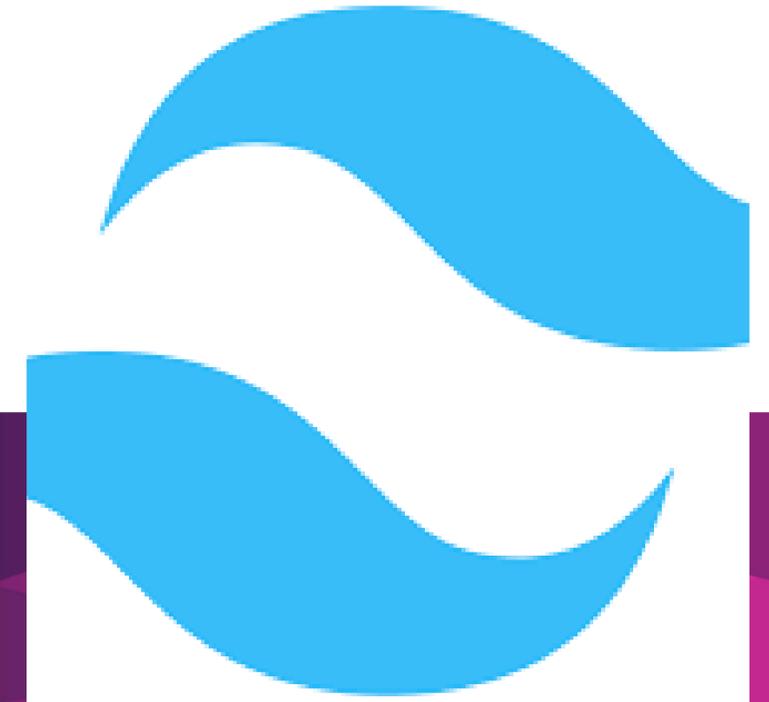


● Marvyn CLOUET

MyTooDooList



Technologies utilisées



Préparation du projet :

```
laravel new nom-de-votre-projet
```

```
composer require laravel/ui
```

```
php artisan ui vue --auth
```

```
npm install && npm run dev
```

```
php artisan key:generate
```

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=8889
DB_DATABASE=projetoodoo
DB_USERNAME=root
DB_PASSWORD=root
```

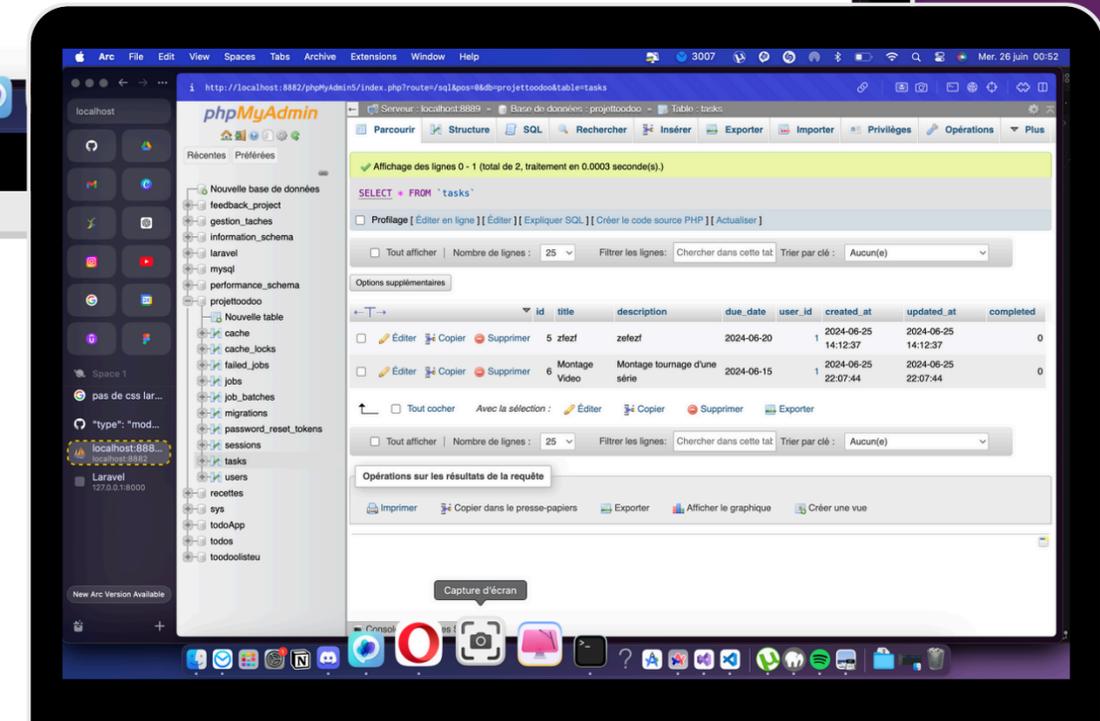
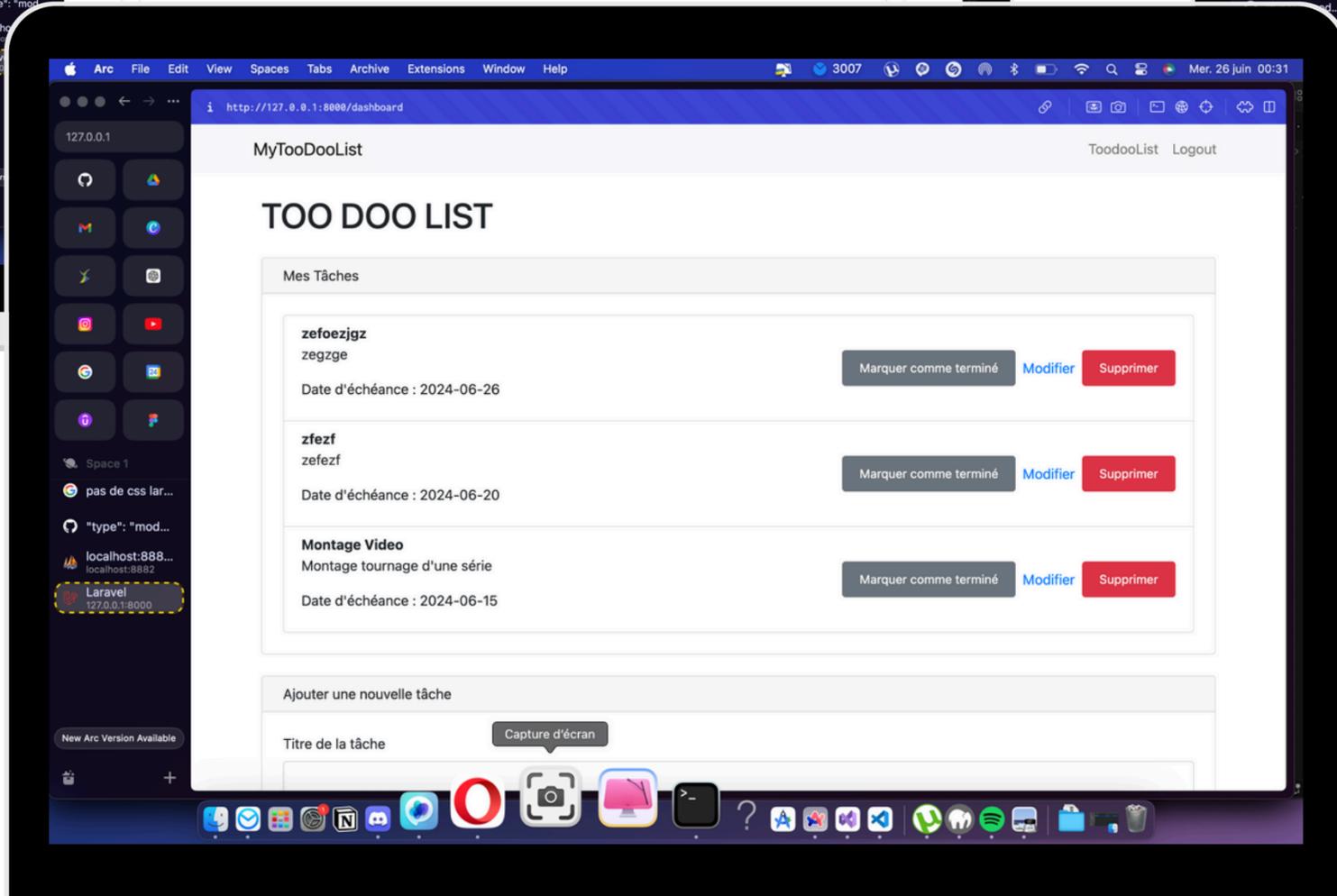
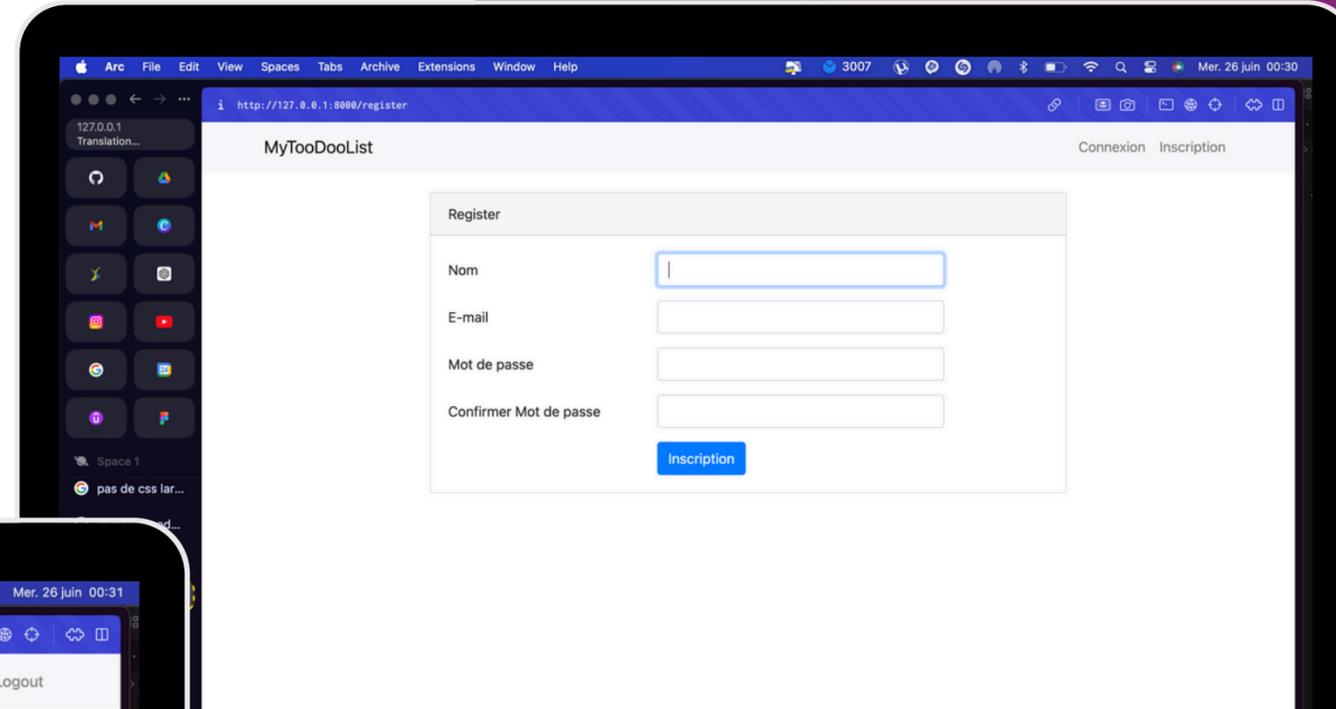
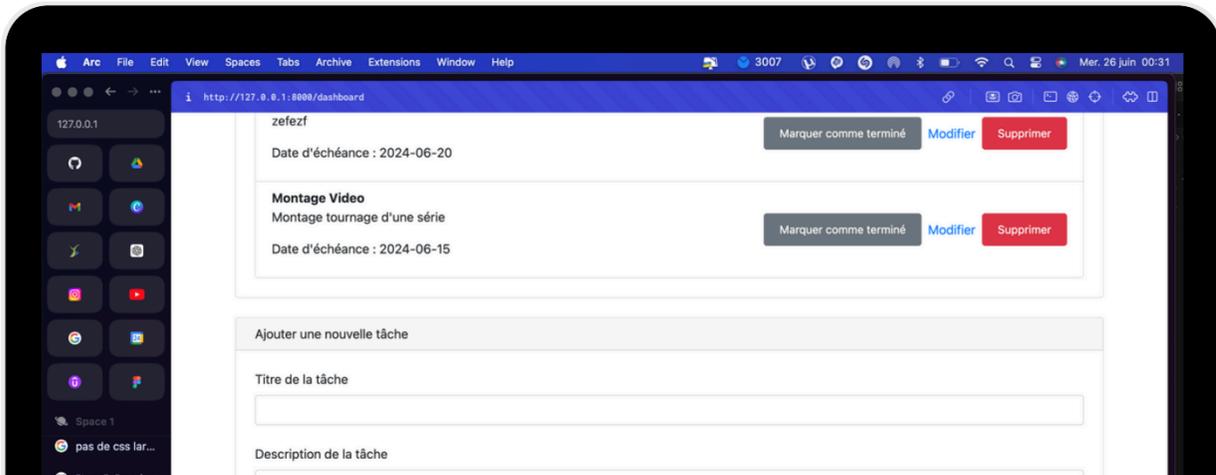
```
php artisan migrate
```

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

```
php artisan make:controller TaskController
```

```
2024_06_25_120855_create_tasks_table.php x TaskController.php 2024_
database > migrations > 2024_06_25_120855_create_tasks_table.php > ...
You, il y a 46 minutes | 1 author (You)
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 You, il y a 46 minutes | 1 author (You)
8 return new class extends Migration
9 {
10     /**
11      * Run the migrations.
12      */
13     public function up(): void
14     {
15         Schema::create('tasks', function (Blueprint $table) {
16             $table->id();
17             $table->string('title');
18             $table->text('description')->nullable();
19             $table->date('due_date');
20             $table->unsignedBigInteger('user_id');
21             $table->timestamps();
22
23             // Foreign key constraint
24             $table->foreign('user_id')->references('id')->on('users')->
25             >onDelete('cascade');
26         });
27     }
28     /**
29      * Reverse the migrations.
30      */
31     public function down(): void
32     {
33         Schema::dropIfExists('tasks');
34     }
35 };
ran
```

MyTooDooList Laravel



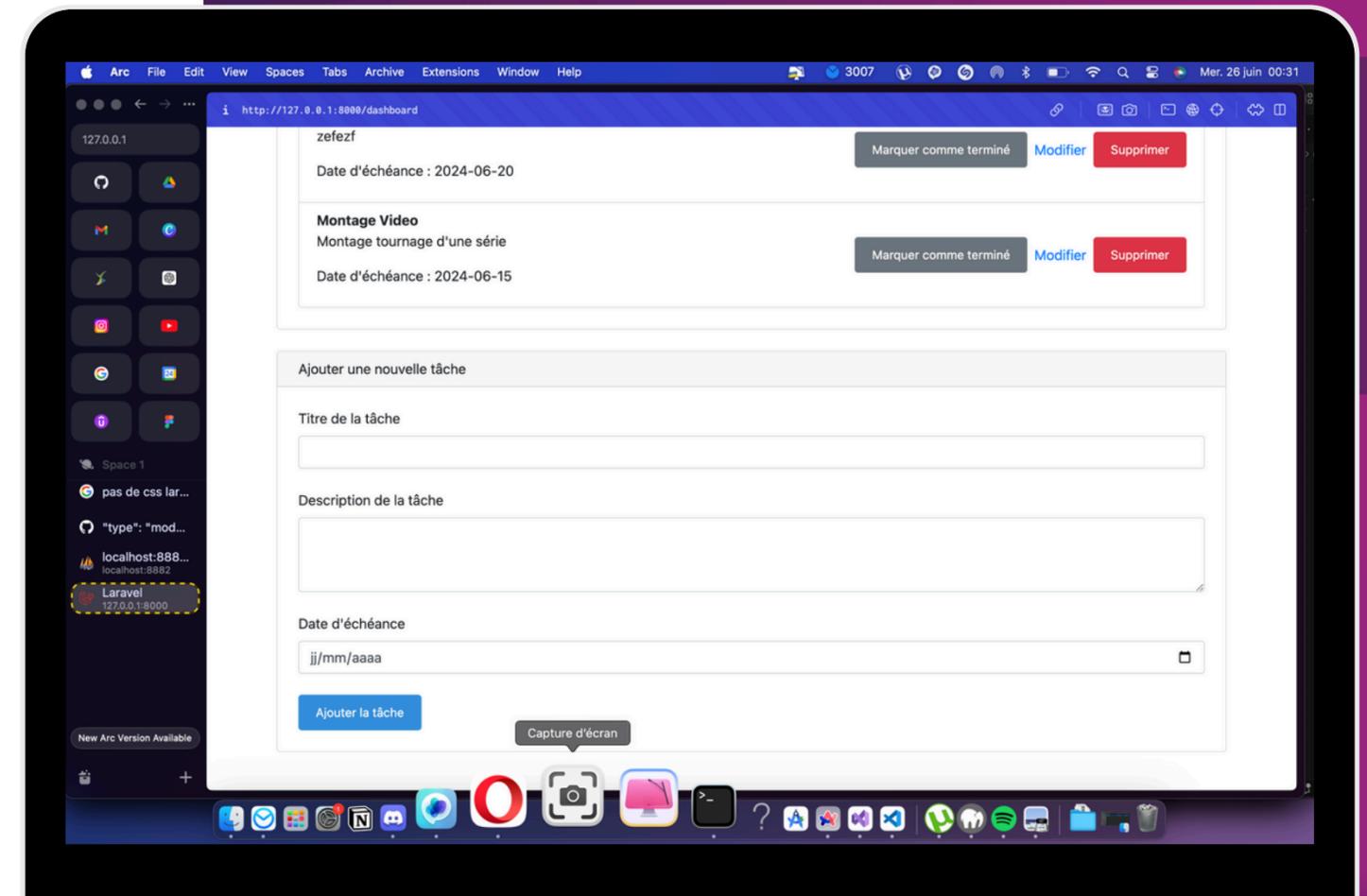
Création de tâches :

```
public function store(Request $request)
{
    // Validation des données reçues du formulaire
    $validatedData = $request->validate([
        'title' => 'required|string|max:255',
        'description' => 'nullable|string',
        'due_date' => 'required|date',
    ]);

    // Création d'une nouvelle tâche pour l'utilisateur connecté
    $task = new Task();
    $task->title = $validatedData['title'];
    $task->description = $validatedData['description'];
    $task->due_date = $validatedData['due_date'];
    $task->user_id = auth()->id(); // Assigner l'ID de l'utilisateur connecté
    $task->completed = false; // Par défaut, la tâche n'est pas complétée

    $task->save();

    // Redirection vers le tableau de bord avec un message de succès
    return redirect()->route('dashboard')->with('success', 'La tâche a été ajoutée avec succès.');
```



Modification de tâches :

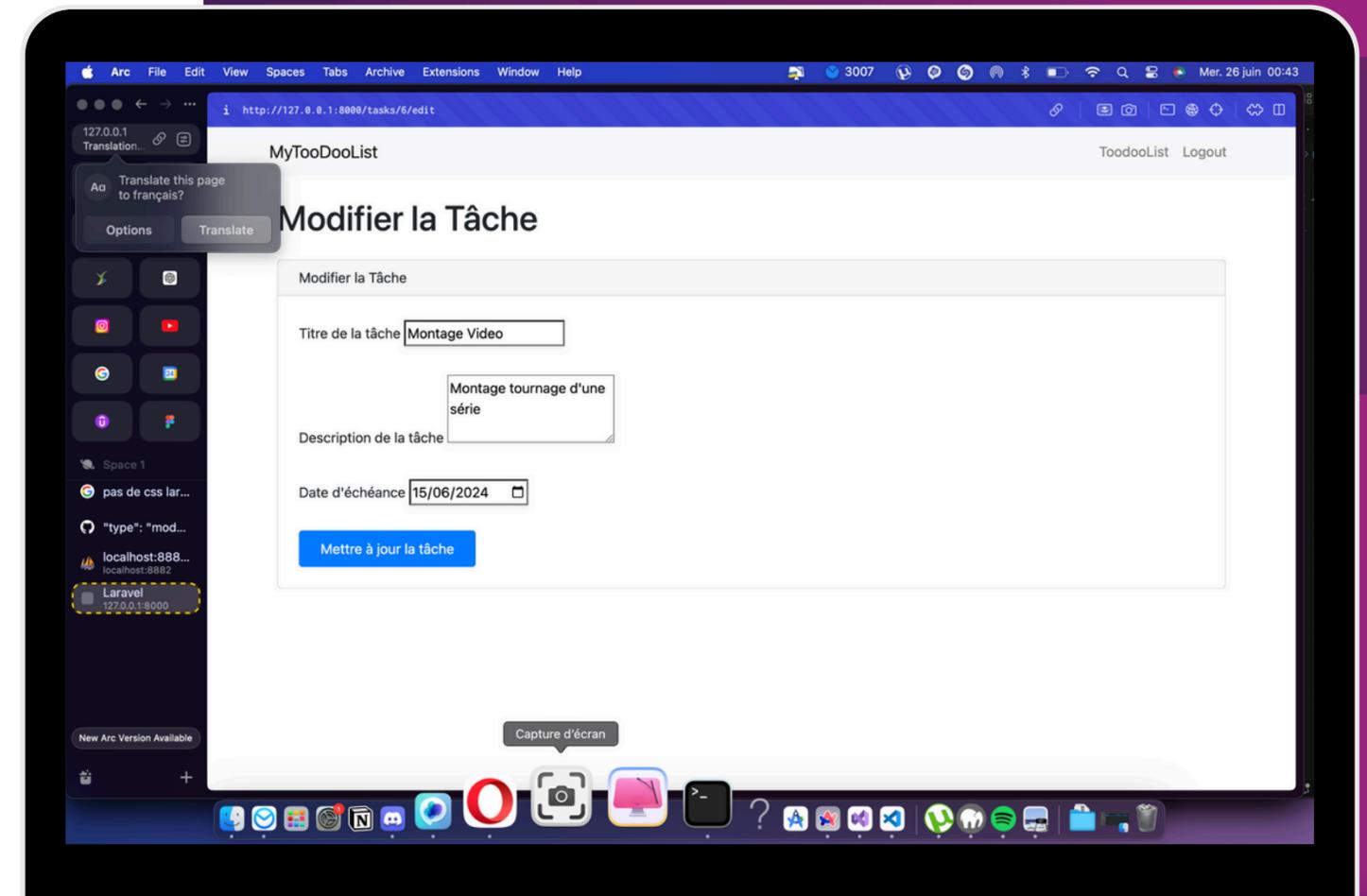
```
// Mise à jour des détails de la tâche
$task->update($validatedData);

// Redirection vers le tableau de bord avec un message de succès
return redirect()->route('dashboard')->with('success', 'La tâche a été mise à jour avec succès.');
```

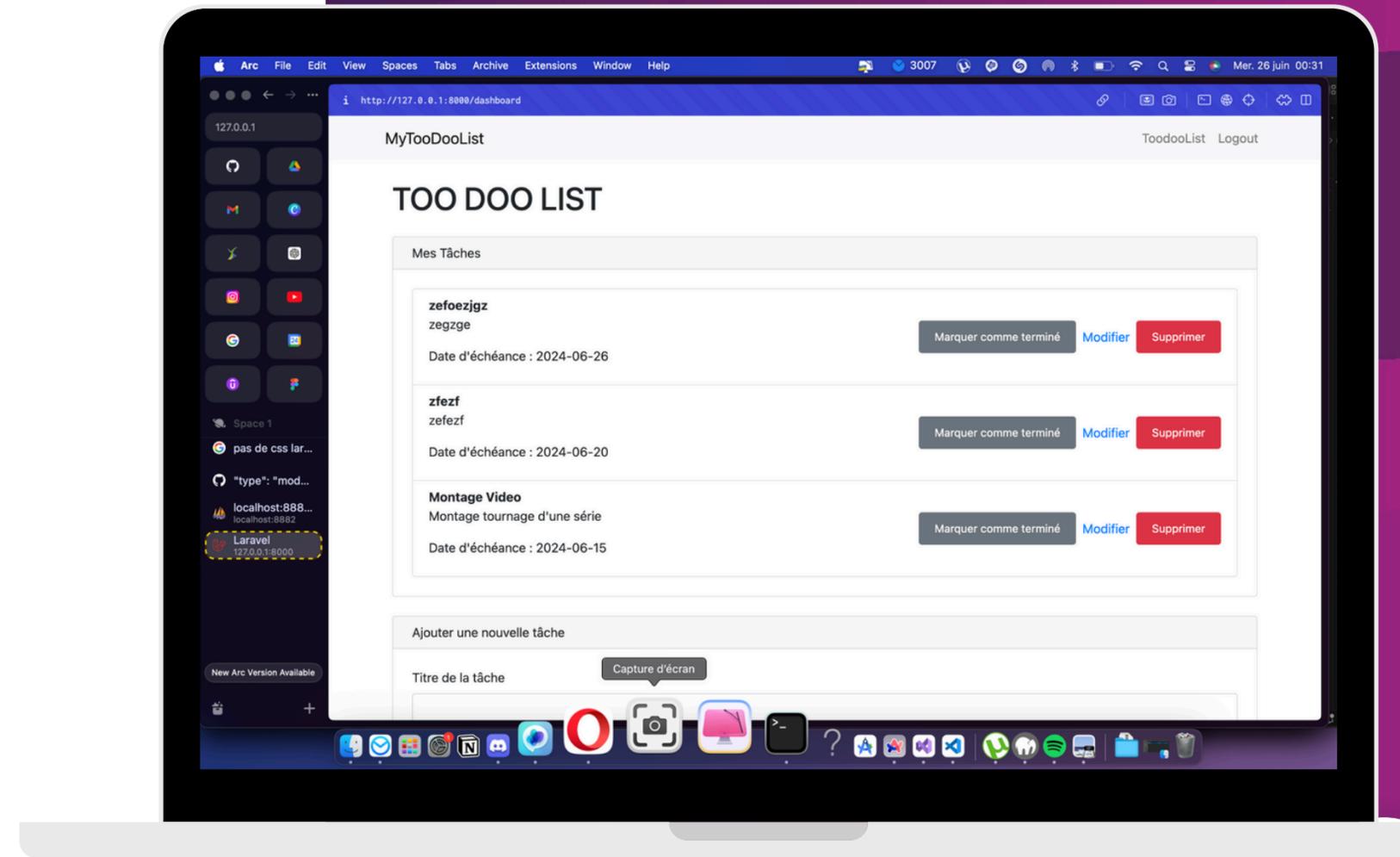
```
public function toggleComplete(Task $task)
{
    // Vérifier si l'utilisateur connecté est le propriétaire de la tâche
    if ($task->user_id !== auth()->id()) {
        return redirect()->route('dashboard')->with('error', 'Vous n'êtes pas autorisé à modifier cette tâche.');
```

```
    // Changer l'état de la tâche (complétée/non complétée)
    $task->completed = !$task->completed;
    $task->save();

    // Redirection vers le tableau de bord avec un message de succès
    return redirect()->route('dashboard')->with('success', 'La tâche a été mise à jour avec succès.');
```



Listes des tâches :

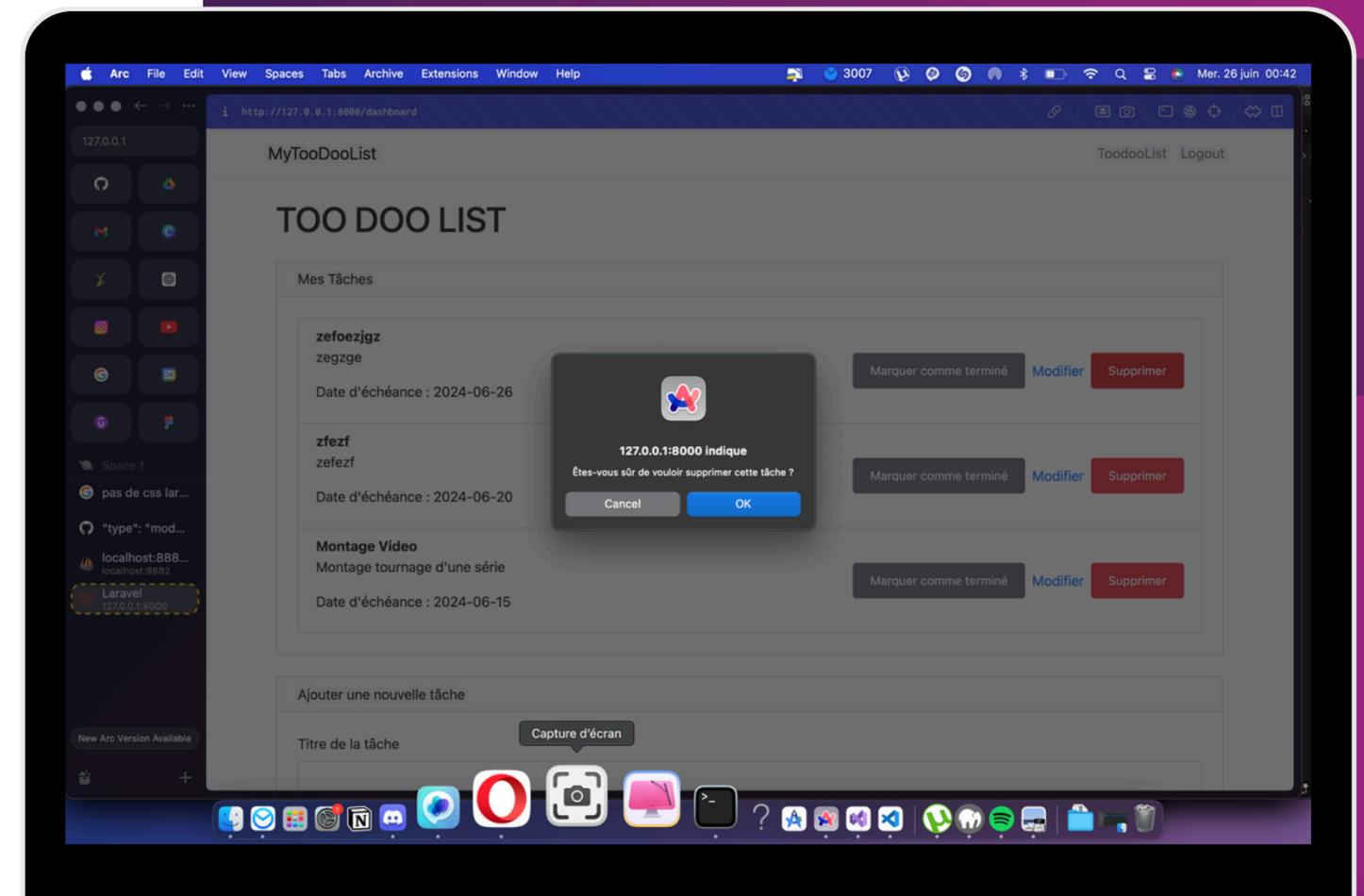


Suppression des tâches :

```
// Vérifier si l'utilisateur connecté est le propriétaire de la tâche
if ($task->user_id !== auth()->id()) {
    return redirect()->route('dashboard')->with('error', 'Vous n\'êtes pas autorisé à supprimer cette tâche.');
```

```
// Supprimer la tâche
$task->delete();

// Redirection vers le tableau de bord avec un message de succès
return redirect()->route('dashboard')->with('success', 'La tâche a été supprimée avec succès.');
```



Base de Données :

