

# **LAPORAN PRAKTIKUM III**

## **LARAVEL**

Disusun untuk Memenuhi Mata Pelajaran Praktikum Pemrograman Web  
Dibimbing oleh Bapak Ricky Septiandy Pratama Putra Husmildin, S.Pd.

Oleh :

**Dimas Yuda Pratama (17691)**

**XI TKJ 2**



**SEKOLAH MENENGAH KEJURUAN NEGERI 2 MALANG**  
**KONSENTRASI KEAHLIAN TEKNIK KOMPUTER DAN JARINGAN**  
**MARET 2025**

## A. LATIHAN

### 1. Persiapan Awal:

- Kode program

```
PS C:\xampp\htdocs\perpustakaan-sekolah-auth> composer require laravel/breeze --dev
```

```
PS C:\xampp\htdocs\perpustakaan-sekolah-auth> php artisan breeze:install
```

```
PS C:\xampp\htdocs\perpustakaan-sekolah-auth> npm install
```

```
PS C:\xampp\htdocs\perpustakaan-sekolah-auth> npm run dev
```

- Keterangan
  - Menginstall Laravel Breeze pada project CRUD sebelumnya

### 2. Implementasi Authentication:

- Kode program

```
PS C:\xampp\htdocs\perpustakaan-sekolah-auth> php artisan make:migration create_roles_table
```

```
database > migrations > 2025_04_15_005844_create_roles_table.php
```

```
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()
13     {
14         Schema::create('roles', function (Blueprint $table) {
15             $table->id();
16             $table->string('name');
17             $table->timestamps();
18         });
19     }
20
21     /**
22      * Reverse the migrations.
23      */
24     public function down(): void
25     {
26         Schema::dropIfExists('roles');
27     }
28 };
29
```

```
PS C:\xampp\htdocs\perpustakaan-sekolah-auth> php artisan make:migration add_role_id_to_users
```

```
database > migrations > 2025_04_15_005905_add_role_id_to_users.php
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()
13     {
14         Schema::table('users', function (Blueprint $table) {
15             $table->foreignId('role_id')->after('id')->constrained();
16         });
17     }
18
19     /**
20      * Reverse the migrations.
21      */
22     public function down(): void
23     {
24         Schema::table('users', function (Blueprint $table) {
25             //
26         });
27     }
28 };
29
```

```
PS C:\xampp\htdocs\perpustakaan-sekolah-auth> php artisan make:seeder RoleSeeder
```

```
database > seeders > RoleSeeder.php
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use Illuminate\Support\Facades\DB;
8
9  class RoleSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      */
14     public function run()
15     {
16         DB::table('roles')->insert([
17             ['name' => 'Admin'],
18             ['name' => 'Pustakawan'],
19             ['name' => 'Anggota']
20         ]);
21     }
22 }
23
```

```

database > seeders > DatabaseSeeder.php
1  <?php
2
3  namespace Database\Seeders;
4
5  use App\Models\User;
6  use Illuminate\Support\Facades\Hash;
7  use Illuminate\Database\Seeder;
8
9  class DatabaseSeeder extends Seeder
10 {
11     /**
12      * Seed the application's database.
13      */
14     public function run()
15     {
16         $this->call([
17             RoleSeeder::class
18         ]);
19
20         // Admin
21         User::create([
22             'name' => 'Admin',
23             'email' => 'admin@example.com',
24             'password' => Hash::make('123'),
25             'role_id' => 1
26         ]);
27
28         // Pustakawan
29         User::create([
30             'name' => 'Pustakawan',
31             'email' => 'pustakawan@example.com',
32             'password' => Hash::make('123'),
33             'role_id' => 2
34         ]);
35
36         // Anggota
37         User::create([
38             'name' => 'Anggota',
39             'email' => 'anggota@example.com',
40             'password' => Hash::make('123'),
41             'role_id' => 3
42         ]);
43     }
44 }
45

```

```

PS C:\xampp\htdocs\perpustakaan-sekolah-auth> php artisan migrate --seed

```

- Keterangan
  - Buat migration untuk roles, lalu edit file  
// database/migrations/[timestamp]\_create\_roles\_table.php
  - Buat migration untuk menambahkan kolom role\_id ke users, lalu edit file  
// database/migrations/[timestamp]\_add\_role\_id\_to\_users.php

- Buat seeder untuk roles, lalu edit file  
// database/seeder/RoleSeeder.php
- Update DatabaseSeeder pada file  
// database/seeder/DatabaseSeeder.php
- Jalankan migration dan seeder

### 3. Implementasi Model dan Relationship:

- Kode program

```
public function role()
{
    return $this->belongsTo(Role::class);
}

/**
 * Cek apakah user memiliki role tertentu
 */
public function hasRole($role)
{
    return $this->role && $this->role->name === $role;
}
```

PS C:\xampp\htdocs\perpustakaan-sekolah-auth> php artisan make:model Role

```
app > Models > Role.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Model;
6  use Illuminate\Database\Eloquent\Factories\HasFactory;
7
8  class Role extends Model
9  {
10     use HasFactory;
11
12     protected $fillable = ['name'];
13
14     public function users()
15     {
16         return $this->hasMany(User::class);
17     }
18 }
```

- Keterangan
  - Update User model pada file  
// app/Models/User.php
  - Buat Role model
  - Edit Role model pada file  
// app/Models/User.php

#### 4. Implementasi Authorization:

- Kode program

```
PS C:\xampp\htdocs\perpustakaan-sekolah-auth> php artisan make:middleware CheckRole
```

```
app > Http > Middleware > CheckRole.php
1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Closure;
6  use Illuminate\Support\Facades\Auth;
7
8  class CheckRole
9  {
10     public function handle($request, Closure $next, ...$roles)
11     {
12         $user = Auth::user();
13
14         if (!$user || !in_array($user->role_id, $this->mapRoles($roles))) {
15             return redirect()->route('dashboard')->with('error', 'Unauthorized access');
16         }
17
18         return $next($request);
19     }
20
21     private function mapRoles($roles)
22     {
23         // Mapping nama role ke role_id dari database
24         $map = [
25             'Admin' => 1,
26             'Pustakawan' => 2,
27             'Anggota' => 3,
28         ];
29
30         return array_map(fn($role) => $map[$role] ?? null, $roles);
31     }
32 }
33
```

```

bootstrap > app.php
1  <?php
2
3  use Illuminate\Foundation\Application;
4  use Illuminate\Foundation\Configuration\Exceptions;
5  use Illuminate\Foundation\Configuration\Middleware;
6
7  return Application::configure(basePath: dirname(__DIR__))
8      ->withRouting(
9          web: __DIR__.'/../routes/web.php',
10         commands: __DIR__.'/../routes/console.php',
11         health: '/up',
12     )
13     ->withMiddleware(function (Middleware $middleware) {
14         $middleware->alias([
15             'role' => \App\Http\Middleware\CheckRole::class,
16         ]);
17     })
18     ->withExceptions(function (Exceptions $exceptions) {
19         //
20     }->create());

```

```

routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\HomeController;
5  use App\Http\Controllers\BukuController;
6  use App\Http\Controllers\PeminjamanController;
7  use App\Http\Controllers\UserController;
8  use App\Http\Controllers\ProfileController;
9
10 // Redirect halaman utama langsung ke login
11 Route::get('/', function () {
12     return redirect()->route('login');
13 });
14
15 // Route untuk halaman home
16 Route::get('/home', [HomeController::class, 'index'])->name('home');
17
18 Route::middleware(['auth'])->group(function () {
19
20     // Pengalihan dari /dashboard ke /home setelah login
21     Route::get('/dashboard', function () {
22         return redirect()->route('home');
23     }->name('dashboard'));
24
25 // AKSES BUKU: hanya Admin dan Pustakawan
26 Route::middleware(['role:Admin,Pustakawan'])->group(function () {
27     Route::resource('buku', BukuController::class);
28 });
29
30 // AKSES PEMINJAMAN: hanya Admin dan Anggota
31 Route::middleware(['role:Admin,Anggota'])->group(function () {
32     Route::resource('peminjaman', PeminjamanController::class);
33     Route::post('/peminjaman/{peminjaman}/return', [PeminjamanController::class, 'returnBook'])->name('peminjaman.return');
34 });
35
36 // Route untuk mengembalikan buku
37 Route::post('/peminjaman/{peminjaman}/return', [PeminjamanController::class, 'returnBook'])->name('peminjaman.return');
38 });
39
40 // ===== PROFILE =====
41 Route::get('/profile', [ProfileController::class, 'edit'])->name('profile.edit');
42 Route::patch('/profile', [ProfileController::class, 'update'])->name('profile.update');
43 Route::delete('/profile', [ProfileController::class, 'destroy'])->name('profile.destroy');
44
45 // Auth routes (Breeze)
46 require __DIR__.'/auth.php';
47

```

- Keterangan
  - Buat middleware untuk role
  - Edit CheckRole middleware  
`// app/Models/User.php`
  - Daftarkan middleware di app.php  
`// bootstrap/app.php`
  - Edit route  
`// routes/web.php`

#untuk output program dari project ini akan saya tampilkan dibagian Additionally

## B. Additionally

1. Setelah menyelesaikan semua tahapan Authentication & Authorization pada modul, saya menambahkan beberapa tahapan sampai web dapat diakses sepenuhnya sesuai dengan ketentuan berikut berikut: :

- Memperjelas fungsi dari setiap user seperti :
  - Admin → dapat akses penuh
  - Pustakawan → dapat akses bagian daftar buku
  - Anggota → dapat akses bagian daftar peminjaman

## HASIL

### 1. Kode Program:



```

<body>
  <div class="header">
    <div style="display: flex; align-items: center;">
      <a href="{{ url('/') }}" style="display: flex; align-items: center; text-decoration: none; color: white;">
        <img alt="Library Logo" style="height: 40px; margin-right: 10px;"/>
        <div>Perpustakaan Sekolah</div>
      </a>
    </div>
    <div class="nav-buttons">
      <a href="{{ route('buku.index') }}" class="btn">📖 Daftar Buku</a>
      <a href="{{ route('peminjaman.index') }}" class="btn">🏠 Peminjaman</a>

      @auth
        <div class="dropdown">
          <button class="btn" onclick="toggleDropdown()">👤 {{ Auth::user()->name }}</button>
          <div id="accountDropdown" class="dropdown-menu">
            <a href="{{ route('profile.edit') }}">⚙️ Profil</a>
            <form method="POST" action="{{ route('logout') }}">
              @csrf
              <button type="submit">🚪 Logout</button>
            </form>
          </div>
        </div>
      @endauth
    </div>
  </div>

  <!-- Main Content -->
  <div class="main-content">
    <div class="content-wrapper">
      @yield('content')
    </div>
  </div>

  <!-- Footer -->
  <div class="footer">
    <p>©copy; 2025 Perpustakaan Sekolah. SMK Negeri 2 Malang.</p>
  </div>

  <script>
    function toggleDropdown() {
      const dropdown = document.getElementById('accountDropdown');
      dropdown.style.display = dropdown.style.display === 'block' ? 'none' : 'block';
    }

    window.addEventListener('click', function (e) {
      const dropdown = document.getElementById('accountDropdown');
      const button = document.querySelector('.dropdown button');
      if (!button.contains(e.target) && !dropdown.contains(e.target)) {
        dropdown.style.display = 'none';
      }
    });
  </script>
</body>

```

```

resources > views > profile > edit.blade.php
1  @extends('layouts.app')
2
3  @section('title', 'Profile')
4
5  @section('content')
6  <div class="settings-container">
7
8      <div class="settings-wrapper">
9          <div class="settings-sidebar">
10             <div class="profile-avatar">
11                 <div class="avatar-image">
12                     <span>{{ substr(Auth::user()->name, 0, 1) }}</span>
13                 </div>
14                 <h3>{{ Auth::user()->name }}</h3>
15                 <p>{{ Auth::user()->email }}</p>
16             </div>
17
18             <ul class="settings-nav">
19                 <li class="active"><a href="#profile">Personal Info</a></li>
20                 <li><a href="#security">Security</a></li>
21                 <li><a href="#account">Account</a></li>
22             </ul>
23         </div>
24
25         <div class="settings-content">
26             <section id="profile" class="settings-section">
27                 <h3 class="section-title">Personal Information</h3>
28                 <p class="section-description">Update your account's profile information and email address.</p>
29
30                 <form method="post" action="{{ route('profile.update') }}" class="settings-form">
31                     @csrf
32                     @method('patch')
33
34                     <div class="form-group">
35                         <label for="name">Name</label>
36                         <input id="name" name="name" type="text" class="form-control" value="{{ old('name', Auth::user()->name) }}" required autofocus autocomplete="name">
37                         @error('name')
38                             <p class="error-message">{{ $message }}</p>
39                         @enderror
40                     </div>
41
42                     <div class="form-group">
43                         <label for="email">Email</label>
44                         <input id="email" name="email" type="email" class="form-control" value="{{ old('email', Auth::user()->email) }}" required autocomplete="email">
45                         @error('email')
46                             <p class="error-message">{{ $message }}</p>
47                         @enderror
48                     </div>
49
50                     <div class="form-actions">
51                         <button type="submit" class="btn btn-primary">Save Changes</button>
52                     </div>
53                 </form>
54             </section>
55
56             <section id="security" class="settings-section">
57                 <h3 class="section-title">Update Password</h3>
58                 <p class="section-description">Ensure your account is using a long, random password to stay secure.</p>
59
60                 <form method="post" action="{{ route('password.update') }}" class="settings-form">
61                     @csrf

```

```

62     @method('put')
63
64     <div class="form-group">
65       <label for="current_password">Current Password</label>
66       <input id="current_password" name="current_password" type="password" class="form-control" autocomplete="current-password">
67       @error('current_password')
68       <p class="error-message">{{ $message }}</p>
69     @enderror
70   </div>
71
72   <div class="form-group">
73     <label for="password">New Password</label>
74     <input id="password" name="password" type="password" class="form-control" autocomplete="new-password">
75     @error('password')
76     <p class="error-message">{{ $message }}</p>
77   @enderror
78 </div>
79
80 <div class="form-group">
81   <label for="password_confirmation">Confirm Password</label>
82   <input id="password_confirmation" name="password_confirmation" type="password" class="form-control" autocomplete="new-password">
83   @error('password_confirmation')
84   <p class="error-message">{{ $message }}</p>
85 @enderror
86 </div>
87
88 <div class="form-actions">
89   <button type="submit" class="btn btn-primary">Update Password</button>
90 </div>
91 </form>
92 </section>
93
94 <section id="account" class="settings-section">
95   <h3 class="section-title">Delete Account</h3>
96   <p class="section-description">Once your account is deleted, all of its resources and data will be permanently deleted.
97   Before deleting your account, please download any data or information that you wish to retain.</p>
98
99   <form method="post" action="{{ route('profile.destroy') }}" class="settings-form danger-zone">
100     @csrf
101     @method('delete')
102
103     <div class="form-group">
104       <label for="password_delete">Password</label>
105       <input id="password_delete" name="password" type="password" class="form-control" placeholder="Enter your password to confirm">
106       @error('password')
107       <p class="error-message">{{ $message }}</p>
108     @enderror
109   </div>
110
111   <div class="form-actions">
112     <button type="submit" class="btn btn-danger">Delete Account</button>
113   </div>
114 </form>
115 </section>
116 </div>
117 </div>

```

```

118 </div>
119
120 <style>
121   .settings-container {
122     max-width: 1100px;
123     margin: 2rem auto;
124     padding: 0 1rem;
125   }
126
127   .settings-title {
128     font-size: 1.75rem;
129     font-weight: 600;
130     color: #111827;
131     margin-bottom: 1.5rem;
132   }
133
134   .settings-wrapper {
135     display: flex;
136     background: #fff;
137     border-radius: 12px;
138     box-shadow: 0 1px 3px #00000010;
139     overflow: hidden;
140   }
141
142   .settings-sidebar {
143     width: 250px;
144     background: #fff8f3; /* Light orange background */
145     border-right: 1px solid #ffdfca;
146     padding: 1.5rem 0;
147   }
148
149   .profile-avatar {
150     padding: 0 1.5rem 1.5rem;
151     text-align: center;
152     border-bottom: 1px solid #ffdfca;
153     margin-bottom: 1.5rem;
154   }
155
156   .avatar-image {
157     width: 80px;
158     height: 80px;
159     border-radius: 50%;
160     background: #f97316; /* Orange */
161     display: flex;
162     align-items: center;
163     justify-content: center;
164     margin: 0 auto 1rem;
165     font-size: 1.5rem;
166     font-weight: 600;
167     color: white;
168   }
169
170   .profile-avatar h3 {
171     font-size: 1.1rem;
172     font-weight: 600;
173     margin-bottom: 0.25rem;
174     color: #111827;
175   }
176
177   .profile-avatar p {

```

```

178     font-size: 0.8/5rem;
179     color: #6b7280;
180     word-break: break-all;
181 }
182
183 .settings-nav {
184     list-style: none;
185     padding: 0;
186     margin: 0;
187 }
188
189 .settings-nav li {
190     margin: 0;
191     padding: 0;
192 }
193
194 .settings-nav a {
195     display: block;
196     padding: 0.75rem 1.5rem;
197     color: #4b5563;
198     text-decoration: none;
199     font-size: 0.95rem;
200     transition: all 0.2s ease;
201     border-left: 3px solid transparent;
202 }
203
204 .settings-nav a:hover {
205     background-color: #fff;
206     color: #f97316; /* Orange on hover */
207     border-left-color: #f97316;
208 }
209
210 .settings-nav li.active a {
211     background-color: #fff;
212     color: #f97316;
213     border-left-color: #f97316;
214     font-weight: 500;
215 }
216
217 .settings-content {
218     flex: 1;
219     padding: 2rem;
220 }
221
222 .settings-section {
223     max-width: 600px;
224 }
225
226 .section-title {
227     font-size: 1.25rem;
228     font-weight: 600;
229     color: #111827;
230     margin-bottom: 0.75rem;
231     text-align: left;
232 }
233
234 .section-description {
235     color: #6b7280;
236     margin-bottom: 1.5rem;
237     text-align: left;

```

```

238     line-height: 1.5;
239 }
240
241 .settings-form {
242     margin-top: 1.5rem;
243     width: 100%;
244 }
245
246 .form-group {
247     margin-bottom: 1.5rem;
248     text-align: left;
249     width: 100%;
250 }
251
252 .form-group label {
253     display: block;
254     font-weight: 500;
255     margin-bottom: 0.5rem;
256     color: #111827;
257     text-align: left;
258 }
259
260 .form-control {
261     width: 100%;
262     padding: 0.75rem;
263     border-radius: 0.375rem;
264     border: 1px solid #d1d5db;
265     background: #fff;
266     transition: all 0.2s;
267     font-size: 1rem;
268 }
269
270 .form-control:focus {
271     border-color: #f97316;
272     box-shadow: 0 0 0 2px rgba(249, 115, 22, 0.2);
273     outline: none;
274 }
275
276 .form-actions {
277     display: flex;
278     justify-content: flex-start;
279     margin-top: 2rem;
280 }
281
282 .btn {
283     display: inline-block;
284     padding: 0.75rem 1.5rem;
285     border-radius: 0.375rem;
286     font-weight: 500;
287     text-align: center;
288     cursor: pointer;
289     transition: all 0.2s;
290     border: none;
291     font-size: 1rem;
292 }
293
294 .btn-primary {
295     background-color: #f97316;
296     color: white;

```

```

297     }
298
299     .btn-primary:hover {
300         background-color: #ea580c;
301     }
302
303     .btn-danger {
304         background-color: #ef4444;
305         color: white;
306     }
307
308     .btn-danger:hover {
309         background-color: #dc2626;
310     }
311
312     .error-message {
313         color: #ef4444;
314         font-size: 0.875rem;
315         margin-top: 0.5rem;
316     }
317
318     .danger-zone {
319         border-top: 1px solid #ffdfca;
320         padding-top: 2rem;
321         margin-top: 2rem;
322     }
323
324     /* Responsive design */
325     @media (max-width: 768px) {
326         .settings-wrapper {
327             flex-direction: column;
328         }
329
330         .settings-sidebar {
331             width: 100%;
332             border-right: none;
333             border-bottom: 1px solid #ffdfca;
334         }
335
336         .settings-nav {
337             display: flex;
338             overflow-x: auto;
339             padding-bottom: 0.5rem;
340         }
341
342         .settings-nav li {
343             flex-shrink: 0;
344         }
345
346         .settings-nav a {
347             border-left: none;
348             border-bottom: 3px solid transparent;
349             white-space: nowrap;
350         }
351
352         .settings-nav a:hover,
353         .settings-nav li.active a {
354             border-left-color: transparent;
355             border-bottom-color: #f97316;
356         }
357

```

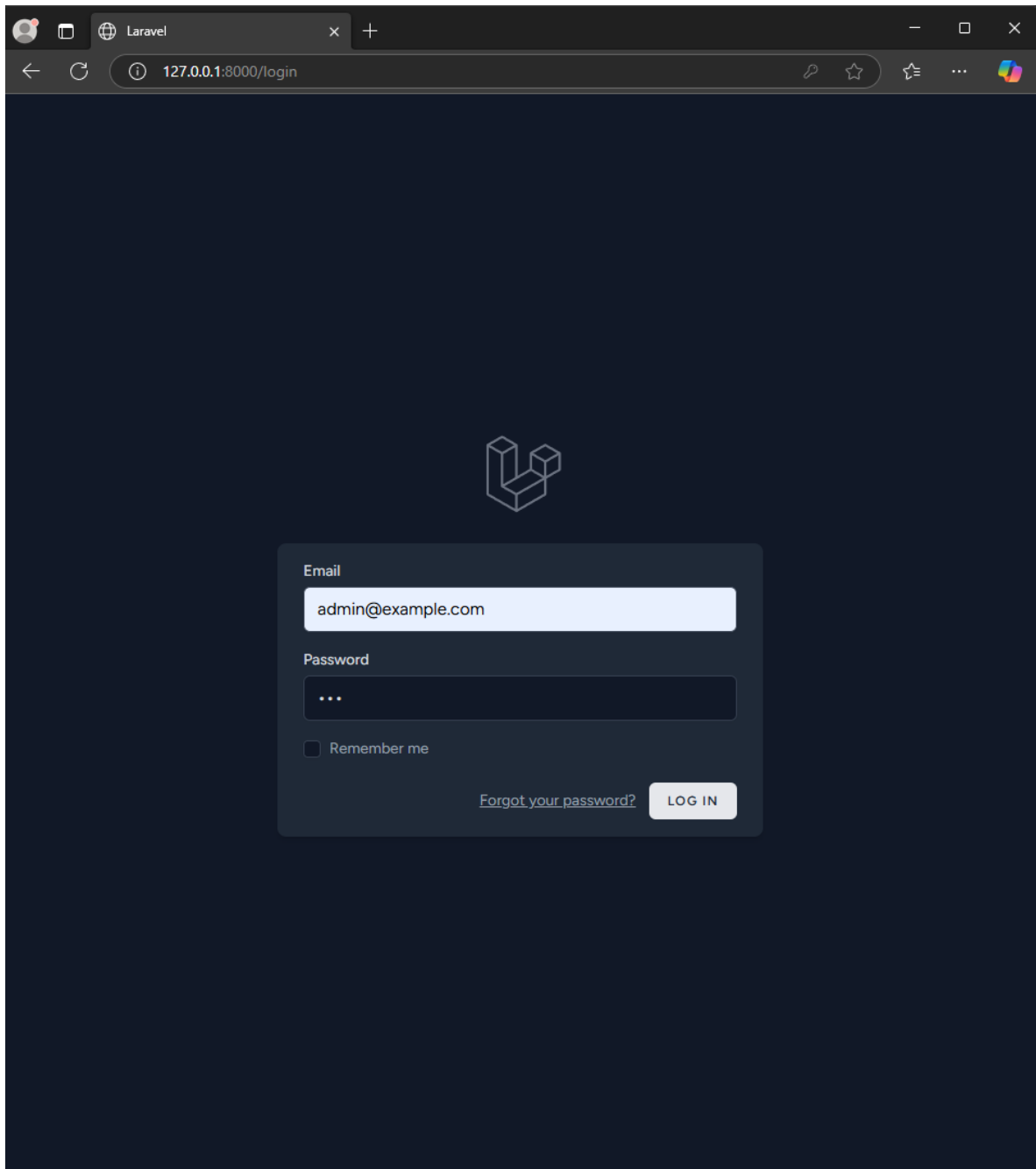
```

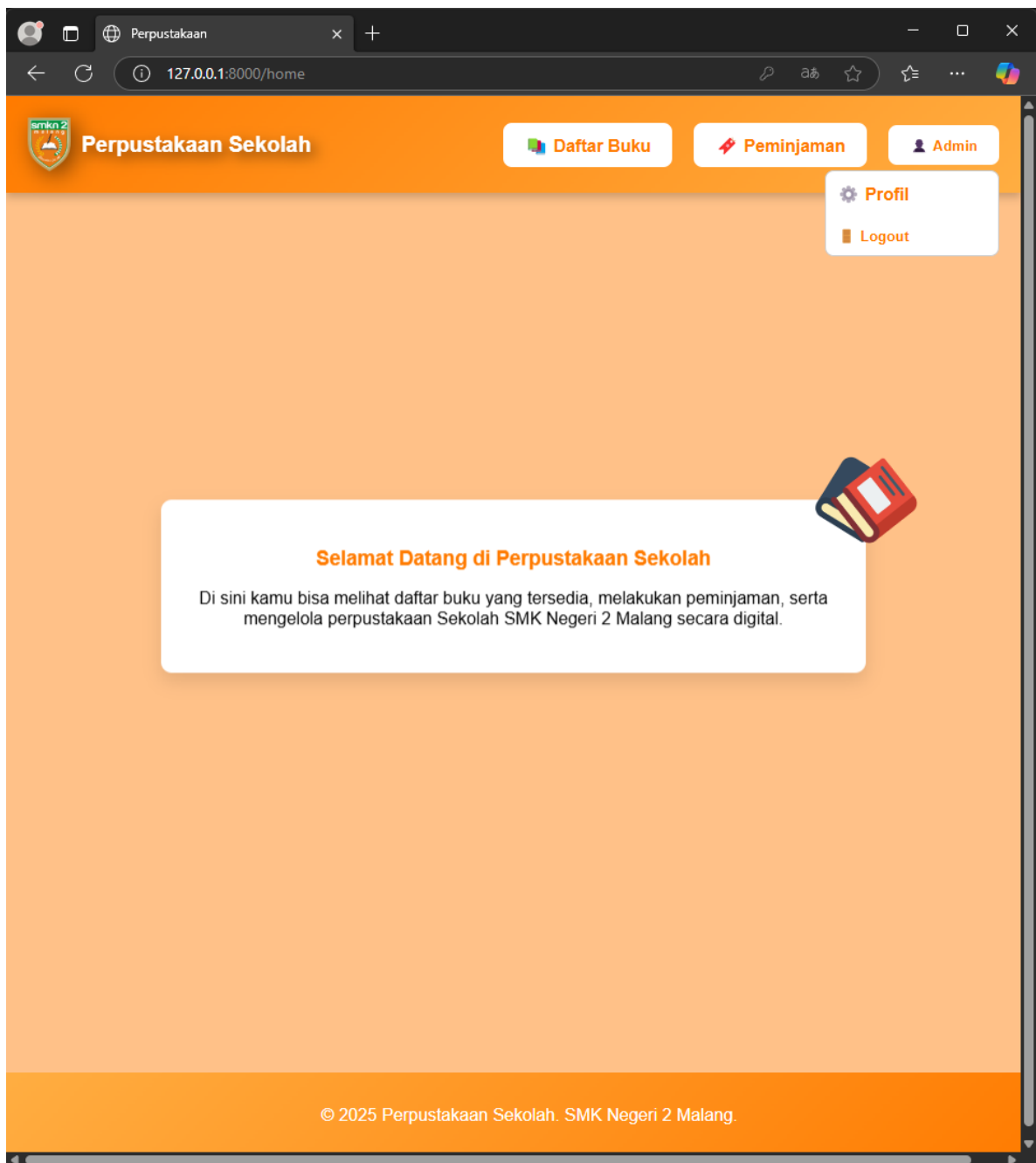
357
358     .settings-content {
359         padding: 1.5rem 1rem;
360     }
361 }
362 </style>
363
364 <script>
365     // Simple tab navigation functionality
366     document.addEventListener('DOMContentLoaded', function() {
367         const navItems = document.querySelectorAll('.settings-nav li');
368         const sections = document.querySelectorAll('.settings-section');
369
370         // Show only the first section initially
371         sections.forEach((section, index) => {
372             if (index > 0) section.style.display = 'none';
373         });
374
375         navItems.forEach(item => {
376             item.addEventListener('click', function(e) {
377                 e.preventDefault();
378
379                 // Remove active class from all items
380                 navItems.forEach(i => i.classList.remove('active'));
381
382                 // Add active class to clicked item
383                 this.classList.add('active');
384
385                 // Get the target section id
386                 const targetId = this.querySelector('a').getAttribute('href').substring(1);
387
388                 // Hide all sections and show the target one
389                 sections.forEach(section => {
390                     section.style.display = section.id === targetId ? 'block' : 'none';
391                 });
392             });
393         });
394     });
395 </script>
396 @endsection

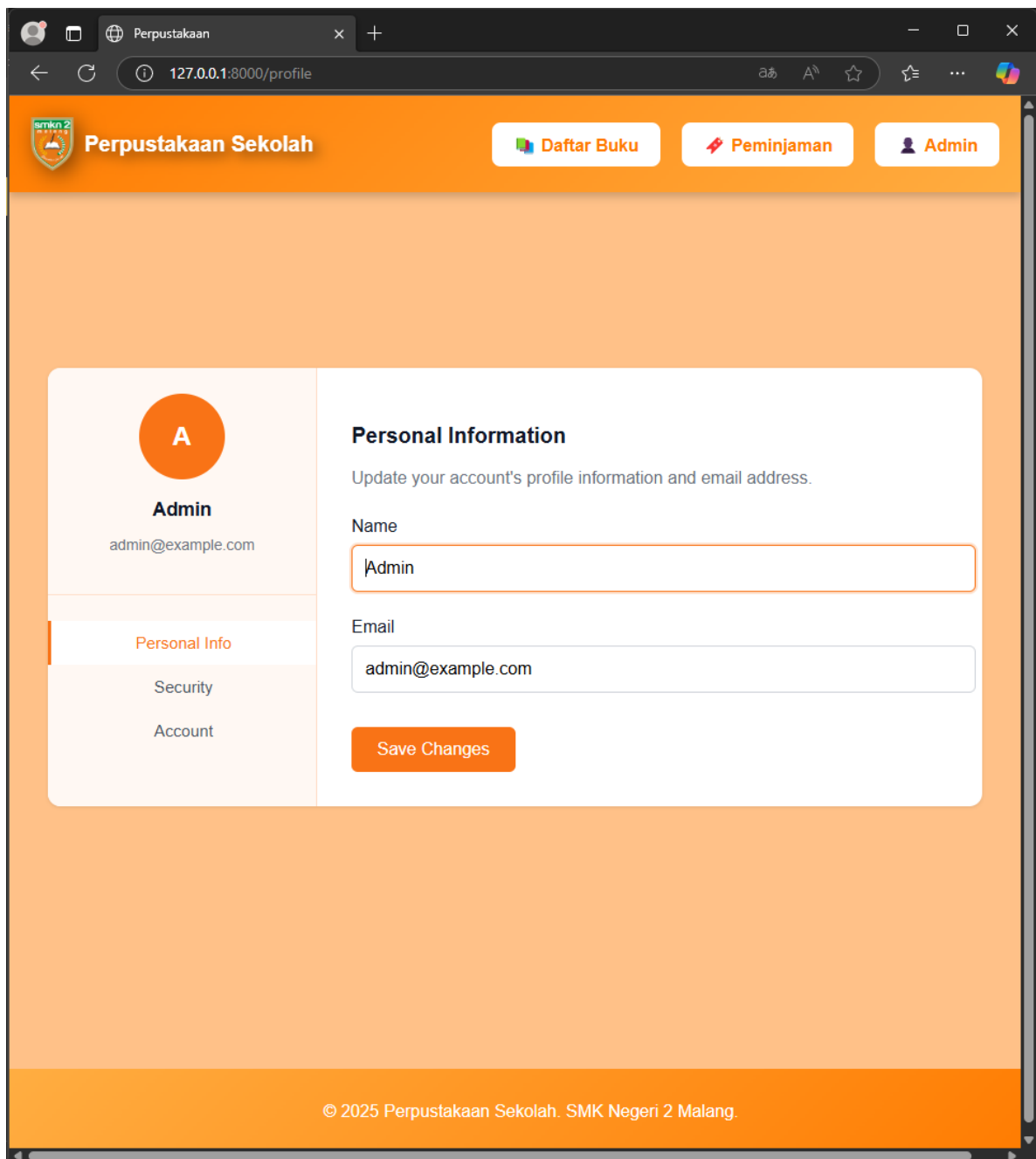
```

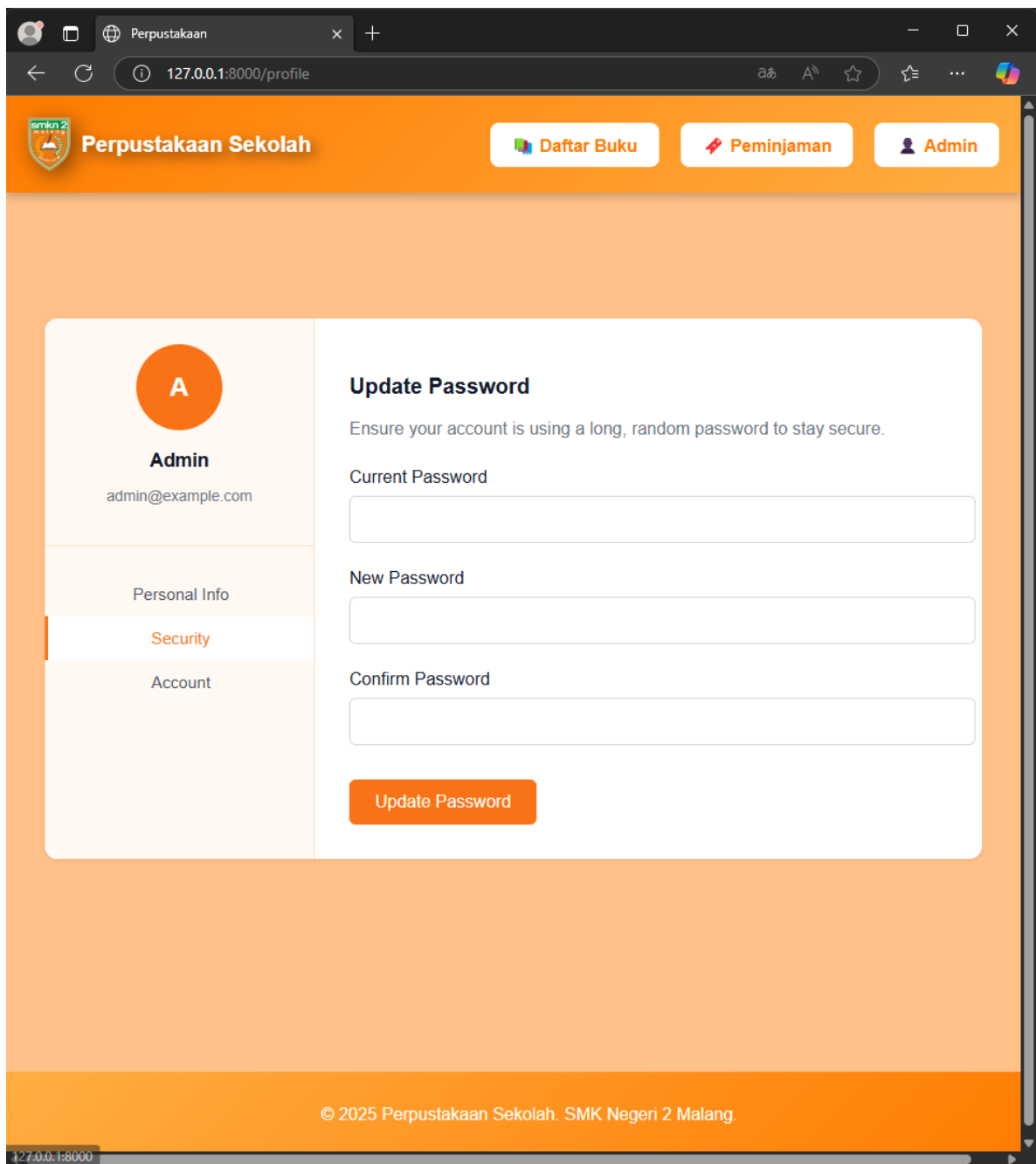


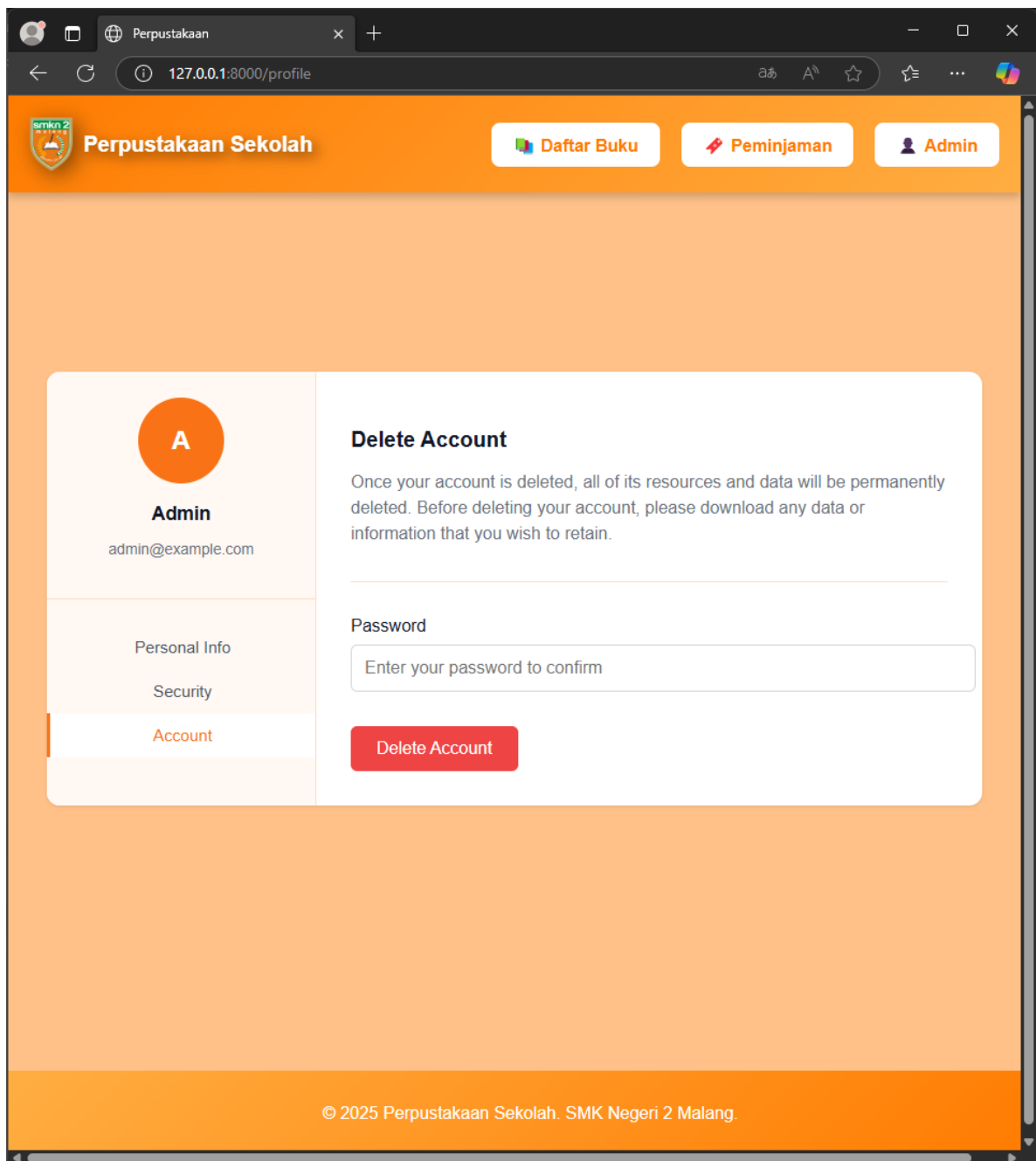
## 2. Output Program











### 3. Keterangan

Karena semua konfigurasi sudah saya satukan pada Langkah-langkah sebelumnya dibagian additional ini hanya menambahkan untuk bagaimana *Authentication & Authorization* ditampilkan, karena yang awalnya tidak menggunakan extends, seperti Format: `<x-layouts.app>` atau `<x-guest-layout>`, dst. Jadi menggunakan extends. Dan juga memodifikasi arahan/tujuan halaman web yang secara default Laravel breeze akan mengarahkan ke tampilan dashboard setelah login, disini saya ubah dengan mengarahkan ke tampilan home(frontend saya). Dan juga saya membuat tampilan sederhana untuk profile setting agar memiliki fitur yang sama seperti pada default Laravel breeze yaitu fitur *Update Profile Information*, *Update Password* dan *Delete Account*.