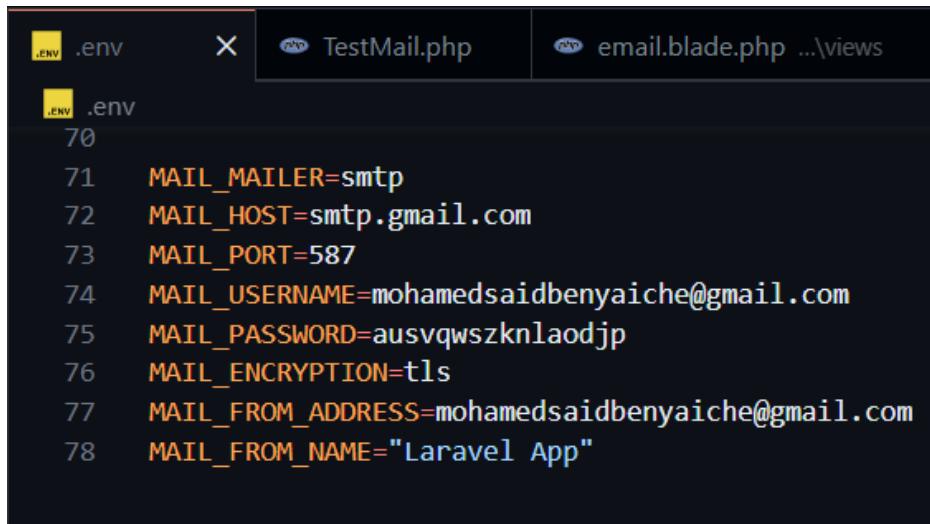


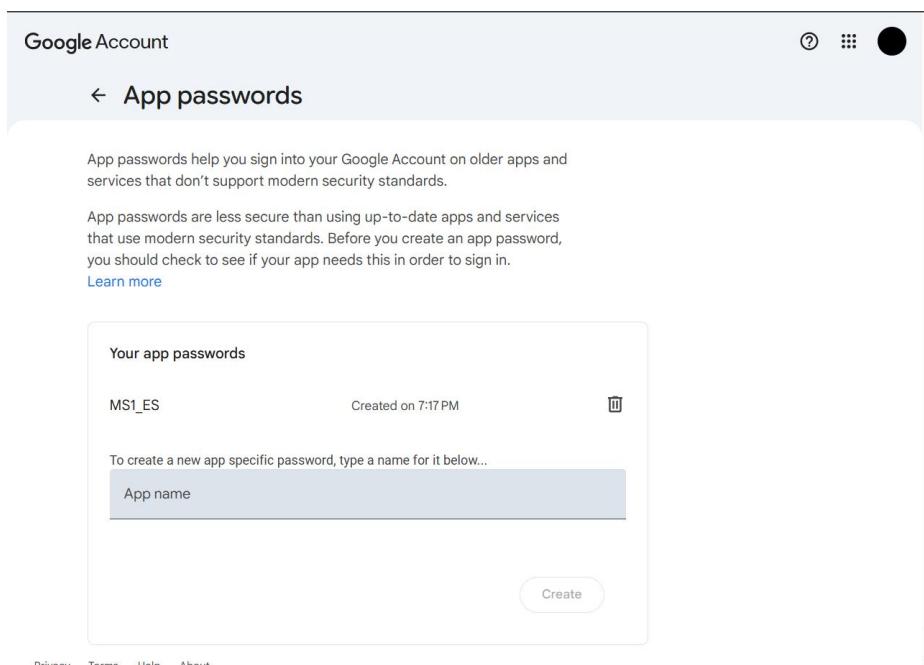
Envoi d'emails avec Gmail dans Laravel 12

Étape 2 — Configuration Gmail dans

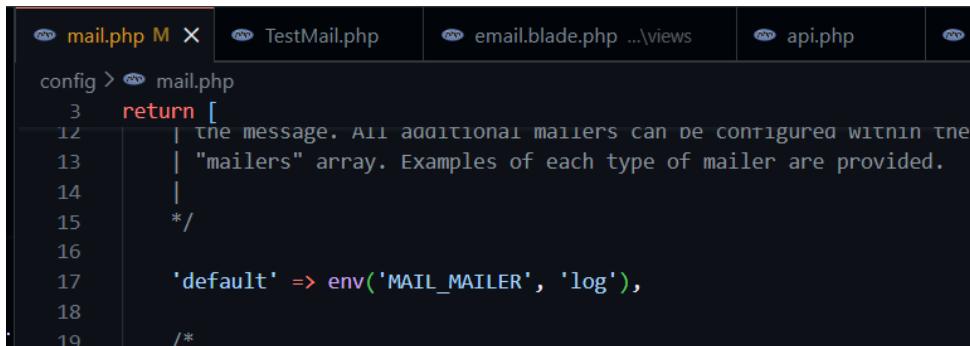


```
.env .env TestMail.php email.blade.php ...\\views
.env .env
70
71 MAIL_MAILER=smtp
72 MAIL_HOST=smtp.gmail.com
73 MAIL_PORT=587
74 MAIL_USERNAME=mohamedsaidbenyaiche@gmail.com
75 MAIL_PASSWORD=ausvqwszknlaodjp
76 MAIL_ENCRYPTION=tls
77 MAIL_FROM_ADDRESS=mohamedsaidbenyaiche@gmail.com
78 MAIL_FROM_NAME="Laravel App"
```

Étape 4 — Créer un mot de passe d'application Gmail



Étape 5 — Vérifier la configuration dans Laravel

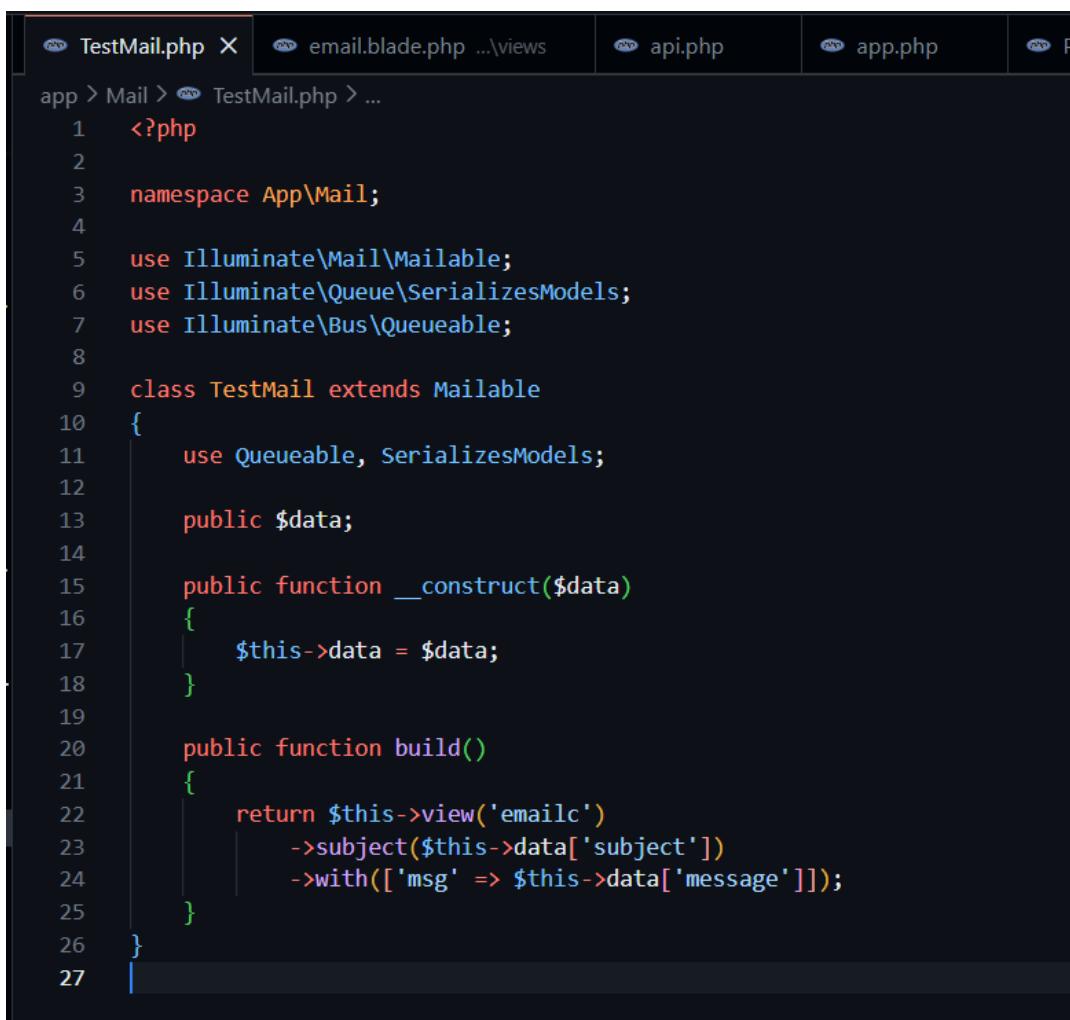


The screenshot shows a code editor with the file `config/mail.php` open. The code defines a `mailers` array with a default value of `'log'`.

```
config > mail.php
  3  return [
 12 |     | the message. All additional mailers can be configured within the
 13 |     | "mailers" array. Examples of each type of mailer are provided.
 14 |
 15 |     */
 16 |
 17 |     'default' => env('MAIL_MAILER', 'log'),
 18 |
 19 | 
```

Envoyer un email (Laravel 12 + Gmail + Formulaire)

1) Le Mailable : TestMail (le modèle d'email)

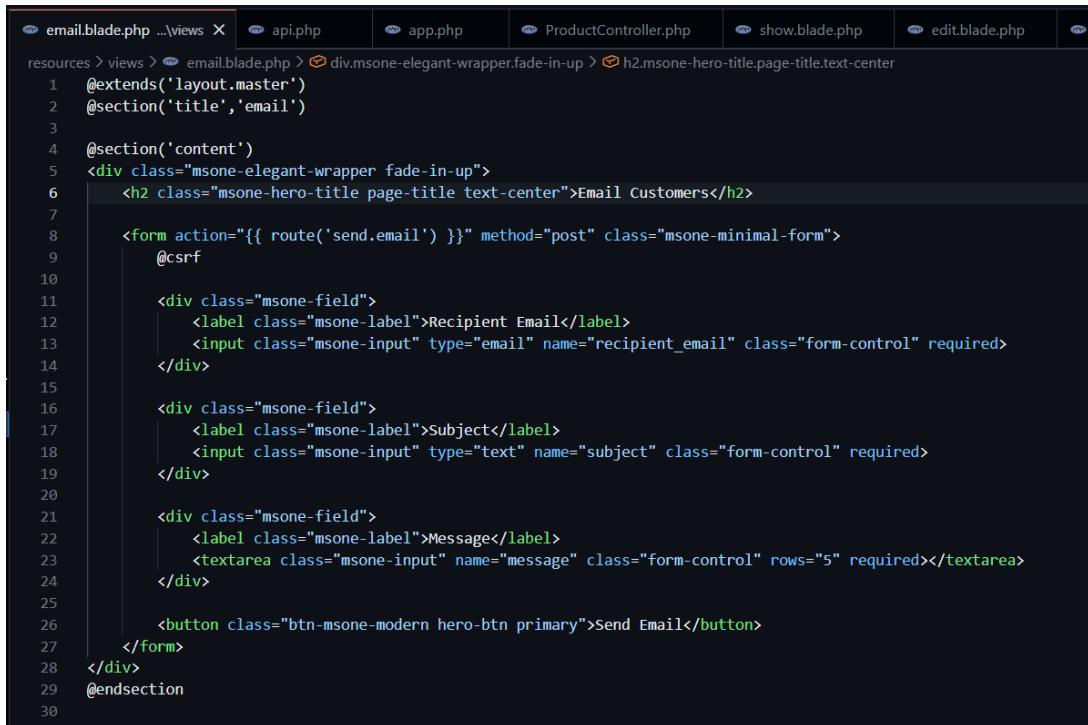


The screenshot shows a code editor with the file `app/Mail/TestMail.php` open. It defines a `TestMail` class that extends `Mailable` and implements `Queueable` and `SerializesModels`. The `__construct` method initializes the `$data` attribute. The `build` method returns a view with subject and message data.

```
app > Mail > TestMail.php > ...
  1  <?php
  2
  3  namespace App\Mail;
  4
  5  use Illuminate\Mail\Mailable;
  6  use Illuminate\Queue\SerializesModels;
  7  use Illuminate\Bus\Queueable;
  8
  9  class TestMail extends Mailable
 10 {
 11     use Queueable, SerializesModels;
 12
 13     public $data;
 14
 15     public function __construct($data)
 16     {
 17         $this->data = $data;
 18     }
 19
 20     public function build()
 21     {
 22         return $this->view('emailc')
 23             ->subject($this->data['subject'])
 24             ->with(['msg' => $this->data['message']]);
 25     }
 26 }
 27 
```

2) Les vues

A) Vue formulaire : email.blade.php



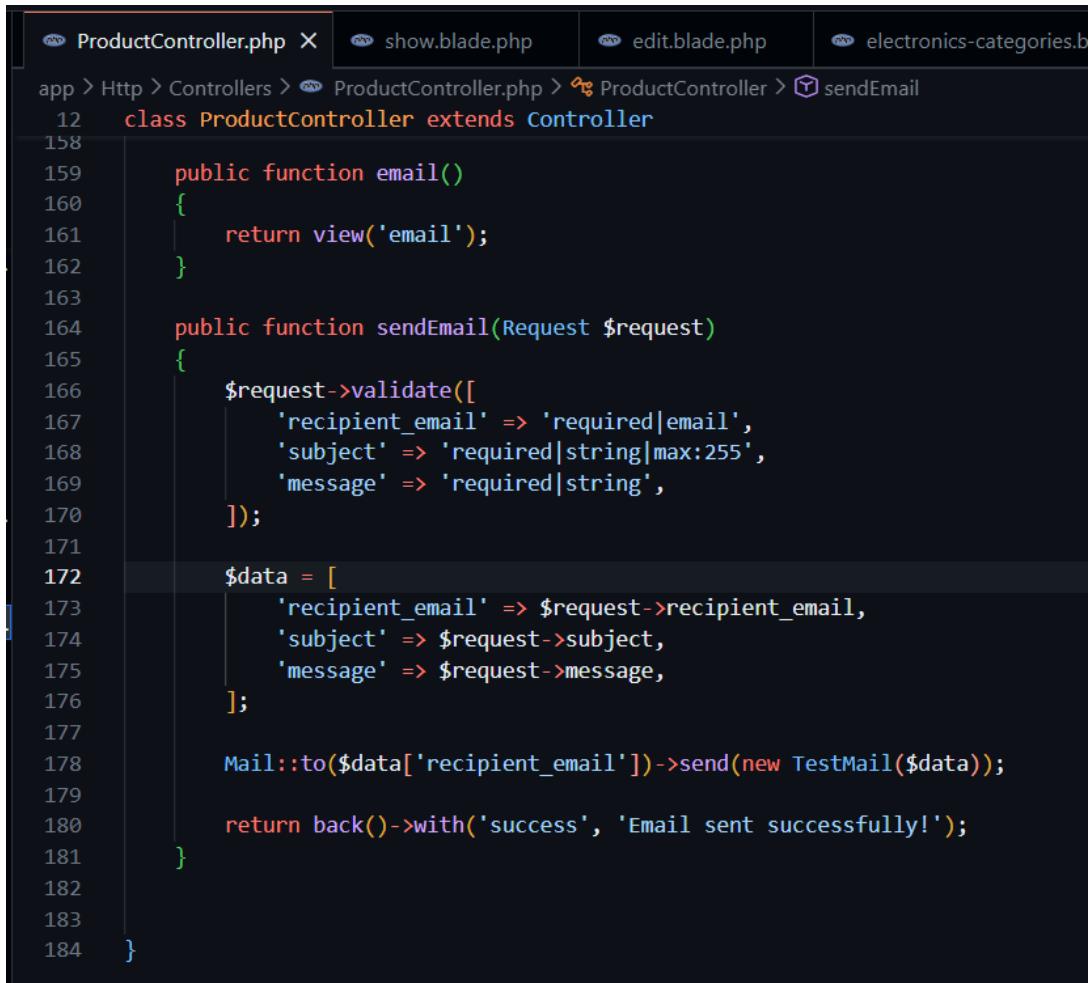
```
resources > views > email.blade.php > ...> div.msone-elegant-wrapper.fade-in-up > h2.msone-hero-title.page-title.text-center
1 @extends('layout.master')
2 @section('title','email')
3
4 @section('content')
5 <div class="msone-elegant-wrapper fade-in-up">
6     <h2 class="msone-hero-title page-title text-center">Email Customers</h2>
7
8     <form action="{{ route('send.email') }}" method="post" class="msone-minimal-form">
9         @csrf
10
11         <div class="msone-field">
12             <label class="msone-label">Recipient Email</label>
13             <input class="msone-input" type="email" name="recipient_email" class="form-control" required>
14         </div>
15
16         <div class="msone-field">
17             <label class="msone-label">Subject</label>
18             <input class="msone-input" type="text" name="subject" class="form-control" required>
19         </div>
20
21         <div class="msone-field">
22             <label class="msone-label">Message</label>
23             <textarea class="msone-input" name="message" class="form-control" rows="5" required></textarea>
24         </div>
25
26         <button class="btn-msone-modern hero-btn primary">Send Email</button>
27     </form>
28 </div>
29 @endsection
30
```

B) Vue email envoyée : emailc.blade.php



```
resources > views > emailc.blade.php > ...
1 <div>
2     Bonjour,<br><br>
3     {{ $msg }}
4 </div>
5
```

3) Le contrôleur : afficher + envoyer



The screenshot shows a code editor with the file `ProductController.php` open. The editor has tabs for `ProductController.php`, `show.blade.php`, `edit.blade.php`, and `electronics-categories.blade.php`. Above the editor, a breadcrumb navigation bar shows the path: `app > Http > Controllers > ProductController.php > ProductController > sendEmail`. The code in `ProductController.php` contains methods for sending emails:

```
12 class ProductController extends Controller
158
159     public function email()
160     {
161         return view('email');
162     }
163
164     public function sendEmail(Request $request)
165     {
166         $request->validate([
167             'recipient_email' => 'required|email',
168             'subject' => 'required|string|max:255',
169             'message' => 'required|string',
170         ]);
171
172         $data = [
173             'recipient_email' => $request->recipient_email,
174             'subject' => $request->subject,
175             'message' => $request->message,
176         ];
177
178         Mail::to($data['recipient_email'])->send(new TestMail($data));
179
180         return back()->with('success', 'Email sent successfully!');
181     }
182
183
184 }
```

4) Les routes

The screenshot shows a code editor with the file `routes/web.php` open. The file contains route definitions using the `Route::middleware` and `Route::group` methods. The code is color-coded, with syntax highlighting for PHP and route parameters. The routes include operations for products creation, storage, editing, updating, and deletion, as well as an admin dashboard and email sending functionality.

```
routes > web.php > Closure
45 Route::middleware(['adminuser'])->group(function () {
46
47     Route::get('/products/create', [ProductController::class, 'create'])
48         ->name('products.create');
49
50     Route::post('/products', [ProductController::class, 'store'])
51         ->name('products.store');
52
53     Route::get('/products/{product}/edit', [ProductController::class, 'edit'])
54         ->name('products.edit');
55
56     Route::put('/products/{product}', [ProductController::class, 'update'])
57         ->name('products.update');
58
59     Route::delete('/products/{product}', [ProductController::class, 'destroy'])
60         ->name('products.destroy');
61
62     Route::get('/admin', fn () => view('admin.dashboard'))
63         ->name('admin.dashboard');
64
65     Route::get('/email', [ProductController::class, 'email']);
66     Route::post('/send/email', [ProductController::class, 'sendEmail'])
67         ->name('send.email');
68 });
});
```