TP : Laravel Bibliothèque 2

On va utiliser le projet « bibliothèque (005-biblio) » pour ce TP.

1 : Création du model « Book ».

Tapez la commande suivante dans votre CMD :

```
php artisan make:model Book -m
```

La commande va créer un modèle et un fichier de migration.

Modifier le fichier de migration comme suit :

```
        Schema::create('books', function (Blueprint $table) {
            $table->id();
            $table->string('designation')->unique()->require();
            $table->text('description');
            $table->string('type')->default('Texte');
            $table->string('langue')->default('Francais');
            $table->string('editeur')->default('Anonyme');
            $table->string('categorie')->default('Nouveau');
            $table->double('prix')->default('0');
            $table->string('auteur')->default('Anonyme');
            $table->string('cover')->default('no_cover.jpg');
            $table->timestamps();
        });
```

Et ajouter la liste des champs modifiables dans le modèle.

```
class Book extends Model
{
    use HasFactory;
    protected $fillable = ['designation', 'description', 'prix', 'auteur', 'cover', 'type',
'langue', 'editeur', 'categorie'];
}
```

Maintenant, créez la table dans la base de données en utilisant la commande suivante.

```
php artisan migrate
```

2 : Créer des routes et un contrôleur

Tout d'abord, créez le **BookController** à l'aide de la commande suivante.

```
php artisan make:controller BookController --resource
```

Notez que nous avons également ajouté **--resource**, qui définira six méthodes à l'intérieur du BookController, à savoir :

Index (utilisé pour afficher une liste de livres)

Create (montrera la vue avec un formulaire pour créer un livre)

Store (utilisé pour créer un livre dans la base de données. Note : la méthode create se soumet à la méthode store)

Show (affichera un livre spécifié)

Edit (affichera le formulaire pour éditer un livre. Le formulaire sera rempli avec les données existantes)

Update (Utilisé pour mettre à jour un livre dans la base de données. Note : edit se soumet à la méthode update)

Destroy (utilisé pour supprimer un livre spécifié)

Maintenant, dans le fichier **routes >> web.php**, ajoutez la ligne de code suivante.

```
Route::resource('book', BookController::class);
```

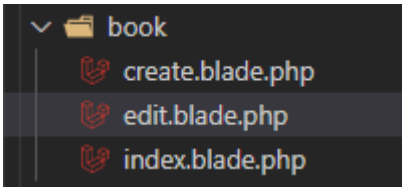
En ajoutant la ligne suivante, nous avons enregistré plusieurs routes pour notre application. Nous pouvons le vérifier en utilisant la commande suivante.

```
php artisan route:list
```

3 : créer les vues

Dans le dossier **Ressources >> Views>>book**, créez les trois fichiers de vues suivants.

- create.blade.php
- edit.blade.php
- index.blade.php
- show.blade.php



- Copier le contenu du fichier **books/details.blade.php** dans le nouveau fichier **book/show.blade.php**.
- Supprimer le fichier **details.blade.php**.

4 : Créer les fichiers Seeder et Factory.

Créer les deux fichiers avec les commandes :

```
php artisan make:seeder BookSeeder
php artisan make:factory BookFactory --model=Book
```

Modifier le fichier BookFactory.php :

```
public function definition(): array
{
 return [
     'designation' => $this->faker->unique()->sentence(3),
     'description' => $this->faker->paragraph(),
     'type' => $this->faker->randomElement(['Texte', 'Image', 'Audio', 'Video']),
     'langue' => $this->faker->randomElement(['Arabe', 'Francais', 'Anglais', 'Espagnol',
'Allemand']),
     'editeur' => $this->faker->company(),
     'categorie' => $this->faker->randomElement(['Classique', 'Science Fiction',
'Fantastique', 'Horreur', 'Romance', 'Mystere']),
     'prix' => $this->faker->randomFloat(2, 0, 900),
     'auteur' => $this->faker->name(),
     'cover' => 'no_cover.jpg',
        ];
}
```

Modifier le fichier BookSeeder.php :

```
public function run(): void

{
   \App\Models\Book::factory()->count(40)->create();
}
```

Lancer la commande :

```
php artisan db:seed --class=BookSeeder
```

5 : Affichez les données (l'action index).

Nous devons écrire la fonction index du BookController pour renvoyer une vue d'index avec des données extraites d'une base de données. Écrivez le code suivant dans la fonction index().

```php
    public function index()
    {
        $books = Book::all();
        return view('book.index',compact('books'));
    }
```

Ajouter un lien vers la page dans le menu du Header.

Et maintenant, modifier le contenu du fichier index.blade.php :

```
@extends('layouts.app')
@section('title', 'Acceuil')
@section('content')
    <div class="py-24 bg-white">
        <div class="container mx-auto px-4 sm:px-6 lg:px-8">
            <div class="text-center mb-12">
                <span class="text-blue-600 font-semibold">Tous les livres</span>
                <h2 class="mt-2 text-3xl font-extrabold text-gray-900 sm:text-
4xl">Parcourir les livres</h2>
            </div>
            <div class="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-
8">
                @foreach ($books as $book)
                    <div
                        class="group relative bg-white rounded-xl shadow-sm hover:shadow-
2xl transition-all duration-300 border border-gray-200 overflow-hidden">
                        <!-- Book Cover -->
                        <div class="aspect-[3/4] overflow-hidden bg-gray-100">
                            <img src="{{ asset('covers/' . $book->id . '.jpg') }}" alt="{{
$book->designation }} cover"
                                class="w-full h-full object-cover group-hover:scale-105
transition-transform duration-300"
                                onerror="this.src='{{ asset('covers/book-cover-
placeholder.png') }}'">
                        </div>

                        <!-- Card Content -->
                        <div class="p-5 space-y-4">
                            <h3 class="font-bold text-lg text-gray-900 line-clamp-2
leading-tight">
                                {{ $book->designation }}
                            </h3>
```

```
                            <p class="text-sm text-gray-600">{{ $book->auteur }}</p>
                            <!-- Action Buttons -->
                            <div class="flex justify-center gap-6 pt-3 border-t border-
gray-100">
                                <!-- View -->
                                <a href="{{ route('book.show', $book->id) }}"
                                    class="group/btn flex flex-col items-center gap-1 text-
gray-600 hover:text-blue-600 transition"
                                    title="View Book">
                                    <div class="p-3 rounded-full bg-blue-50 group-
hover/btn:bg-blue-100 transition">
                                        <svg class="w-5 h-5" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                                            <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                                d="M15 12a3 3 0 11-6 0 3 3 0 016 0z" />
                                            <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                                d="M2.458 12C3.732 7.943 7.523 5 12 5c4.478
0 8.268 2.943 9.542 7-1.274 4.057-5.064 7-9.542 7-4.477 0-8.268-2.943-9.542-7z" />
                                        </svg>
                                    </div>
                                    <span class="text-xs font-medium">View</span>
                                </a>

                                <!-- Edit -->
                                <a href="{{ route('book.edit', $book->id) }}"
                                    class="group/btn flex flex-col items-center gap-1 text-
gray-600 hover:text-amber-600 transition"
                                    title="Edit Book">
                                    <div class="p-3 rounded-full bg-amber-50 group-
hover/btn:bg-amber-100 transition">
                                        <svg class="w-5 h-5" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                                            <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
                                                d="M11 5H6a2 2 0 00-2 2v11a2 2 0 002 2h11a2
2 0 002-2v-5m-1.414-9.414a2 2 0 112.828 2.828L11.828 15H9v-2.828l8.586-8.586z" />
                                        </svg>
                                    </div>
                                    <span class="text-xs font-medium">Edit</span>
                                </a>

                                <!-- Delete -->
                                <form action="{{ route('book.destroy', $book->id) }}"
method="POST" class="inline">
                                    @csrf
                                    @method('DELETE')
                                    <button type="submit"
                                        onclick="return confirm('Are you sure you want to
delete this book?')"
                                        class="group/btn flex flex-col items-center gap-1
text-gray-600 hover:text-red-600 transition"
                                        title="Delete Book">
                                        <div class="p-3 rounded-full bg-red-50 group-
hover/btn:bg-red-100 transition">
                                            <svg class="w-5 h-5" fill="none"
stroke="currentColor" viewBox="0 0 24 24">
                                                <path stroke-linecap="round" stroke-
linejoin="round" stroke-width="2"
```

```
                                                    d="M19 7l-.867 12.142A2 2 0 0116.138
21H7.862a2 2 0 01-1.995-1.858L5 7m5 4v6m4-6v6m1-10V4a1 1 0 00-1-1h-4a1 1 0 00-1 1v3M4 7h16"
/>
                                                </svg>
                                            </div>
                                            <span class="text-xs font-medium">Delete</span>
                                        </button>
                                    </form>
                                </div>
                            </div>
                        </div>
                    @endforeach
                </div>
            </div>
        </div>
@endsection
```

6 : Affichez les données (l’action show).

Nous devons écrire la fonction index du BookController pour renvoyer une vue d'index avec des données extraites d'une base de données. Écrivez le code suivant dans la fonction show().

```php
    public function show($id)
    {
        //
        $book = Book::findOrFail($id);
        return view('book.show',compact('book'));
    }
```

De même, créer les vues correspondantes pour afficher et modifier un livre.