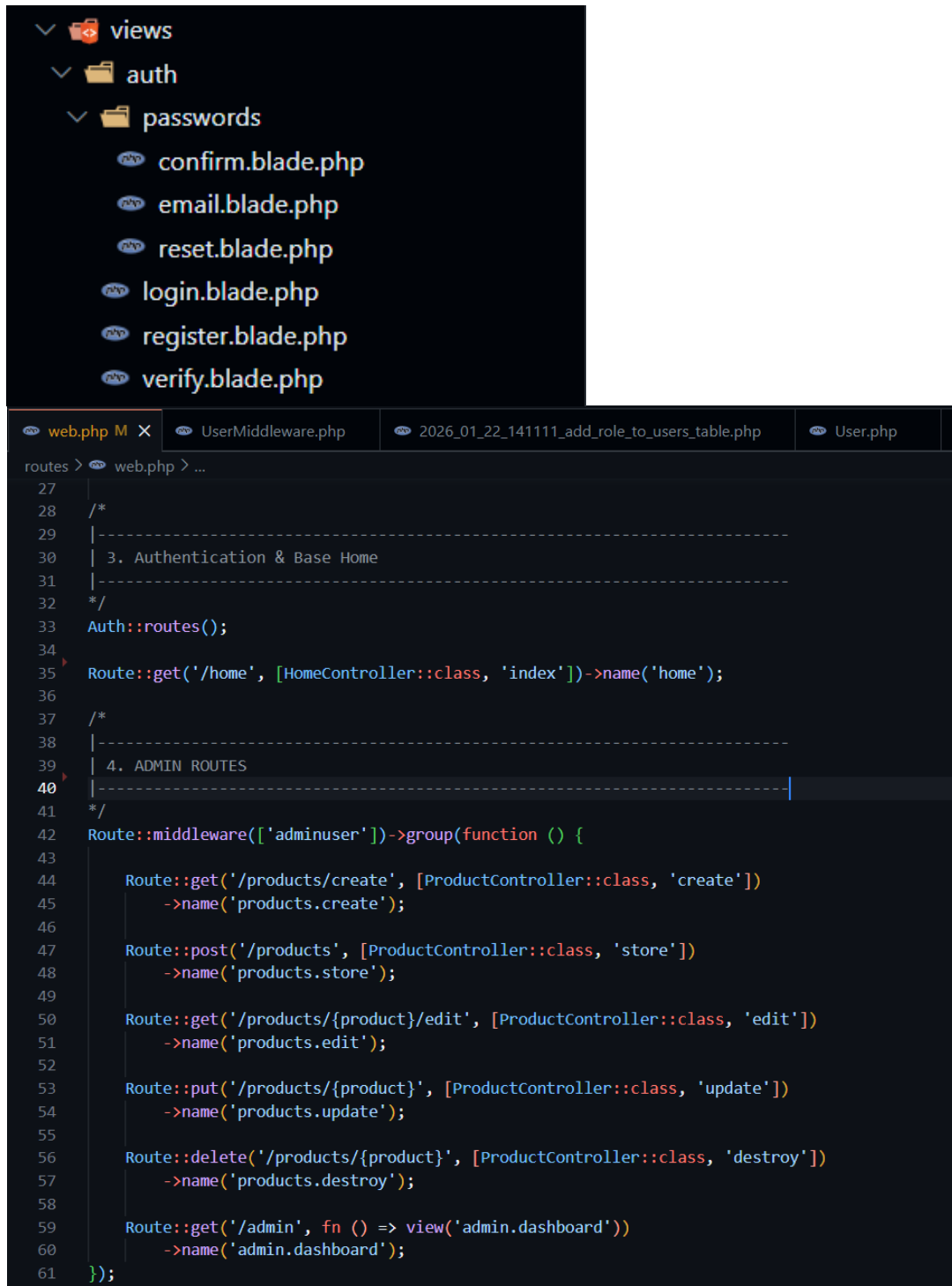


# Atelier 11 : Sécurisation de l'accès aux routes par rôles

## 1- Les vues et routes d'authentification



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the following structure:

- views
  - auth
    - passwords
      - confirm.blade.php
      - email.blade.php
      - reset.blade.php
      - login.blade.php
      - register.blade.php
      - verify.blade.php

The code editor shows the following code in the `routes.php` file:

```
27 |
28 | /*
29 | |-----
30 | | 3. Authentication & Base Home
31 | |-----
32 | */
33 | Auth::routes();
34 |
35 | Route::get('/home', [HomeController::class, 'index'])->name('home');
36 |
37 | /*
38 | |-----
39 | | 4. ADMIN ROUTES
40 | |-----
41 | */
42 | Route::middleware(['adminuser'])->group(function () {
43 |
44 |     Route::get('/products/create', [ProductController::class, 'create'])
45 |         ->name('products.create');
46 |
47 |     Route::post('/products', [ProductController::class, 'store'])
48 |         ->name('products.store');
49 |
50 |     Route::get('/products/{product}/edit', [ProductController::class, 'edit'])
51 |         ->name('products.edit');
52 |
53 |     Route::put('/products/{product}', [ProductController::class, 'update'])
54 |         ->name('products.update');
55 |
56 |     Route::delete('/products/{product}', [ProductController::class, 'destroy'])
57 |         ->name('products.destroy');
58 |
59 |     Route::get('/admin', fn () => view('admin.dashboard'))
60 |         ->name('admin.dashboard');
61 | });
```

```
web.php M X UserMiddleware.php 2026_01_22_141111_add_role_to_users_table.php User.php
routes > web.php > ...
62
63  /*
64  |-----
65  | 5. USER ROUTES
66  |-----
67  */
68  Route::middleware(['useruser'])->group(function () {
69      Route::get('/client', [ProductController::class, 'client'])
70          ->name('client.space');
71  });
72
73  /*
74  |-----
75  | 6. Product Details (Wildcard)
76  |-----
77  */
78  Route::get('/products/{product}', [ProductController::class, 'show'])
79      ->middleware('auth')
80      ->name('products.show');
```

## 2- Ajouter la gestion des rôles

```
web.php M UserMiddleware.php 2026_01_22_141111_add_role_to_users_table.php X
database > migrations > 2026_01_22_141111_add_role_to_users_table.php > class > down > Close
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12      public function up(): void
13      {
14          Schema::table('users', function (Blueprint $table) {
15              $table->string('role')->default('USER')->after('password');
16          });
17      }
18
19      /**
20       * Reverse the migrations.
21       */
22      public function down(): void
23      {
24          Schema::table('users', function (Blueprint $table) {
25              $table->dropColumn('role');
26          });
27      }
28  };
29
```

### 3- Déclarer les rôles dans le modèle User

```
User.php M X AddProductRequest.php filesystems.php vite.config.js 2026.01
app > Models > User.php > User
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Foundation\Auth\User as Authenticatable;
7  use Illuminate\Notifications\Notifiable;
8
9  class User extends Authenticatable
10 {
11     use HasFactory, Notifiable;
12
13     /**
14      * Roles constants
15      */
16     public const USER_ROLE = 'USER';
17     public const ADMIN_ROLE = 'ADMIN';
18
19     /**
20      * The attributes that are mass assignable.
21      *
22      * @var array<int, string>
23      */
24     protected $fillable = [
25         'name',
26         'email',
27         'password',
28         'role',
29     ];
30
```

### 4- Code expliqué du menu

```
9  <nav class="nav-links">
10  {{{-- AUTHENTICATED USERS --}}
11  @auth
12
13      {{{-- ADMIN ONLY --}}
14      @if(auth()->user()->role === 'admin')
15      <a href="/">Home</a>
16      <a href="/electronics">Electronics</a>
17      <a href="{{ route('products.create') }}">Add Product</a>
18      @endif
19
20      @if(auth()->user()->role === 'user')
21      <a href="/">Home</a>
22      <a href="/electronics">Electronics</a>
23      <a href="/contact">Contact</a>
24      @endif
25
26      {{{-- LOGOUT --}}
27      <form action="{{ route('logout') }}" method="POST" style="display:inline;">
28          @csrf
29          <button type="submit" class="msone-nav-logout">
30              Logout
31          </button>
32      </form>
33
34      {{{-- GUEST USERS --}}
35      @else
36      <a href="/">Home</a>
37      <a href="/electronics">Electronics</a>
38      <a href="{{ route('login') }}">Login</a>
39      <a href="{{ route('register') }}">Register</a>
40      @endauth
41  </nav>
```

## 5- Sécuriser les routes avec les middlewares

```
AdminUserMiddleware.php X web.php UserMiddleware.php 2026_01_22_141111_add_role_to
app > Http > Middleware > AdminUserMiddleware.php > AdminUserMiddleware > handle
1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Closure;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Auth;
8  use Symfony\Component\HttpFoundation\Response;
9
10 class AdminUserMiddleware
11 {
12     public function handle(Request $request, Closure $next): Response
13     {
14         $user = Auth::user();
15
16         // Not logged in
17         if (!$user) {
18             return redirect()->route('login');
19         }
20
21         // Not admin
22         if ($user->role !== 'admin') {
23             abort(403, 'Unauthorized');
24         }
25
26         return $next($request);
27     }
28 }
29
```

```
UserMiddleware.php X 2026_01_22_141111_add_role_to_users_table.php User.php A
app > Http > Middleware > UserMiddleware.php > UserMiddleware > handle
1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Closure;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Auth;
8  use Symfony\Component\HttpFoundation\Response;
9
10 class UserMiddleware
11 {
12     public function handle(Request $request, Closure $next): Response
13     {
14         $user = Auth::user();
15
16         // Not logged in
17         if (!$user) {
18             return redirect()->route('login');
19         }
20
21         // Not normal user
22         if ($user->role !== 'user') {
23             abort(403, 'Unauthorized');
24         }
25
26         return $next($request);
27     }
28 }
29
```