# Laravel 8 Jetstream Livewire CRUD avec modal tailwind css

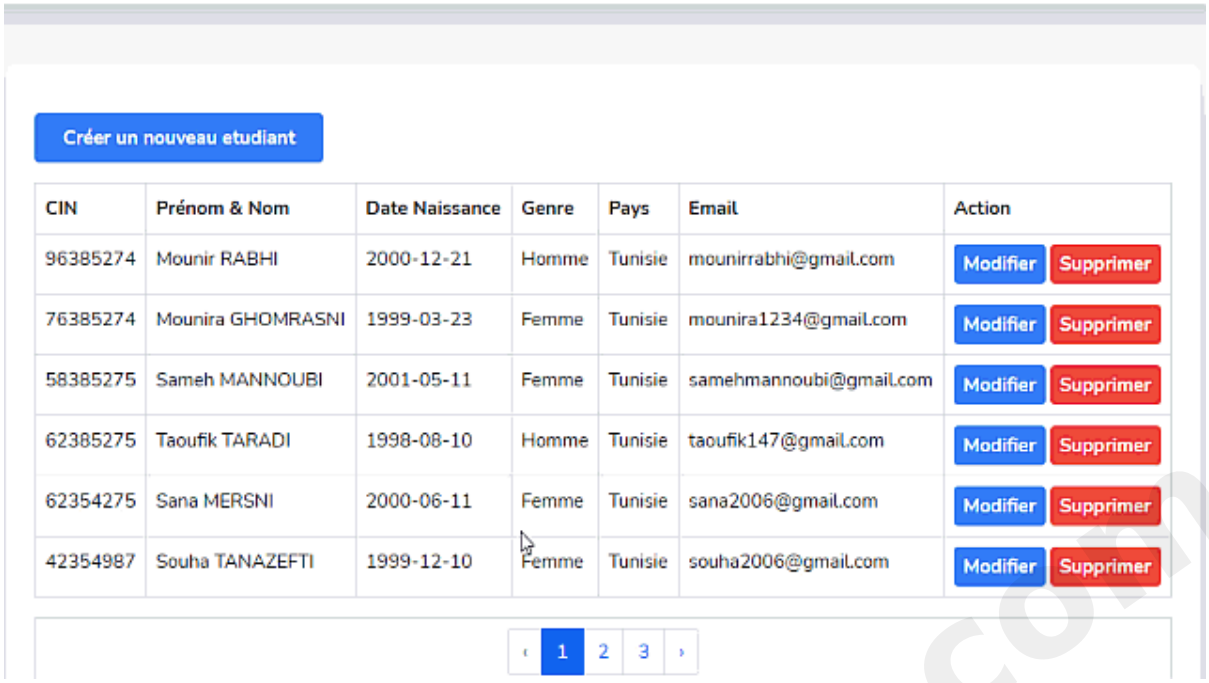
1. ## Objectifs

   - Être capable de créer des opérations CRUD livewire avec jetstream dans une application modal

2. ## Qu'est-ce que Tailwind CSS?

   - **Tailwind CSS** est un framework CSS qui permet de construire rapidement des designs personnalisés.
   - Plutôt que de proposer des composants pré-conçus, Tailwind offre des classes utilitaires de bas niveau qui permettent de construire des designs et éléments entièrement personnalisables.
   - Tous les éléments de base sont donnés, à vous de les modifier pour obtenir ce que vous souhaitez. Les couleurs, la taille, les bordures et bien d'autres éléments peuvent être modifiés.
   - Tailwind CSS propose une documentation complète de chaque élément afin de les personnaliser à votre guise et sans accroc. Tous les utilitaires fournis par Tailwind proposent une variante responsive. Pour trouver les classes responsives, il suffit de chercher {screen} dans le balisage.
   - Ce framework permet ainsi d'avoir le contrôle sur chaque élément et ainsi créer des sites uniques et qui correspondent à vos attentes.
   - Dans ce tutoriel, nous verrons comment nous pouvons utiliser Livewire pour implémenter la pagination ,la suppression à l'aide de la confirmation modale, l'ajout à l'aide de modal et l'édition à l'aide de modal.
   - Nous utiliserons Jestream Livewire Stack, qui utilise également Tailwind CSS pour implémenter toutes ces fonctionnalités. Nous utiliserons des modaux et d'autres composants fournis par Jetstream.
   - Voyons donc un processus étape par étape pour mettre en œuvre cela et vous obtiendrez la mise en page ci-dessous:

apcpedagogie.com




3. ## Créer un nouveau projet Laravel8

   - Dans la première étape, nous devons obtenir une nouvelle application de version laravel à l'aide de la commande ci-dessous. Alors ouvrons le terminal et exécutons la commande ci-dessous.

```
composer create-project --prefer-dist laravel/laravel livewire-jetstream-projet02
```

4. Quand on installe Laravel, si on veut déjà un système d'authentification complet, on installe Jetstream qui va lui-même installer Livewire comme dépendance :

5. ## Créer Jetstream Auth avec Livewire

   - Installer **jetstream** à l'aide de la commande ci-dessous, alors ouvrir le terminal et exécuter la commande ci-dessous:

```
composer require laravel/jetstream
```

6. Maintenant, créer une **authentification** à l'aide de la commande ci-dessous, vous pouvez créer **une connexion** de base, une **inscription** et une **vérification par e-mail**. si vous souhaitez créer **une gestion d'équipe**, vous devez passer le **paramètre d'addition**. vous pouvez voir les commandes ci-dessous:

apcpedagogie.com

```
php artisan jetstream:install livewire
```

7. Maintenant, installer le package **node js**:

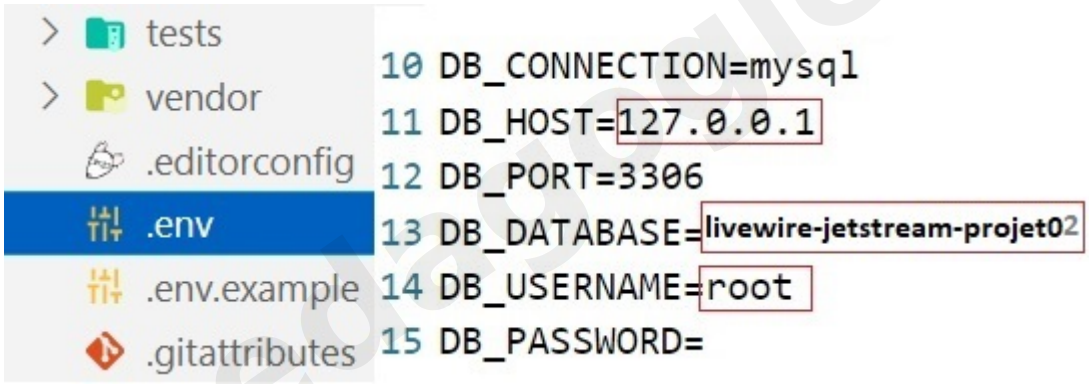
```
npm install
```

8. Exécuter le package:

```
npm run dev
```

9. ## Configurer la base de données

   - Pour configurer la base de données. Ouvrir le fichier ".env"
   - changer le nom de la base de données ("**livewire-jetstream-projet02**"), le nom d'utilisateur et le mot de passe.




   - Si votre base de données est de type MySql5.6.XXX ou autre version antérieur, vous risquez d'obtenir un erreur en cours de migration, une solution (la plus simple mais pas la meilleure) consiste à modifier le fichier **database.php** dans le dossier config :
     - `'engine' => null,`
     - Devient :`'engine' => 'InnoDB ROW_FORMAT=DYNAMIC',`
     - Ou tout autre type de engine paramétré lors de l'install de MySql...

10. ## Créer une migration et un modèle

    1. **Une migration**

- Dans cette étape, nous devons créer la table de migration pour les **etudiants** à l'aide de la commande artisanale ci-dessous. Alors ouvrons le terminal et exécutons la commande:

```
php artisan make:migration create_etudiants_table
```

2. Après l'exécution de cette commande un fichier sera crée dans le chemin suivant: `database/migrations/2021_04_11_085510_create_etudiants_table.php`
3. Modifier la méthode up comme suit:

```
public function up()
    {
        Schema::create('etudiants', function (Blueprint $table) {
            $table->integer('cin')->primary();
            $table->string('prenomNom');
            $table->date('dteNaissance');
            $table->string('genre');
            $table->string('pays');
            $table->string('email');
            $table->timestamps();
        });
    }
```

4. Pour exécuter la migration pour un seul fichier, utiliser la commande suivante:`php artisan migrate --path=database/migrations/2021_04_11_085510_create_etudiants_table.php`

5. **Un model**

- Dans cette étape, nous devons créer lemodèle **Etudiant** à l'aide de la commande artisanale ci-dessous. Alors ouvrons le terminal et exécutons la commande:

```
php artisan make:model Etudiant
```

6. Modifier le fichier "**app\Models\Etudiant.php**" comme suit:

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class Etudiant extends Model
{
    use HasFactory;
```

apcpedagogie.com

```
    public $primaryKey = 'cin';
    public $incrementing = false;
    protected $fillable = [
        'cin', 'prenomNom', 'dteNaissance', 'genre', 'pays' , 'email',
    ];
}
```

## 11. La population (seeding)

- **Écriture des seeders**
  - Pour générer un seeder, exécutez la commande Artisan : `php artisan make:seeder EtudiantsTableSeeder`.
  - Tous les seeders générés par le Framework seront placés dans le répertoire **database/seeders:**

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

class EtudiantsTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        \DB::table('Etudiants')->insert([
            [
                'cin' => 96385274,
                'prenomNom' => 'Mounir RABHI',
                'dteNaissance' => '2000-12-21',
                'genre' => 'Homme',
                'pays' => 'Tunisie',
                'email' => 'mounirrabhi@gmail.com',
                'created_at'=>now(),
                'updated_at'=>now(),
```

apcpedagogie.com

```
            ],
            [
                'cin' => 76385274,
                'prenomNom' => 'Mounira GHOMRASNI',
                'dteNaissance' => '1999-03-21',
                'genre' => 'Femme',
                'pays' => 'Tunisie',
                'email' => 'mounira1234@gmail.com',
                'created_at'=>now(),
                'updated_at'=>now(),
            ],
            [
                'cin' => 58385275,
                'prenomNom' => 'Sameh MANNOUBI',
                'dteNaissance' => '2001-05-11',
                'genre' => 'Femme',
                'pays' => 'Tunisie',
                'email' => 'samehmannoubi@gmail.com',
                'created_at'=>now(),
                'updated_at'=>now(),
            ],
            [
                'cin' => 62385275,
                'prenomNom' => 'Taoufik TARADI',
                'dteNaissance' => '1998-08-10',
                'genre' => 'Homme',
                'pays' => 'Tunisie',
                'email' => 'taoufik147@gmail.com',
                'created_at'=>now(),
                'updated_at'=>now(),
            ],
            [
                'cin' => 62354275,
                'prenomNom' => 'Sana MERSNI',
                'dteNaissance' => '2000-06-11',
                'genre' => 'Femme',
                'pays' => 'Tunisie',
                'email' => 'sana2006@gmail.com',
                'created_at'=>now(),
                'updated_at'=>now(),
            ],
            [
                'cin' => 42354987,
```

apcpedagogie.com

```
                'prenomNom' => 'Souha TANAZEFTI',
                'dteNaissance' => '1999-12-10',
                'genre' => 'Femme',
                'pays' => 'Tunisie',
                'email' => 'souha2006@gmail.com',
                'created_at'=>now(),
                'updated_at'=>now(),
            ],
        [
                'cin' => 98754987,
                'prenomNom' => 'Mounira HAMDI',
                'dteNaissance' => '2001-12-10',
                'genre' => 'Femme',
                'pays' => 'Tunisie',
                'email' => 'mounirahamdi2006@gmail.com',
                'created_at'=>now(),
                'updated_at'=>now(),
            ],
        [
                'cin' => 36263587,
                'prenomNom' => 'Sourour TANAZEFTI',
                'dteNaissance' => '1998-02-17',
                'genre' => 'Femme',
                'pays' => 'Tunisie',
                'email' => 'sourour1998@gmail.com',
                'created_at'=>now(),
                'updated_at'=>now(),
            ],
            [
                'cin' => 27263887,
                'prenomNom' => 'Noura HWEDI',
                'dteNaissance' => '2000-02-19',
                'genre' => 'Femme',
                'pays' => 'Tunisie',
                'email' => 'nourahwedi1@gmail.com',
                'created_at'=>now(),
                'updated_at'=>now(),
            ],
        ]);
    }
}
```

apcpedagogie.com

12. **Génération des seeders**

    - Vous pouvez utilisez la commande suivante pour insérer des données dans le attbles de votre base de données : `php artisan db:seed --class=EtudiantsTableSeeder`

## 13. Créer un composant (component)

- Dans cette étape, nous devons **créer un composant de etudiant livewire** à l'aide de la commande ci-dessous. Alors ouvrons le terminal et exécutons la commande ci-dessous:
  - `php artisan make:livewire etudiants`
- Maintenant, ils ont créé des fies sur les deux chemins:
  - **app/Http/Livewire/Etudiants.php**
    **ressources/vues/livewire/etudiants.blade.php**

## 14. Mettre à jour le fichier de composant (component)

- Ici, nous écrirons les méthodes `render()`, `create()`, `openModal()`, `closeModal()`, `resetInputFields()`, `store()`, `edit()` et `delete()` pour notre application CRUD.
- Alors, mettons à jour le fichier comme suit:

apcpedagogie.com

```php
<?php
namespace App\Http\Livewire;
use Livewire\Component;
use App\Models\Etudiant;
use Livewire\WithPagination;
class Etudiants extends Component
{
  use WithPagination;
  protected $paginationTheme = 'bootstrap';
  public  $cin, $prenomNom,$dteNaissance,$genre
  ,$pays,$email, $Etudiant_id;
  public $isOpen = 0;

  public function render()
  {
      $this->etudiants = Etudiant::all();
      return view('livewire.Etudiants', [
          'etudiants' => Etudiant::paginate(6),
      ]);
  }
  public function create()
  {
      $this->resetInputFields();
      $this->openModal();
  }
  public function openModal()
  {
      $this->isOpen = true;
  }
  public function closeModal()
  {
      $this->isOpen = false;
  }
  private function resetInputFields(){
      $this->cin = '';
      $this->prenomNom = '';
      $this->dteNaissance = '';
      $this->genre = '';
      $this->pays = '';
      $this->email = '';
      $this->Etudiant_id = '';
  }
  public function store()
  {
      $this->validate([
          'cin' => 'required',
          'prenomNom' => 'required',
          'dteNaissance' => 'required',
          'genre' => 'required',
          'pays' => 'required',
          'email' => 'required',
      ]);
      Etudiant::updateOrCreate(
          ['cin' => $this->Etudiant_id], [
              'cin' => $this->cin,
              'prenomNom' => $this->prenomNom,
              'dteNaissance' => $this->dteNaissance,
              'genre' => $this->genre,
              'pays' => $this->pays,
              'email' => $this->email,
          ]);
      session()->flash('message',
          $this->Etudiant_id ?
           'Etudiant mis à jour avec succès.'
            : 'Etudiant créé avec succès.');
      $this->closeModal();
      $this->resetInputFields();
  }
  public function edit($id)
  {
      $Etudiant = Etudiant::findOrFail($id);
      $this->Etudiant_id = $id;
      $this->cin = $Etudiant->cin;
      $this->prenomNom = $Etudiant->prenomNom;
      $this->dteNaissance = $Etudiant->dteNaissance;
      $this->genre = $Etudiant->genre;
      $this->pays = $Etudiant->pays;
      $this->email = $Etudiant->email;
      $this->openModal();
  }
  public function delete($id)
  {
      Etudiant::find($id)->delete();
      session()->flash('message',
      'Etudiant supprimé avec succès.');
  }
}
```

## 15. Mettre à jour les fichiers Blade

1. **Fichier `etudiants.blade.php`**

   - Ici, nous mettrons à jour la liste suivante de fichiers pour notre page de liste, créer une page.
   - Alors, mettons à jour le fichier suivant:
     **resources/views/livewire/etudiants.blade.php**

```blade
<x-slot name="header">
    <h2 class="font-semibold text-xl text-gray-800 leading-tight">
        livewire jetstream projet01 -apcpedagogie.com)
    </h2>
</x-slot>
<div class="py-12">
    <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
        <div class="bg-white overflow-hidden shadow-xl sm:rounded-lg px-4 py-4">
            @if (session()->has('message'))
                <div class="bg-teal-100 border-t-4 border-teal-500 rounded-b
                px-4 py-3 shadow-md my-3" role="alert">        text-teal-900
                  <div class="flex">
                    <div>
                      <p class="text-sm">{{ session('message') }}</p>
                    </div>
                  </div>
                </div>
            @endif                                    hover:bg-blue-700 text-white
            <button wire:click="create()" class="bg-blue-500
            font-bold py-2 px-4 rounded my-3">Créer un nouveau etudiant</button>
            @if($isOpen)
                @include('livewire.create')
            @endif
            <table class="table table-bordered w-full">
                <thead>
                    <tr class="border px-4 py-2"><th>CIN</th><th>Prénom & Nom</th>
                  <th>Date Naissance</th><th>Genre</th><th>Pavs</th><th>Email</th>
                </thead>                                          <th>Action</th></tr>
                <tbody>
                    @foreach($etudiants as $etudiant)
                    <tr>
                        <td>{{ $etudiant->cin }}</td>
                        <td >{{ $etudiant->prenomNom }}</td>
                        <td >{{ $etudiant->dteNaissance }}</td>

                        <td >{{ $etudiant->genre }}</td>
                        <td >{{ $etudiant->pays }}</td>
                        <td >{{ $etudiant->email }}</td>

                        <td >
                        <button wire:click="edit({{ $etudiant->cin }})"
                            class="bg-blue-500 hover:bg-blue-700 text-white
                            font-bold py-1 px-2 rounded">Modifier</button>
                           <button wire:click="delete({{ $etudiant->cin }})"
                            class="bg-red-500 hover:bg-red-700 text-white
                            font-bold py-1 px-2 rounded">Supprimer</button>
                        </td>
                    </tr>
                    @endforeach
                </tbody>
            </table>
            <div class="d-flex pb-0 pt-2 border  justify-content-center">
                {{ $etudiants->links() }}
            </div>
        </div>
    </div>
</div>
```

apcpedagogie.com

2. **Fichier `create.blade.php`**

```
<div class="fixed z-10 inset-0 ease-out duration-400 ">
    <div class="justify-center c pt-4 px-4 pb-20 text-center sm:block sm:p-0">
        <div class="fixed inset-0 transition-opacity">
            <div class="absolute inset-0 bg-gray-500 opacity-75"></div>
        </div>
        <!-- Cet élément consiste à inciter le navigateur à centrer le contenu
modal. -->
        <span class="hidden sm:inline-block sm:align-middle sm:h-screen"></span>
        <div class="inline-block align-middle bg-white rounded-lg
        text-left overflow-hidden shadow-xl transform transition-all
        sm:my-8 sm:align-middle max-w-screen-md sm:w-full"
        role="dialog" aria-modal="true" aria-labelledby="modal-headline">
            <form>
                <div class="bg-white px-4 pt-5 pb-4 sm:p-6 sm:pb-4">
                    <div class="grid grid-cols-6 gap-6">
                        <div class="col-span-6 sm:col-span-3">
                            <label for="exampleFormControlInput1" class="block text-
gray-700 text-sm font-bold mb-2">CIN:</label>
                            <input type="text" class="shadow appearance-none border
rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none
focus:shadow-outline"
                            id="cin" placeholder="Saisir le numéro du CIN"
wire:model="cin">
                            @error('cin') <span class="text-red-500">{{ $message
}}</span>@enderror
                        </div>

                        <div class="col-span-6 sm:col-span-3">
                            <label for="exampleFormControlInput1" class="block text-
gray-700 text-sm font-bold mb-2">Nom & Prenom:</label>
                            <input type="text" class="shadow appearance-none border
rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none
focus:shadow-outline"
                            id="prenomNom" placeholder="Saisir le nom et le prénom"
wire:model="prenomNom">
                            @error('prenomNom') <span class="text-red-500">{{
$message }}</span>@enderror
                        </div>
                    </div>
                    <div>
```

```
                        <div class="grid grid-cols-6 gap-6">
                            <div class="col-span-6 sm:col-span-3">
                                <label for="exampleFormControlInput1" class="block
text-gray-700 text-sm font-bold mb-2">Date Naissance:</label>
                                <input type="date" class="shadow appearance-none
border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none
focus:shadow-outline"
                                id="dteNaissance" placeholder="Date naissance"
wire:model="dteNaissance">
                                @error('dteNaissance') <span class="text-red-500">{{
$message }}</span>@enderror
                            </div>
                            <div class="col-span-6 sm:col-span-3">
                                <label for="exampleFormControlInput2" class="block
text-gray-700 text-sm font-bold mb-2">Genre:</label>
                                <select class="shadow appearance-none border rounded
w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-
outline"
                                id="genre" wire:model="genre">
                                <option>Homme</option>
                                <option>Femme</option>
                              </select>
                                @error('genre') <span class="text-red-500">{{
$message }}</span>@enderror
                            </div>
                        </div>
                        <div class="grid grid-cols-6 gap-6">
                            <div class="col-span-6 sm:col-span-3">
                                <label for="exampleFormControlInput1" class="block
text-gray-700 text-sm font-bold mb-2">Pays:</label>
                                <input type="text" class="shadow appearance-none
border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none
focus:shadow-outline"
                                id="pays" placeholder="Enter Title"
wire:model="pays">
                                @error('pays') <span class="text-red-500">{{
$message }}</span>@enderror
                            </div>
                            <div class="col-span-6 sm:col-span-3">
                                <label for="exampleFormControlInput2" class="block
text-gray-700 text-sm font-bold mb-2">Email:</label>
                                <input type="email"  class="shadow appearance-none
border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none
```

apcpedagogie.com

```
focus:shadow-outline"
                                id="email" wire:model="email" placeholder="Enter
email">
                                @error('email') <span class="text-red-500">{{
$message }}</span>@enderror
                            </div>
                        </div>


                    </div>
                </div>
                <div class="bg-gray-50 px-4 py-3 sm:px-6 sm:flex sm:flex-row-
reverse">
                    <span class="mt-3 flex w-full rounded-md shadow-sm sm:ml-3 sm:w-
auto">
                        <button wire:click.prevent="store()" type="button"
                        class="inline-flex justify-center w-full rounded-md border
border-transparent
                        px-4 py-2 bg-green-600 text-base leading-6 font-medium text-
white shadow-sm
                        hover:bg-green-500 focus:outline-none focus:border-green-700
focus:shadow-outline-green transition
                        ease-in-out duration-150 sm:text-sm sm:leading-5">
                        Enregistrer
                        </button>
                    </span>
                    <span class="mt-3 flex w-full rounded-md shadow-sm sm:mt-0 sm:w-
auto">
                        <button wire:click="closeModal()" type="button"
                        class="inline-flex justify-center w-full rounded-md border
border-gray-300 px-4
                        py-2 bg-red-400 text-base leading-6 font-medium text-red-700
shadow-sm
                        hover:text-gray-500 focus:outline-none focus:border-blue-300
focus:shadow-outline-blue
                        transition ease-in-out duration-150 sm:text-sm sm:leading-
5">
                            Supprimer
                        </button>
                    </span>
                </div>
            </form>
        </div>
```

apcpedagogie.com

```
    </div>
</div>
<!--https://shuffle.dev/tailwind/classes-->
```

## 16. Mettre à jour les routes "`routes/web.php`"

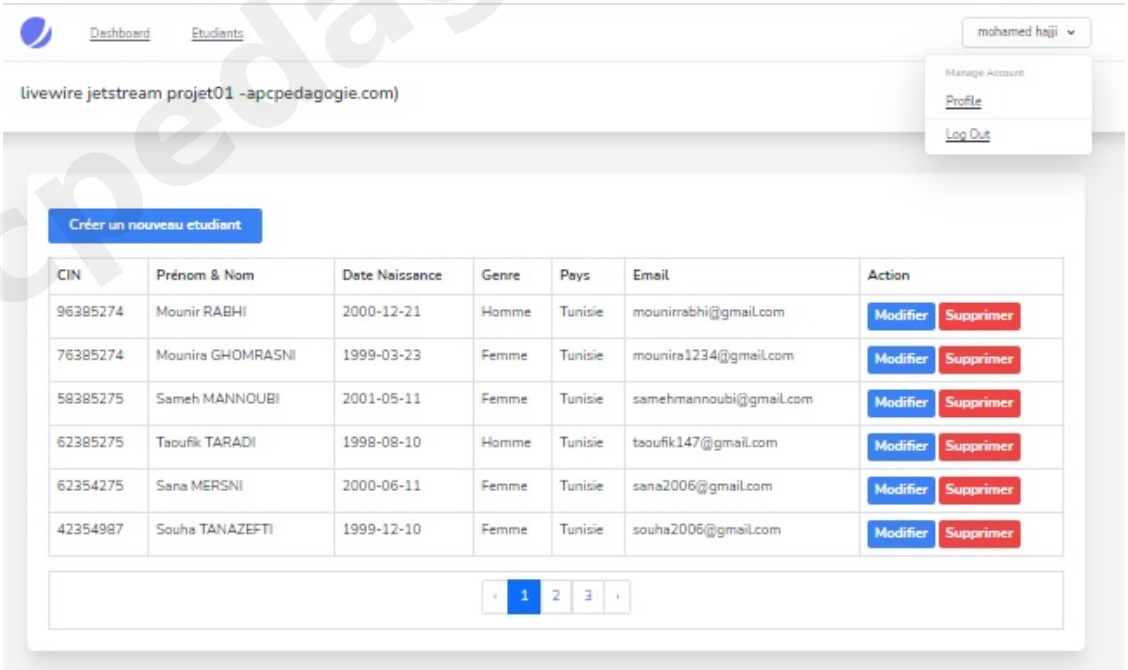
```
use App\Http\Livewire\Etudiants;
Route::get('etudiant', Etudiants::class);
```

## 17. Exécuter le projet

- Vous pouvez maintenant ouvrir l'URL ci-dessous sur votre navigateur:
  - `php artisan serve`
  - `localhost:8000/etudiant`

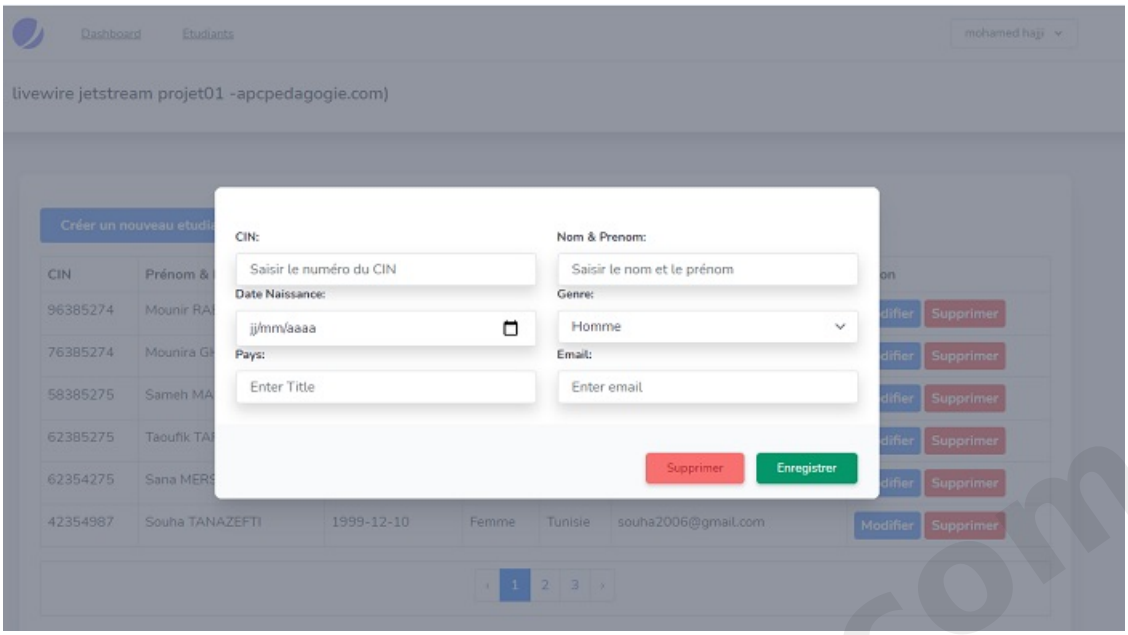
## 18. Les écrans

- **Liste des étudiants:**

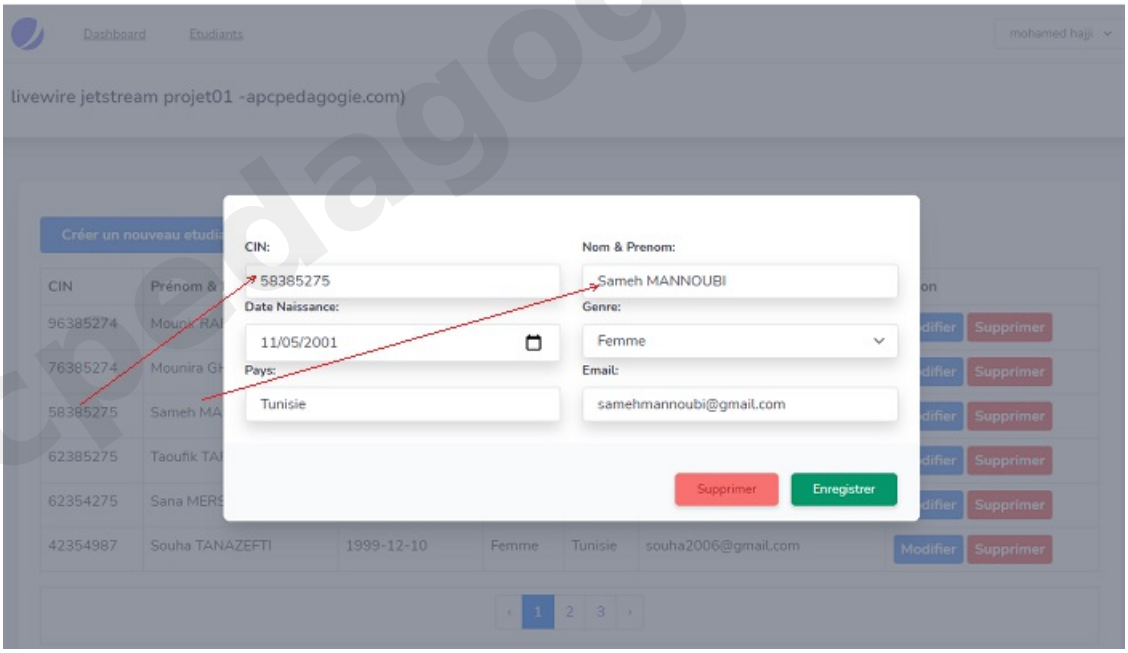



- **Formulaire d'ajout:**

apcpedagogie.com




- **Formulaire de modification:**




---

## Sommaire du cours Laravel

Source : https://apcpedagogie.com/laravel-8-jetstream-livewire-crud-avec-modal-tailwind-css · téléchargé le 07/11/2026 · © APC Pédagogie · apcpedagogie.com