



مكتب التكوين المهني وإنعاش الشغل  
Office de la Formation Professionnelle et de la  
Promotion du Travail



Direction Régionale Casablanca-Settat  
EVALUATION DE FIN DE MODULE  
AU TITRE DE L'ANNEE : 2022/2023

Filière : DEVWOFS  
Niveau : TS  
N° du module : M205  
Intitulé du module : développement back-end  
Date d'évaluation : 25/04/2023

Année de formation : 2A  
Epreuve : Th  
Durée : 2H  
Variante : V2  
Barème / 40

Dans un pays on veut gérer les habitants d'une ville donnée. En utilisant Laravel Framework créer une application qui répond aux besoins en suivant les questions.

Ville (id, ville, nombreHabitats)

Habitant (id, cin, nom, prenom, #ville\_id, photo)

NB : ville\_id fait référence à la table villes

- 1- Ecrire la commande pour créer le model Ville avec migration. (3pts)
- 2- Ecrire la commande pour créer le model Habitant avec migration. (3pts)
- 3- Ecrire la commande pour créer le contrôleur HabitantController. (3pts)
- 4- Dans la class HabitantController Créer les méthodes suivantes :
  - a. Method index qui retourne vers la vue "index" la liste des habitants. (3pts)
  - b. Method create qui retourne vers la vue "create" le formulaire pour ajouter un habitant. (3pts)
  - c. Et la méthode "store" pour sauvegarder les données. (4pts)
  - d. Method edit qui retourne vers la vue "edit" les infos d'un habitant. (4pts)
  - e. Method update pour confirmer la mise à jour. (4pts)
  - f. Method destroy pour supprimer un habitant. (4pts)
- 5- Créer les routes pour toutes les méthodes du contrôleur HabitantController. (4pts)
- 6- Ecrire le code pour la vue index avec les liens vers ajouter, éditer, et supprimer (5pts).

Bahae Eddine Halim

# 1. Create the model Ville with migration:

- Create a new file called `Ville.php` in the `app` directory:

```
'''
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Ville extends Model
{
    protected $table = 'villes';
    protected $primaryKey = 'id';
    public $timestamps = false;

    // Define the relationships or any other methods you need
}
'''
```

- Create a new migration file using the command: `php artisan make:migration create_villes_table --create=villes``.
- Open the newly created migration file and define the table schema for the `villes` table:

```
'''
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateVillesTable extends Migration
{
    public function up()
    {
        Schema::create('villes', function (Blueprint $table) {
            $table->increments('id');
            $table->string('ville');
            $table->integer('nombreHabitats');
        });
    }

    public function down()
```

```
{
    Schema::dropIfExists('villes');
}
}
...
}
```

- Run the migration using the command: ``php artisan migrate``.

## 2. Create the model Habitant with migration:

- Create a new file called ``Habitant.php`` in the ``app`` directory:

```
...
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Habitant extends Model
{
    protected $table = 'habitants';
    protected $primaryKey = 'u';
    public $timestamps = false;

    // Define the relationships or any other methods you need
}
...
}
```

- Create a new migration file using the command: ``php artisan make:migration create_habitants_table --create=habitants``.
- Open the newly created migration file and define the table schema for the ``habitants`` table:

```
...
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateHabitantsTable extends Migration
{
    public function up()
    {
```

```

Schema::create('habitants', function (Blueprint $table) {
    $table->integer('u');
    $table->string('cin');
    $table->string('nom');
    $table->string('prenom');
    $table->unsignedInteger('ville_id');
    $table->string('photo');
    $table->foreign('ville_id')->references('id')->on('villes');
});
}

```

```

public function down()
{
    Schema::dropIfExists('habitants');
}
}
...

```

- Run the migration using the command: ``php artisan migrate``.

### 3. Create the controller HabitantController:

- Create a new file called ``HabitantController.php`` in the ``app/Http/Controllers`` directory:

```

...
<?php

namespace App\Http\Controllers;

use App\Habitant;
use Illuminate\Http\Request;

class HabitantController extends Controller
{
    public function index()
    {
        $habitants = Habitant::all();
        return view('habitants.index', compact('habitants'));
    }

    public function create()
    {
        return view('habitants.create');
    }

    public function store(Request $request)

```

```

    {
        // Validate the input and store the habitant
    }

    public function edit($u)

    {
        $habitant = Habitant::findOrFail($u);
        return view('habitants.edit', compact('habitant'));
    }

    public function update(Request $request, $u)
    {
        // Validate the input and update the habitant
    }

    public function destroy($u)
    {
        $habitant = Habitant::findOrFail($u);
        $habitant->delete();
        return redirect()->route('habitants.index');
    }
}
...

```

## 4. Define the routes:

- Open the `routes/web.php` file.
- Add the necessary route definitions for the methods in the `HabitantController` class:

```

...
<?php

use Illuminate\Support\Facades\Route;

Route::get('/habitants',
'App\Http\Controllers\HabitantController@index')->name('habitants.index');
Route::get('/habitants/create',
'App\Http\Controllers\HabitantController@create')->name('habitants.create');
Route::post('/habitants',
'App\Http\Controllers\HabitantController@store')->name('habitants.store');
Route::get('/habitants/{u}/edit',
'App\Http\Controllers\HabitantController@edit')->name('habitants.edit');
Route::put('/habitants/{u}',
'App\Http\Controllers\HabitantController@update')->name('habitants.update');

```

```
Route::delete('/habitants/{u}',
'App\Http\Controllers\HabitantController@destroy')->name('habitants.destroy');
''
```

## 5. Create the views:

- Create a new folder called `habitants` inside the `resources/views` directory.
- Inside the `habitants` folder, create the following view files:

1. Create a new folder called `habitants` inside the `resources/views` directory:

- Open your terminal or command prompt.
- Navigate to the `resources/views` directory of your Laravel project.
- Create a new directory called `habitants` using the command: `mkdir habitants`.

2. Inside the `habitants` folder, create the following view files:

`index.blade.php`:

```
'''
<!-- resources/views/habitants/index.blade.php -->

<h1>List of Habitants</h1>

<a href="{{ route('habitants.create') }}">Add Habitant</a>

<table>
  <thead>
    <tr>
      <th>CIN</th>
      <th>Name</th>
      <th>Actions</th>
    </tr>
  </thead>
  <tbody>
    @foreach ($habitants as $habitant)
      <tr>
        <td>{{ $habitant->cin }}</td>
        <td>{{ $habitant->nom }} {{ $habitant->prenom }}</td>
        <td>
          <a href="{{ route('habitants.edit', $habitant->u) }}">Edit</a>
          <form action="{{ route('habitants.destroy', $habitant->u) }}" method="POST">
            @csrf
            @method('DELETE')
            <button type="submit">Delete</button>
          </form>
        </td>
      </tr>
    @endforeach
  </tbody>
</table>
```

```
        @endforeach
    </tbody>
</table>
'''
```

`create.blade.php`:

```
'''
<!-- resources/views/habitants/create.blade.php -->

<h1>Add Habitant</h1>

<form action="{{ route('habitants.store') }}" method="POST">
    @csrf
    <!-- Form fields for habitant creation -->
    <label for="cin">CIN:</label>
    <input type="text" name="cin" id="cin">
    <br>
    <label for="nom">Nom:</label>
    <input type="text" name="nom" id="nom">
    <br>
    <label for="prenom">Prénom:</label>
    <input type="text" name="prenom" id="prenom">
    <br>
    <label for="ville_id">Ville:</label>
    <select name="ville_id" id="ville_id">
        <!-- Populate with options from the "villes" table -->
        @foreach ($villes as $ville)
            <option value="{{ $ville->id }}">{{ $ville->ville }}</option>
        @endforeach
    </select>
    <br>
    <label for="photo">Photo:</label>
    <input type="text" name="photo" id="photo">
    <br>
    <button type="submit">Save</button>
</form>
'''
```

`edit.blade.php`:

```
'''
<!-- resources/views/habitants/edit.blade.php -->

<h1>Edit Habitant</h1>

<form action="{{ route('habitants.update', $habitant->u) }}" method="POST">
    @csrf
    @method('PUT')
    <!-- Form fields for habitant editing -->
```

```

<label for="cin">CIN:</label>
<input type="text" name="cin" id="cin" value="{{ $habitant->cin }}">
<br>
<label for="nom">Nom:</label>
<input type="text" name="nom" id="nom" value="{{ $habitant->nom }}">
<br>
<label for="prenom">Prénom:</label>
<input type="text" name="prenom
" id="prenom" value="{{ $habitant->prenom }}">
<br>
<label for="ville_id">Ville:</label>
<select name="ville_id" id="ville_id">
<!-- Populate with options from the "villes" table -->
@foreach ($villes as $ville)
    <option value="{{ $ville->id }}" {{ $habitant->ville_id == $ville->id ? 'selected' : " }}">
        {{ $ville->ville }}
    </option>
@endforeach
</select>
<br>
<label for="photo">Photo:</label>
<input type="text" name="photo" id="photo" value="{{ $habitant->photo }}">
<br>
<button type="submit">Save</button>
</form>
'''

```