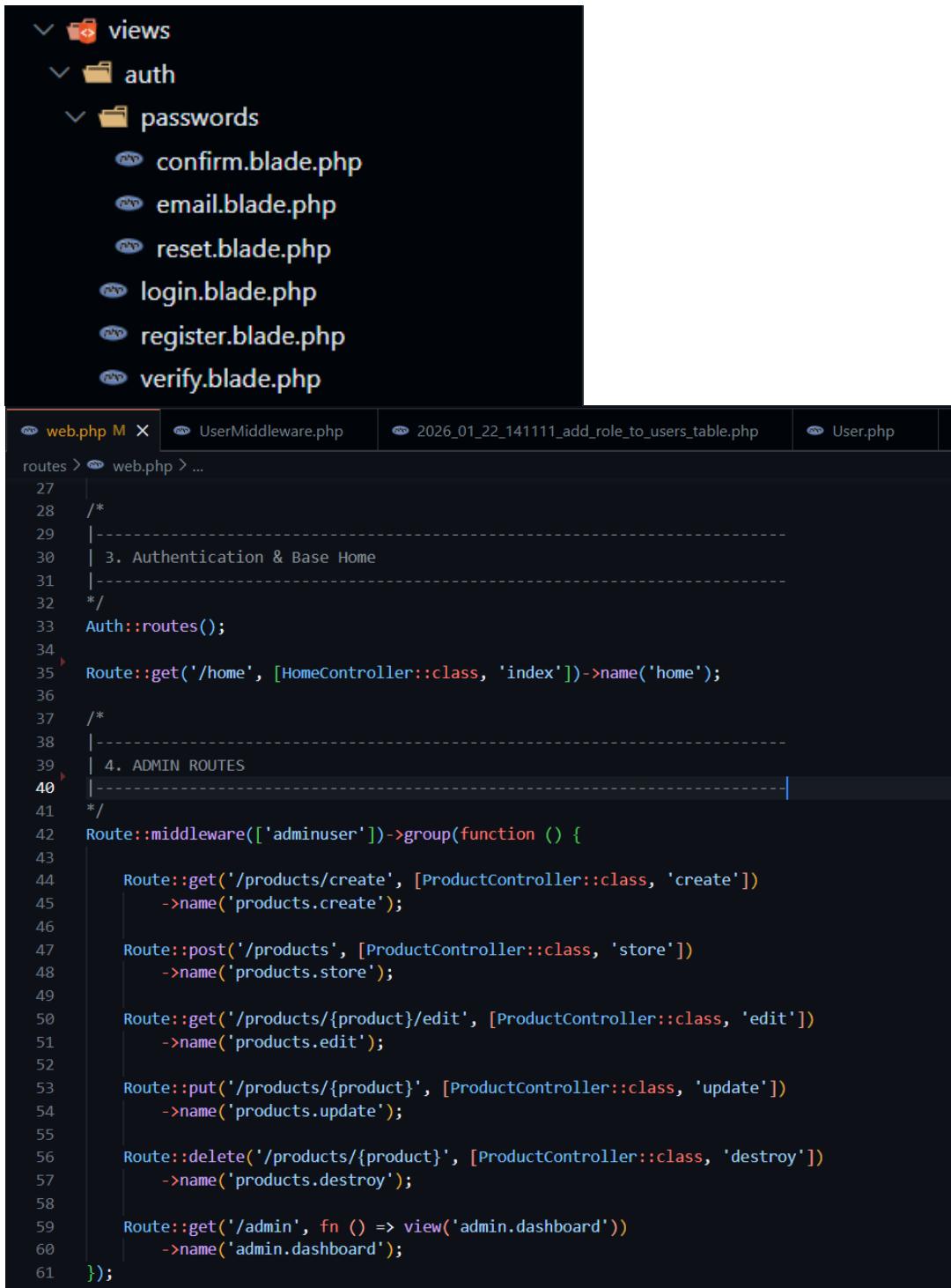


Atelier 11 : Sécurisation de l'accès aux routes par rôles

1- Les vues et routes d'authentification



The screenshot shows a code editor with a file structure on the left and the contents of `web.php` on the right.

File Structure:

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

Code Editor Content (`web.php`):

```
routes > web.php > ...
27  /*
28  |-----
29  | 3. Authentication & Base Home
30  |-----
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 });

  
```

The screenshot shows a code editor with two tabs open: `routes/web.php` and `UserMiddleware.php`. The `routes/web.php` tab is active, displaying the following code:

```
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

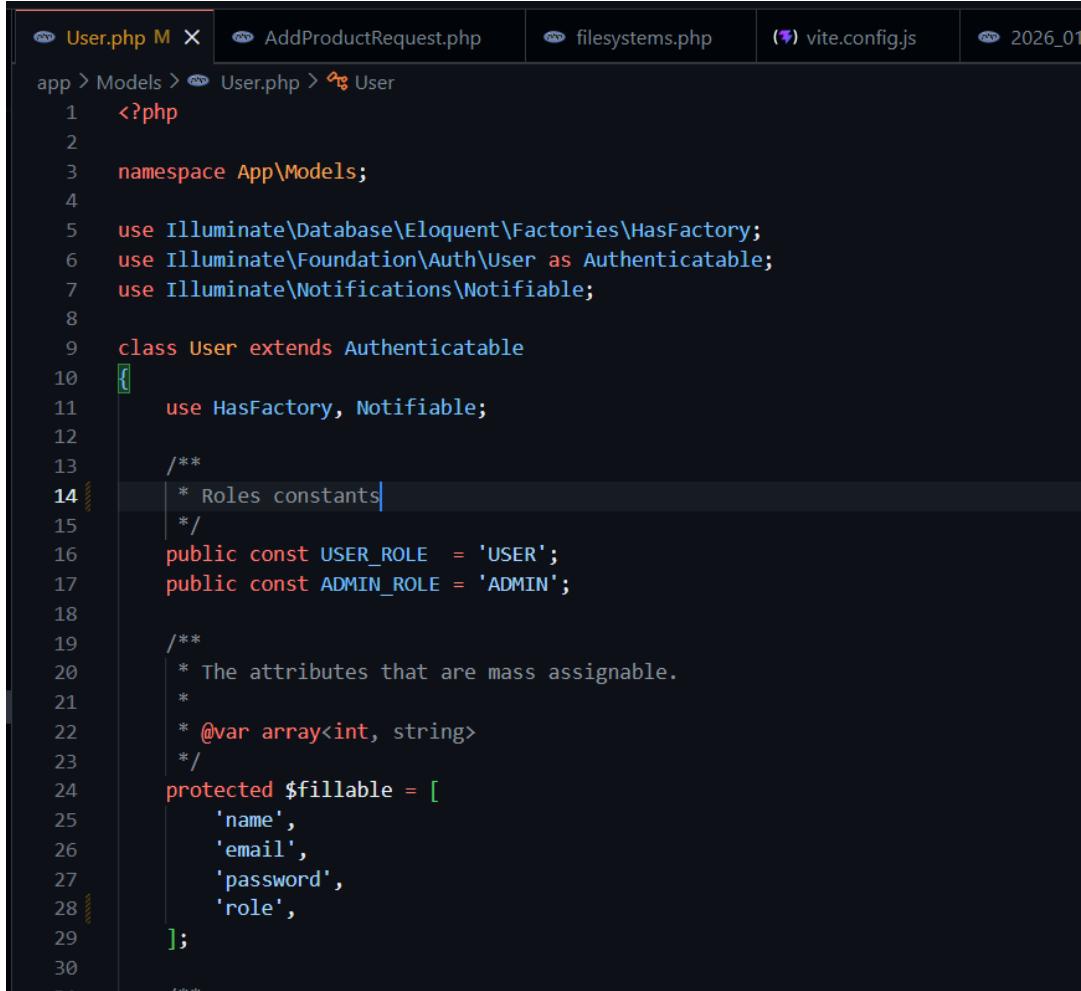
The screenshot shows a code editor with three tabs: `web.php`, `UserMiddleware.php`, and `2026_01_22_141111_add_role_to_users_table.php`. The `2026_01_22_141111_add_role_to_users_table.php` tab is active, displaying the following migration code:

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::table('users', function (Blueprint $table) {
            $table->string('role')->default('USER')->after('password');
        });
    }

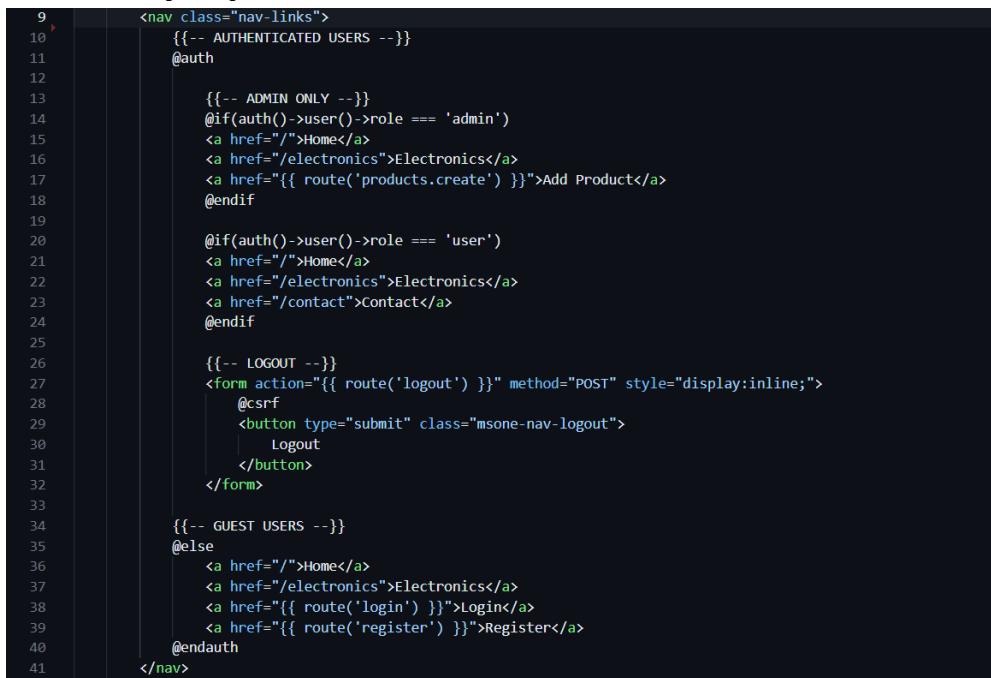
    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::table('users', function (Blueprint $table) {
            $table->dropColumn('role');
        });
    }
};
```

3- Déclarer les rôles dans le modèle 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         @endifauth
41     </nav>
```

5- Sécuriser les routes avec les middlewares

```
AdminUserMiddleware.php X web.php UserMiddleware.php 2026_01_22_141111_add_role_to_users_table.php
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 AdminUserMiddleware.php
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
```