

## Models

### class Etudiant extends Model{

```
use HasFactory;
protected $fillable = ['nom', 'email', 'photo', 'cv',
'datenaissance', 'filiere'];
public $timestamps = false;
}
```

## Les migrations

### public function up(){

```
Schema::create('etudiants', function(Blueprint $table){
    $table->id();
    $table->string('nom');
    $table->string('email');
    $table->string('photo');
    $table->string('cv');
    $table->date('datenaissance');
    $table->string('filiere');
    $table->foreign('filiere')->references('codefiliere')-
>on('filiere')->onDelete('cascade');
    $table->timestamps();
});
```

```
// Reverse the migrations.//
```

### public function down(): void{

```
Schema::dropIfExists('etudiants');
}
```

## Seeder

```
class EtudiantsTableDataSeeder extends Seeder{
```

```
/* Run the database seeds. */
```

### public function run(): void{

```
Etudiant::create([
    'id'=>'1',
    'nom'=>'Allaoui',
    'prenom'=>'Sarah',
    'email'=>'allaoui@gmail.com',
    'photo'=>'images/img1.jpg',
    'cv'=>'docs/cv1.pdf',
    'ville'=>'Casa',
    'datenaissance'=>'2000-12-22',
    'filiere'=>'MIAM',
]);
}
```

## Les controllers API:

```
class EtudiantControler extends Controller{
```

### public function index(){

```
$categories = Category::all();
return response()->json($categories);
}
```

### public function store(Request \$request){

```
$validator = Validator::make($request->all(), [
    'code' => 'required',
    'libelle' => 'required',
]);
if ($validator->fails()) {
    return response()->json(['error' => $validator->errors()],
400);
}
$cat = new Category;
$cat->code = $request->input('code');
$cat->libelle = $request->input('libelle');
$cat->description = $request->input('description');
//telecharger img//
if ($request->hasFile('image')) {
    $image = $request->file('image');
    $path = Storage::disk('public')->put('categories', $image);
    $cat->image = $path;
    $cat->save();
    return response()->json(['message' => 'Catégorie bien
créée!']);
}
```

### public function show(\$id){

```
$cat = Category::findOrFail($id);
return response()->json($cat);
}
```

### public function update(Request \$request, \$id){

```
$validator = Validator::make($request->all(), [
    'code' => 'required',
    'libelle' => 'required',
]);
if ($validator->fails()) {
    return response()->json(['error' => $validator->errors()],
400);
}
$cat = Category::findOrFail($id);
$cat->code = $request->input('code');
$cat->libelle = $request->input('libelle');
$cat->description = $request->input('description');
$cat->save()
```

```
return response()->json(['message' => 'Catégorie bien
modifiée']);
}
```

## Laravel

### public function destroy(\$id){

```
$cat = Category::find($id);
if($cat){
    $cat->delete();
    return response()->json(['message' => 'Catégorie bien
supprimée']);
}else{
    return response()->json(['error' => 'Catégorie non
trouvée'], 404);
}
}
```

## Authentification

```
class AuthController extends Controller{
```

### public function register(Request \$request){

```
$validator = Validator::make($request->all(), [
    'nom' => 'required',
    'email' => 'required',
    'password' => 'required',
    'c_password' => 'required | same:password',
]);
if($validator->fails()){
    return response()->json(['error' => $validator -> error()],
401);
}else{
    $input = $request->all();
    $input['password'] = bcrypt($input['password']);
    $user = User::create($input);
    $success['token'] = $user->createToken('AppName')-
>accessToken;
    return response()->json(['success' => $success], 200);
}
```

### public function login(){

```
if(Auth::attempt(['email' => request('email'), 'password' =>
request('password')])){
    $user = Auth::user();
    $success['token'] = $user->createToken('AppName')-
>accessToken;
    return response()->json(['success' => $success], 200);
}else{
    return response()->json(['error' => 'Unauthorised'], 401);
}
}
```

### public function getUser(){

```
$user = Auth::user();
return response()->json(['success' => $user], 200);
}
```

```
▪ Route::get('/produits', [ProduitController::class, 'index']
```

```
->name('produit.index');
```

```
▪ Route::get('/produits/create', [ProduitController::class,
```

```
'create']->name('produit.create');
```

```
▪ Route::post('/produits', [ProduitController::class, 'store']
```

```
->name('produit.store');
```

```
▪ Route::get('/produits/{produit}/edit',
```

```
[ProduitController::class, 'edit']->name('produit.edit');
```

```
▪ Route::put('/produits/{produit}', [ProduitController::class,
```

```
'update']->name('produit.update');
```

```
▪ Route::delete('/produits/{produit}', [ProduitController::class,
```

```
'destroy']->name('produit.destroy');
```

```
voire tout ca dans une route
```

```
Route::resources('/test,nom_controller::class
```

```
middleware
```

```
Route::group(['middleware' => 'Auth:api'], function(){
```

```
Route::get('/getUser', [AuthController::class, 'getUser'];
```

```
});
```

```
Route::post('/login', [AuthController::class, 'login'];
```

```
Route::post('/register', [AuthController::class, 'register'];
```

## Middleware

### public function handle(\$request, Closure \$next){

```
if (!Auth::check()) {
    return redirect()->route('login');
}
return $next($request);
}
```

## CMD :

```
composer require "laravel/passport:^10.2"
```

```
php artisan migrate
```

```
php artisan make:controller AuthController
```

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

```
php artisan make:migration create_filiere_table
```

```
php artisan make:controller FiliereController
```

```
php artisan migrate:fresh
```

```
php artisan passport:install
```

```
php artisan make:seeder etudiantsTableDataSeeder
```

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

```
php artisan make:middleware AuthMiddleware
```

## View Controllers

### public function index(){

```
$products = Produit::with('categorie')->get();
$categories = Category::all();
return view('produits.index', compact('products',
'categories'));
}
```

### public function create(){

```
$categories = Category::all();
return view('produits.create', compact('categories'));
}
```

### public function store(Request \$request){

```
$request->validate([
    'reference' => 'required',
    'libelle' => 'required',
]);
$produit = new Produit;
$produit->reference = $request->input('reference');
$produit->libelle = $request->input('libelle');
$produit->categorie_id = $request->input('categorie_id');
if ($request->hasFile('image')) {
    $image = $request->file('image');
    $imageName = time().'.'.$image-
>getClientOriginalName();
    $image->move(public_path('images'), $imageName);
    $produit->image = $imageName;
}
$produit->save();
return redirect()->route('produits.index')-
>with('success', 'Le produit bien créé!');
```

### public function show(\$id){

```
$product = Produit::findOrFail($id);
return view('produits.show', compact('product'));
}
```

### public function edit(\$id){

```
$product = Produit::findOrFail($id);
$categories = Category::all();
return view('produits.edit', compact('product',
'categories'));
}
```

### public function update(Request \$request, \$id){

```
$request->validate([
    'reference' => 'required',
    'libelle' => 'required',
]);
$produit = Produit::findOrFail($id);
$produit->reference = $request->input('reference');
$produit->libelle = $request->input('libelle');
$produit->description = $request->input('description');
$produit->categorie_id = $request->input('categorie_id');
if ($request->hasFile('image')) {
    if ($produit->image) {
        $imagePath = public_path('images/' . $produit->image);
        if (file_exists($imagePath)) {
            unlink($imagePath);
        }
    }
    $image = $request->file('image');
    $imageName = time() . '_' . $image-
>getClientOriginalName();
    $image->move(public_path('images'), $imageName);
    $produit->image = $imageName;
}
$produit->save();
return redirect()->route('produits.index')->with('success',
'Product updated successfully.');
```

### public function destroy(\$id){

```
$produit = Produit::findOrFail($id);
// Supprimer l'image si elle existe
if ($produit->image) {
    $imagePath = public_path('images/' . $produit->image);
    if (file_exists($imagePath)) {
        unlink($imagePath);
    }
}
$produit->delete();
return redirect()->route('produits.index')->with('success',
'Product deleted successfully.');
```

### //Layouts Views

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Gestion des Catégories</title>
</head>
<body>
  <nav>
    <a href="{{ route('categories.index') }}">Gestion des
Catégories</a>
    <button type="button" data-toggle="collapse"
      data-target="#navbarNav" aria-controls="navbarNav"
      aria-expanded="false" aria-label="Toggle navigation">
      <span></span>
    </button>
    <div id="navbarNav">
      <ul>
        <li>
          <a href="{{ route('categories.index') }}">Categories</a>
        </li>
      </ul>
    </div>
  </nav>

  <div class="container mt-4">
    @yield('content')
  </div>
  <script src="{{ asset('js/app.js') }}"></script>
</body>
</html>
```

### //index views//

```
@extends('layouts.app')
@section('content')
<div >
  <h1>Liste des produits</h1>
  <div>
    <button type="button" data-toggle="modal" data-
target="#addProductModal">
      Nouveau Produit
    </button>
  </div>
  <table>
    <thead>
      <tr>
        <th>Image</th>
        <th>Référence</th>
        <th>Libellé</th>
        <th>Action</th>
      </tr>
    </thead>
    <tbody>
      @foreach ($products as $product)
        <tr>
          <td></td>
          <td>{{ $product->reference }}</td>
          <td>{{ $product->libelle }}</td>
          <td>
            <button type="button" data-toggle="modal" data-
target="#editProductModal{{ $product->id }}"> Editer
          </button>
            <button type="button" data-toggle="modal" data-
target="#deleteProductModal{{ $product->id }}"> Supprimer
          </button>
        </td>
        </tr>
      @include('produits.modal.edit', ['product' =>
$product])
      @include('produits.modal.delete', ['product' =>
$product])
    @endforeach
  </tbody>
</table>
</div>
@include('produits.modal.add')
@endsection
```

### //create views//

```
@extends('layouts.app')
@section('content')
<div>
  <h1>Nouvelle Catégorie</h1>
  <form action="{{ route('categories.store') }}"
method="POST">
    @csrf
    <div>
      <label for="code">Code:</label>
      <input type="text" name="code" required>
    </div>
```

```
<div>
  <label for="description">Description:</label>
  <textarea name="description"></textarea>
</div>
<button type="submit">Enregistrer</button>
</form>
</div>
@endsection
```

---

### //edit views//

```
@extends('layouts.app')
@section('content')
<div>
  <h1>Editer une catégorie</h1>
  <form action="{{ route('categories.update', $cat->id) }}"
method="POST">
    @csrf
    @method('PUT')
    <div>
      <label for="code">Code:</label>
      <input type="text" name="code" value="{{ $cat->code
}}" required>
    </div>
    <div>
      <label for="description">Description:</label>
      <textarea name="description">{{ $cat->description
}}</textarea>
    </div>
    <button type="submit">Energister</button>
  </form>
</div>
@endsection
```

### //show views//

```
@extends('layouts.app')
@section('content')
<div>
  <h1>Détails</h1>
  <table>
    <tbody>
      <tr>
        <th>ID:</th>
        <td>{{ $cat->id }}</td>
      </tr>
      <tr>
        <th>Code:</th>
        <td>{{ $cat->code }}</td>
      </tr>
      <tr>
        <th>Description:</th>
        <td>{{ $cat->description }}</td>
      </tr>
    </tbody>
  </table>
  <a href="{{ route('categories.index') }}">Retour</a>
</div>
@endsection
```