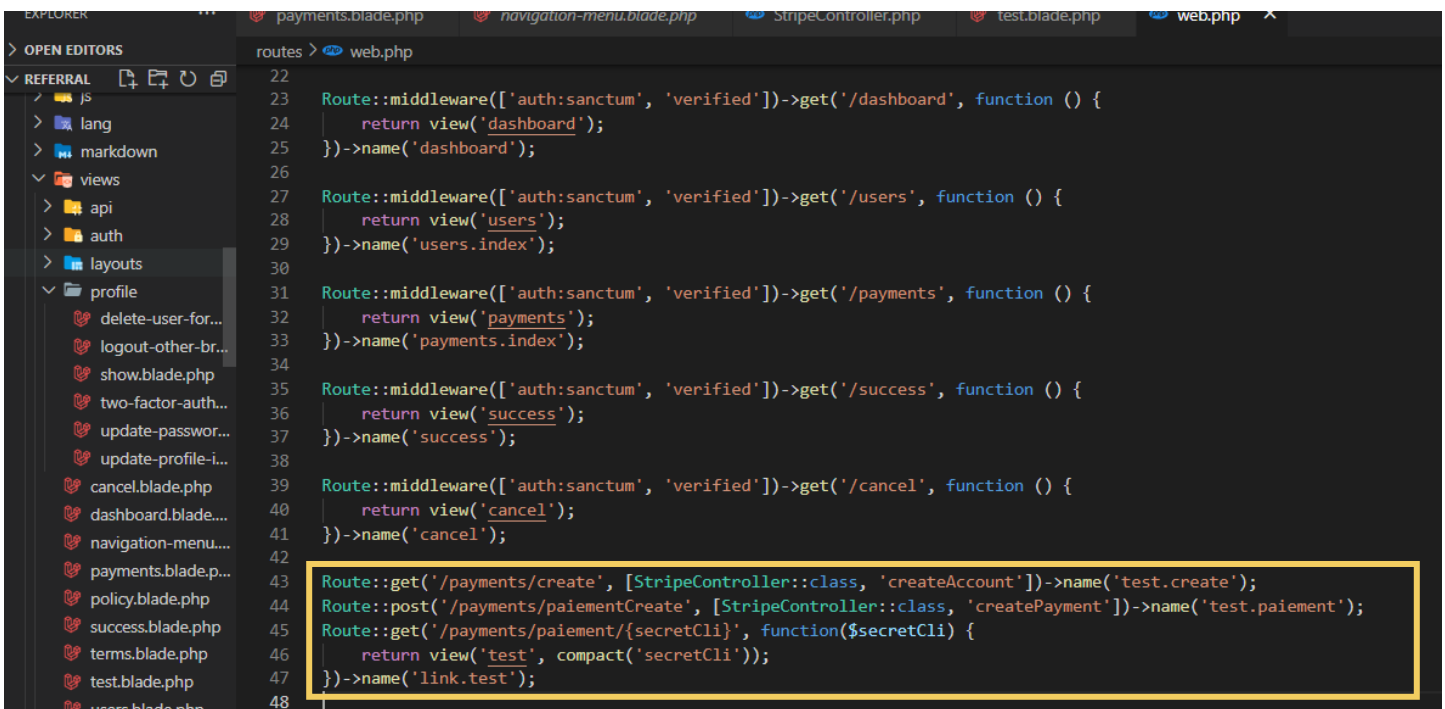
- Variable d'environnement pour les clés de l'API Stripe :



```

51 MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
52 MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
53
54 STRIPE_KEY=pk_test_5KtG1F2oT3R4Qo4wioCvief0p0t0z73f1k00e0u0E0U0C03o19Pc0S0e0h1
55 STRIPE_SECRET=sk_test_1Uj0d12079j21C56N0b3f0e0e04u0a040N0d130c0A1J0S0n0101S006060h0
56
  
```

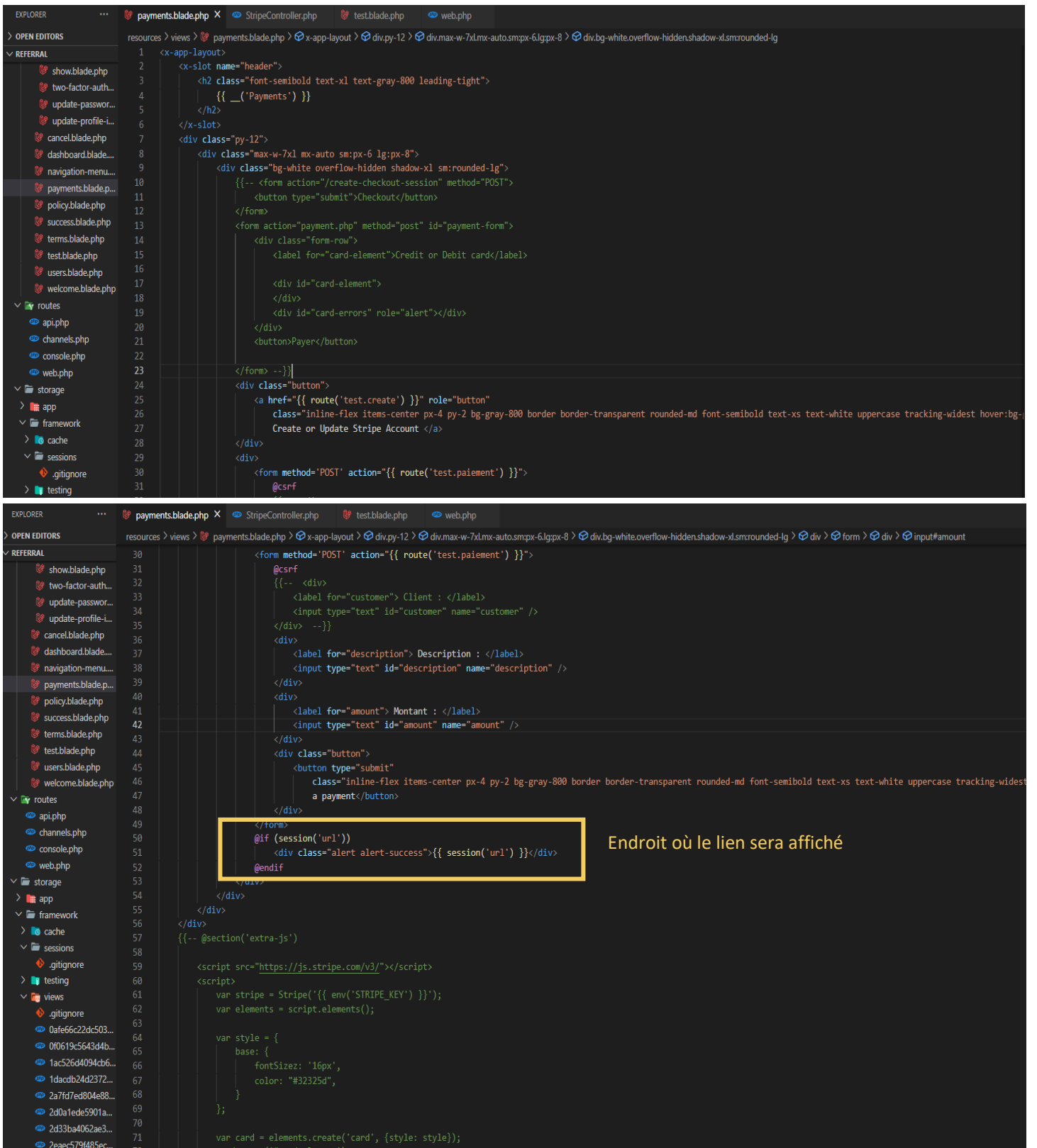
- Routes :



```

22
23 Route::middleware(['auth:sanctum', 'verified'])->get('/dashboard', function () {
24     return view('dashboard');
25 })->name('dashboard');
26
27 Route::middleware(['auth:sanctum', 'verified'])->get('/users', function () {
28     return view('users');
29 })->name('users.index');
30
31 Route::middleware(['auth:sanctum', 'verified'])->get('/payments', function () {
32     return view('payments');
33 })->name('payments.index');
34
35 Route::middleware(['auth:sanctum', 'verified'])->get('/success', function () {
36     return view('success');
37 })->name('success');
38
39 Route::middleware(['auth:sanctum', 'verified'])->get('/cancel', function () {
40     return view('cancel');
41 })->name('cancel');
42
43 Route::get('/payments/create', [StripeController::class, 'createAccount'])->name('test.create');
44 Route::post('/payments/paiementCreate', [StripeController::class, 'createPayment'])->name('test.paiement');
45 Route::get('/payments/paiement/{secretCli}', function($secretCli) {
46     return view('test', compact('secretCli'));
47 })->name('link.test');
48
  
```

## ● Vue de création de compte/ paiement :



```

1 <x-app-layout>
2   <x-slot name="header">
3     <h2 class="font-semibold text-xl text-gray-800 leading-tight">
4       {{ __('Payments') }}
5     </h2>
6   </x-slot>
7   <div class="py-12">
8     <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
9       <div class="bg-white overflow-hidden shadow-xl sm:rounded-lg">
10        <!-- <form action="/create-checkout-session" method="POST">
11          <button type="submit">Checkout</button>
12        </form>
13        <form action="payment.php" method="post" id="payment-form">
14          <div class="form-row">
15            <label for="card-element">Credit or Debit card</label>
16            <div id="card-element">
17              </div>
18            <div id="card-errors" role="alert"></div>
19          </div>
20          <button>Pay</button>
21        </form> -->
22      </div>
23    </div>
24    <div class="button">
25      <a href="{{ route('test.create') }}" role="button"
26        class="inline-flex items-center px-4 py-2 bg-gray-800 border border-transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest hover:bg-
27        Create or Update Stripe Account </a>
28    </div>
29    <div>
30      <form method="POST" action="{{ route('test.paielement') }}">
31        @csrf
32        <div>
33          <!-- <div>
34            <label for="customer"> Client : </label>
35            <input type="text" id="customer" name="customer" />
36          </div> -->
37          <div>
38            <label for="description"> Description : </label>
39            <input type="text" id="description" name="description" />
40          </div>
41          <div>
42            <label for="amount"> Montant : </label>
43            <input type="text" id="amount" name="amount" />
44          </div>
45          <div class="button">
46            <button type="submit"
47              class="inline-flex items-center px-4 py-2 bg-gray-800 border border-transparent rounded-md font-semibold text-xs text-white uppercase tracking-widest
48              a payment</button>
49          </div>
50        </form>
51        @if (session('url'))
52          <div class="alert alert-success">{{ session('url') }}</div>
53        @endif
54      </div>
55    </div>
56  </div>
57  <!-- @section('extra-js')
58  </section>
59  <script src="https://js.stripe.com/v3/"></script>
60  <script>
61    var stripe = Stripe('{{ env('STRIPE_KEY') }}');
62    var elements = stripe.elements();
63
64    var style = {
65      base: {
66        fontSize: '16px',
67        color: '#32325d',
68      }
69    };
70
71    var card = elements.create('card', {style: style});
72    card.mount('#card-element');
73  </script>
74</x-app-layout>

```

Endroit où le lien sera affiché

- Vue pour le règlement du paiement :

**Éléments HTML permettant la saisie d'une carte bancaire générée par Stripe JS**

```

1 <x-slot name="header">
2   <h2 class="font-semibold text-xl text-gray-800 leading-tight">
3     {{ __('Dashboard') }}
4   </h2>
5 </x-slot>
6
7 <div class="py-12">
8   <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
9     <div class="bg-white overflow-hidden shadow-xl sm:rounded-lg">
10      <script src="https://js.stripe.com/v3/"></script>
11      <form id="payment-form">
12        <div id="card-element">
13          <!-- Elements will create input elements here -->
14        </div>
15
16        <!-- We'll put the error messages in this element -->
17        <div id="card-errors" role="alert"></div>
18
19        <button id="submit">Pay</button>
20        <input type="hidden" id="secretCli" value="{{ $secretCli }}" />
21      </form>
22    </div>
23
24    var stripe = Stripe(
25      'pk_test_51KJGjDf12ZbjT91ZRX4ZQsqz4mynqwQYVteqFaZ1puaj8JD67A8ajKmbEnt9UcKOUAEiUuOXcP3Qqf1A9ePScC6
26    );
27
28    var elements = stripe.elements();
29    var style = {
30      base: {
31        color: "#32325d",
32      }
33    };
34
35    var card = elements.create("card", {
36      style: style
37    });
38    card.mount("#card-element");
39
40    card.on("change", function(event) {
41      var displayError = document.getElementById("card-errors");
42      if (event.error) {
43        displayError.textContent = event.error.message;
44      } else {
45        displayError.textContent = '';
46      }
47    });
48
49    // ... (rest of the JavaScript code)
50
51    </div>
52  </div>
53 </x-app-layout>

```

## Visuels :

- Visuel de la vue de création de compte/ paiement :

Payments

CREATE OR UPDATE STRIPE ACCOUNT

Description : Frais de test

Montant : 6000

CREATE A PAYMENT

Donnée entrée en centime, amélioré par la suite

Payments

CREATE OR UPDATE STRIPE ACCOUNT

Description :

Montant :

CREATE A PAYMENT

Lien vers la page de paiement

[http://referral-jet/payments/paiement/pi\\_3KKtvQRVT7As6Q6z0zCulQ0d\\_secret\\_aW7stzXoFy6MK3wU5Ulp4EMfD](http://referral-jet/payments/paiement/pi_3KKtvQRVT7As6Q6z0zCulQ0d_secret_aW7stzXoFy6MK3wU5Ulp4EMfD)

- Visuel Stripe de création de compte pour les avocats :

connect.stripe.com/.../4shQhZsJddd

Test UH

Test UH a établi un partenariat avec Stripe pour des paiements sécurisés.

← Return to Test UH

Mode test

Premiers pas avec Stripe

Lorsqu'un formulaire est renseigné pour le compte d'une entreprise, il doit être renseigné par le propriétaire de cette entreprise ou une personne y assumant des responsabilités importantes.

E-mail

nahum.sith@gmail.com

Continuer →

Exemple de champ prérempli par notre application

Vous avez un compte Stripe ? Vous pouvez utiliser la même adresse e-mail.

Portail de connexion Stripe que rencontre le client lorsqu'il clique sur « CREATE OR UPDATE STRIPE ACCOUNT »

Powered by stripe

Français (FR)

Laravel Test UH

Rechercher...

Créer Aide

Accueil Paiements Solde Clients Produits Rapports Plus

Développeurs Mode test

Test UH a accès à de nouveaux outils pour mieux gérer votre compte. [En savoir plus](#)

**Heureux de vous revoir !**

Pour démarrer sur Stripe, consultez notre [documentation pour les développeurs](#) ou [découvrez nos recommandations](#).

Personnalisez votre Dashboard

Ajoutez votre logo

Chargez un logo PNG ou JPG

Terminer

Démarrer avec Stripe

- Visuel de la vue pour le règlement du paiement :

Non sécurisé | referral-jet/payments/paiement/pi\_3KKtvQRVT7As6Q6z0zCulQ0d\_secret\_aW7stzXoFy6MK3wU5Ulp4EMfD

Dashboard Users Payments Marc Dupont

Dashboard

VISA 4242 4242 4242 4242 05 / 24 152 83130

Pay

- Résultat sur le site Stripe :

fr.wikipedia.org

Rechercher...

Créer Aide

Accueil Paiements Solde Clients Produits Rapports Connect Plus

Développeurs Mode test

v.schalckens@gmail.com En attente Paiements Désactivés Virements Désactivés Solde total 1521,47 € Copier l'ID

**Activité**

**Solde**

Envoyer les fonds

Solde total 1521,47 € EUR Fonds suspendus 1521,47 € EUR En transit vers la banque 0,00 € EUR

**Activité**

Paiements Transferts Virements Frais perçus

MONTANT	DESCRIPTION	CLIENT	DATE
60,00 €	Réussi ✓ Frais de test		23 janv. à 01:29
500,00 €	Réussi ✓ frais d'avocat		21 janv. à 22:45
10,00 €	Réussi ✓ ch_3KKMpLRVT7As6Q6z1TFKPw7k		21 janv. à 15:11
1000,00 €	Réussi ✓ ch_3KKMPjRVT7As6Q6z1fgEeNw5		21 janv. à 13:41
10,00 €	Réussi ✓ ch_3KKJr3RVT7As6Q6z1sq2yCn2		21 janv. à 10:57

Visuel sur les transactions depuis le compte de la plateforme



fr.wikipedia.org

Rechercher...

Créer

Aide

Accueil

Palements

Solde

Clients

Produits

Rapports

Connect

Plus

DONNÉES TEST

Développeurs

Mode test

←

S

v.schalckens@gmail.com

En attente

Palements Désactivés

Virements Désactivés

Solde total 1521,47 €

Copier

1°ID

Activité

Identité

Paramètres

Développeurs

Palements

Factures

Abonnements

Clients

Produits

Solde

Envoyer les fonds

Solde total

Fonds suspendus

En transit vers la banque

1521,47 € EUR

1521,47 € EUR

0.00 € EUR

Commissions prises par la plateforme par transaction/paiement

Activité

Palements

Transferts

Virements

Frais perçus

MONTANT	DESCRIPTION	DATE
2,00 €	Frais de test	23 janv. à 01:29
2,00 €	frais d'avocat	21 janv. à 22:45
1,23 €	ch_3KKNpLRVT7As6Q6z1TfKPw7k	21 janv. à 15:11
5,00 €	ch_3KKMPjRVT7As6Q6z1fgEeNw5	21 janv. à 13:41
1,23 €	ch_3KKJrJRV7As6Q6z1sq2yCn2	21 janv. à 10:57