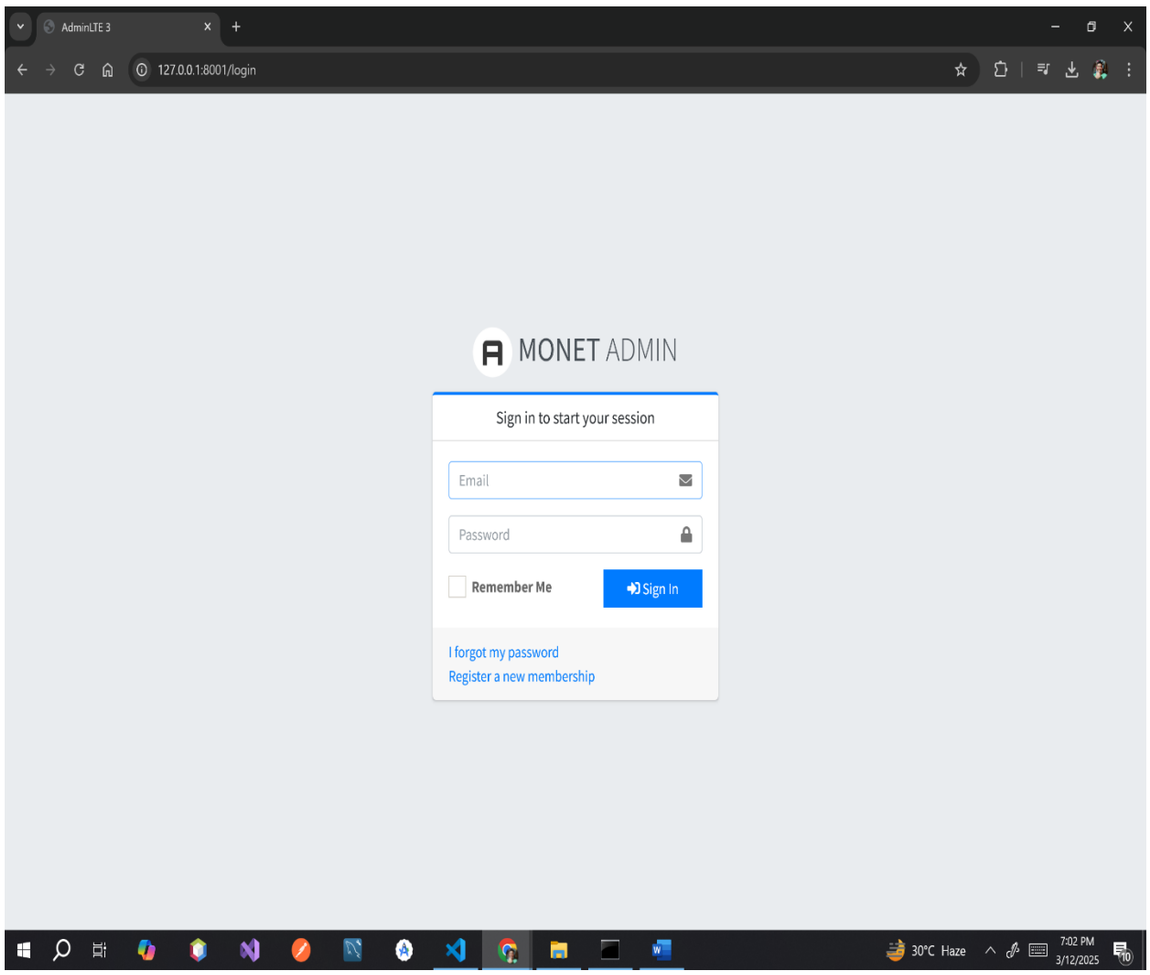
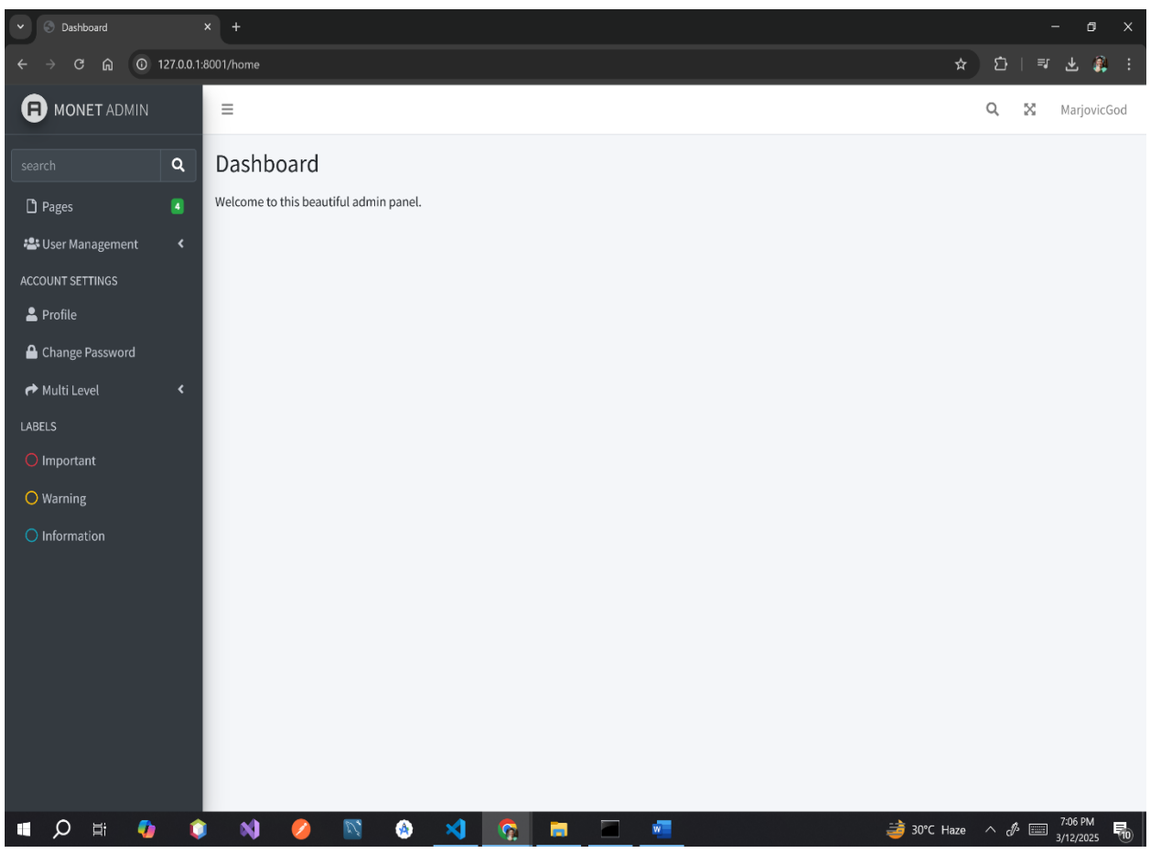
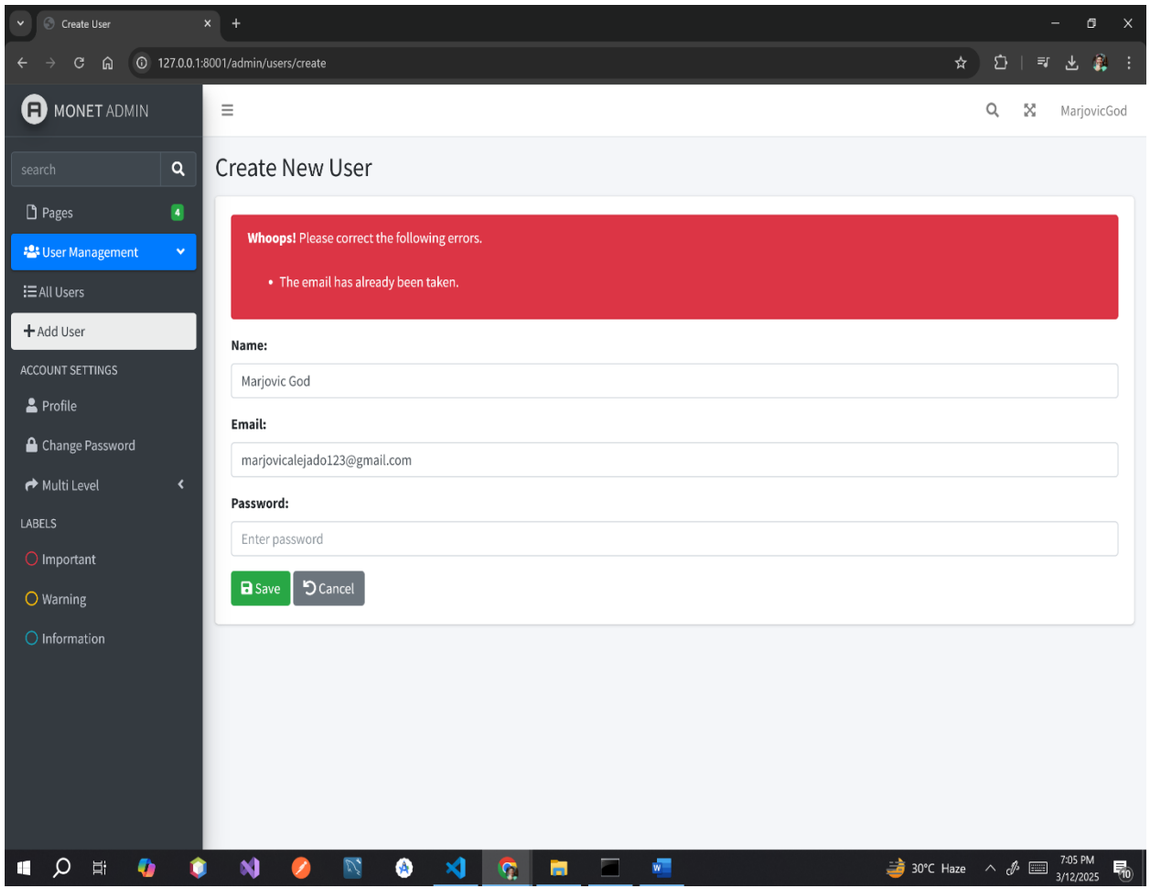
GITHUB LINK: https://github.com/MONET-SYSTEM/monetAPI

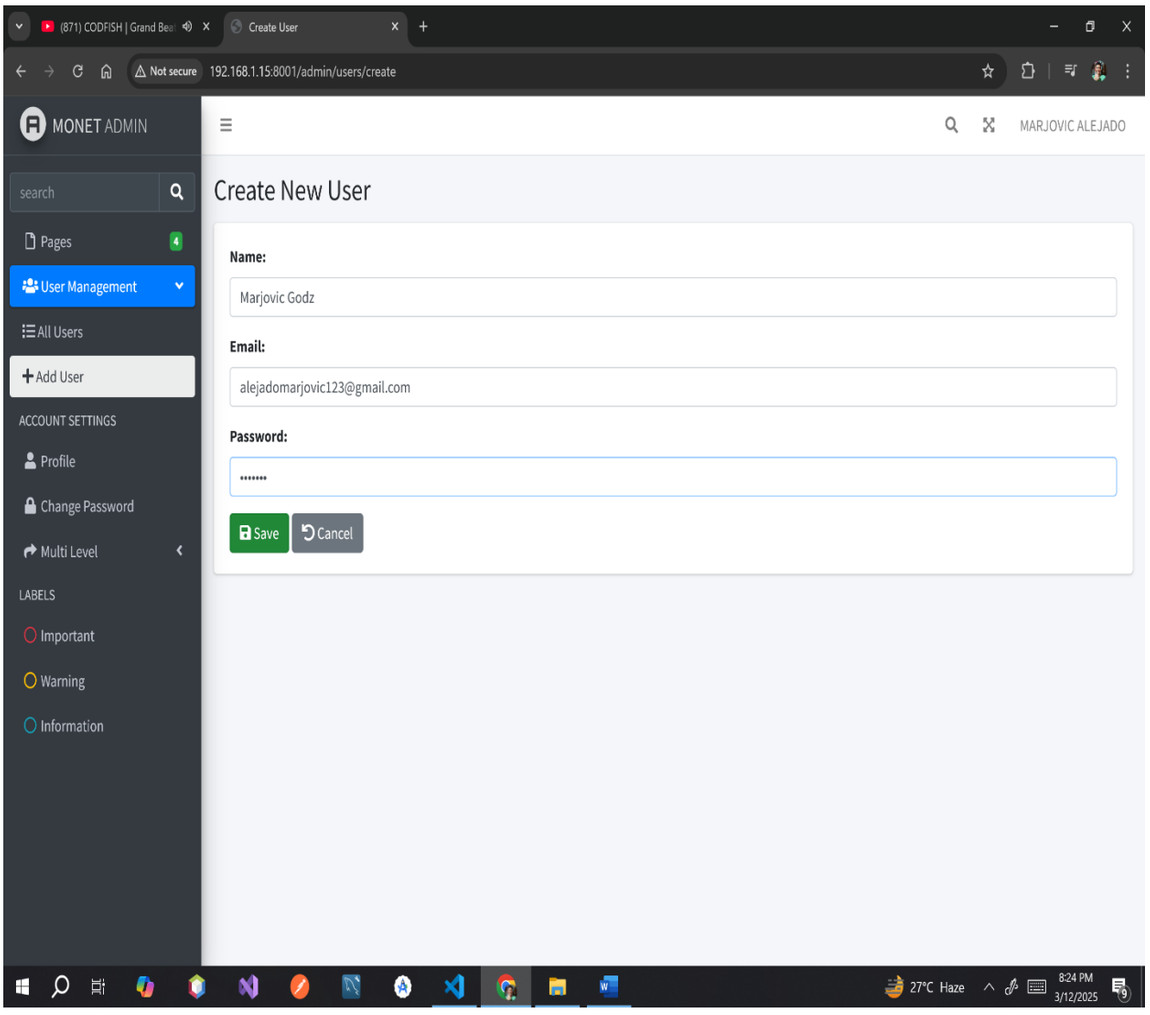
CRUD MONET

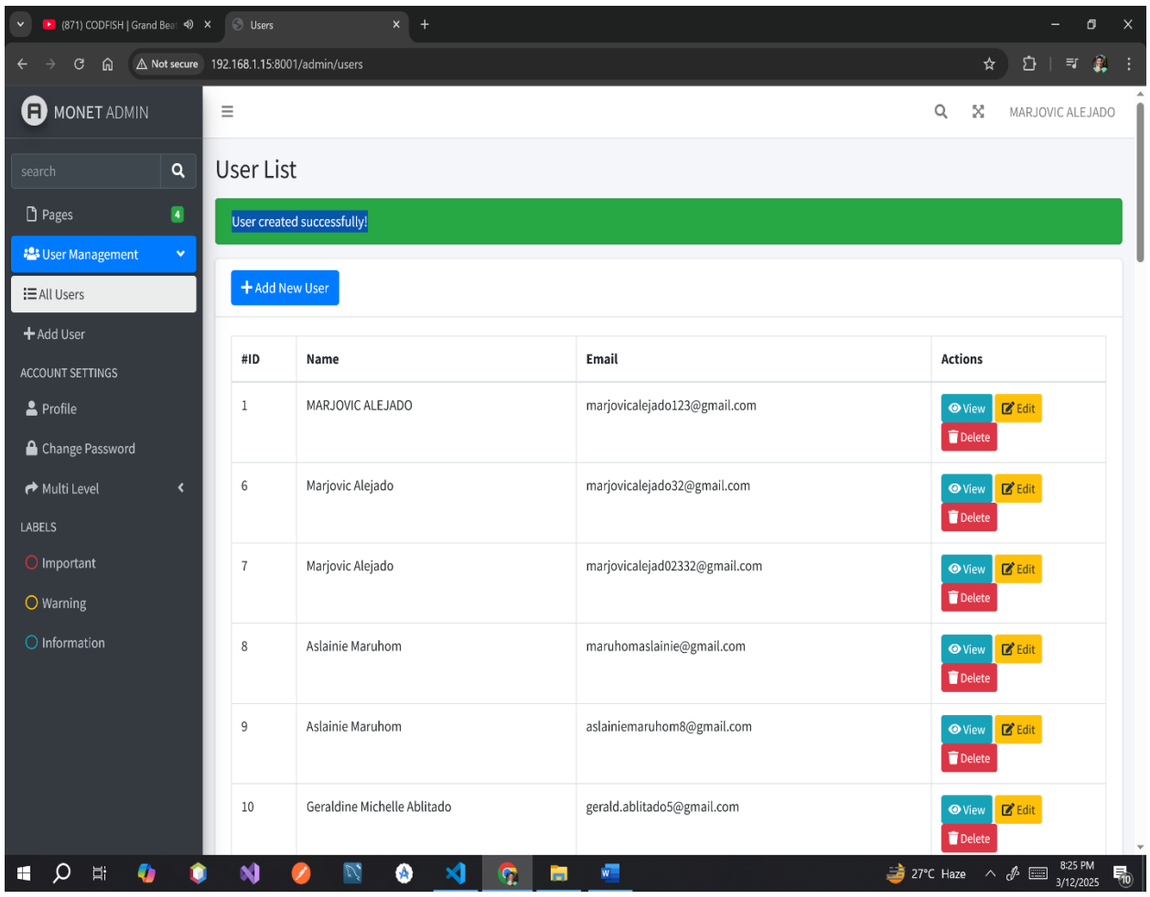


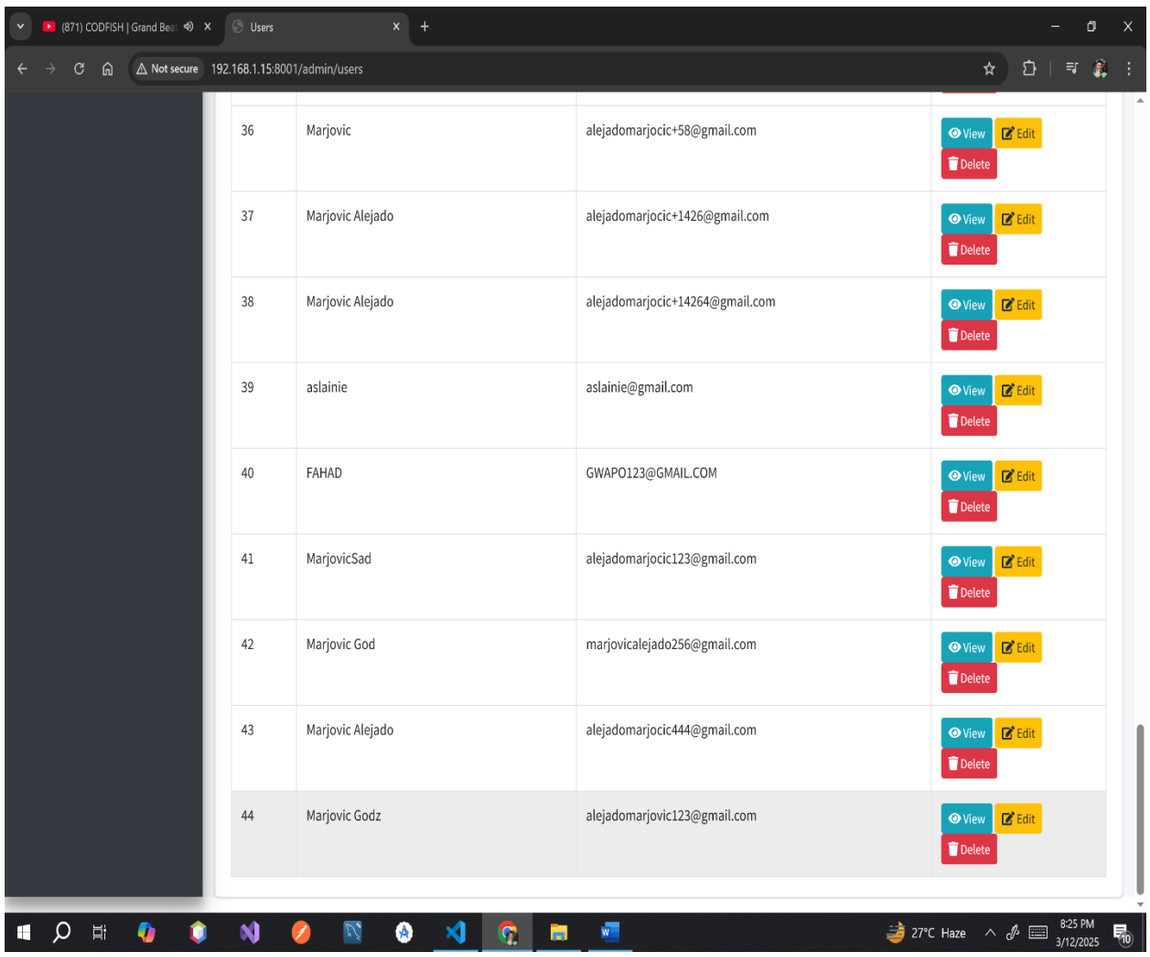


CREATE USER

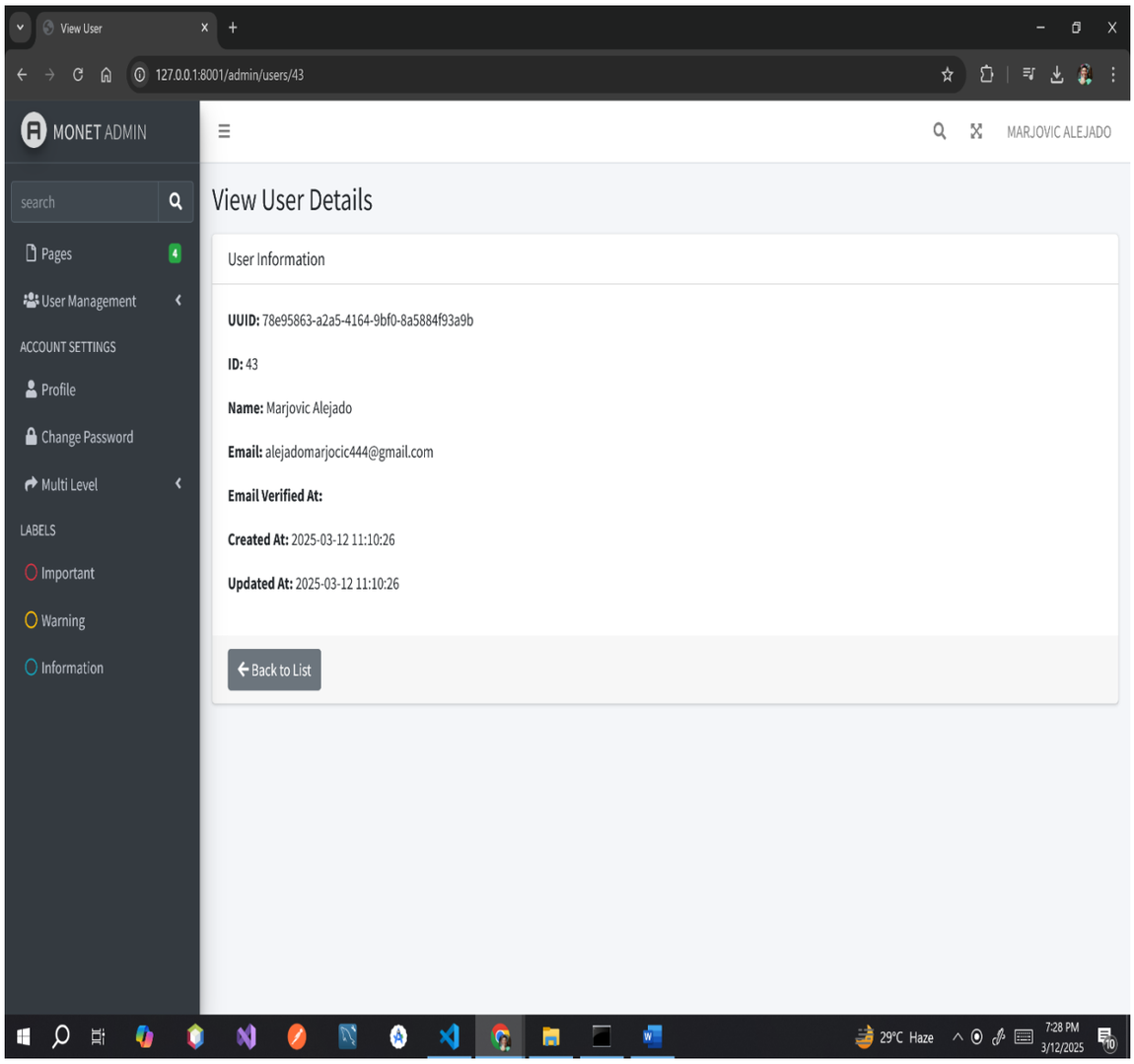

Create User
127.0.0.1:8001/admin/users/create
MONET ADMIN
MarjovicGod
search
Pages
User Management
All Users
Add User
ACCOUNT SETTINGS
Profile
Change Password
Multi Level
LABELS
Important
Warning
Information
Create New User
Whoops! Please correct the following errors.
The email has already been taken.
Name:
Marjovic God
Email:
marjovicalejado123@gmail.com
Password:
Enter password
Save
Cancel
30°C Haze
7:05 PM
3/12/2025

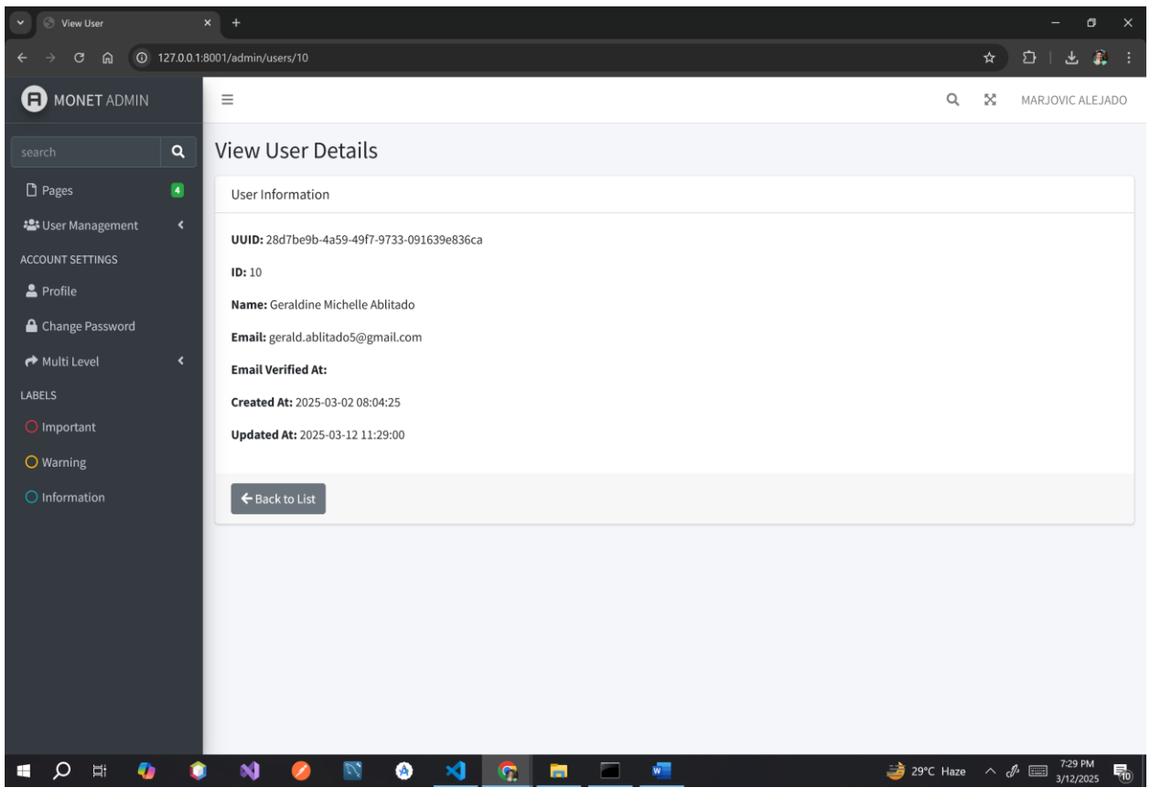

Create User
Not secure
192.168.1.15:8001/admin/users/create
MONET ADMIN
MARJOVIC ALEJADO
search
Pages
User Management
All Users
Add User
ACCOUNT SETTINGS
Profile
Change Password
Multi Level
LABELS
Important
Warning
Information
Create New User
Name:
Marjovic Godz
Email:
alejadomarjovic123@gmail.com
Password:
Save
Cancel
27°C Haze
8:24 PM
3/12/2025


(871) CODFISH | Grand Bea
Users
Not secure
192.168.1.15:8001/admin/users
MONET ADMIN
search
Pages
User Management
All Users
Add User
ACCOUNT SETTINGS
Profile
Change Password
Multi Level
LABELS
Important
Warning
Information
MARJOVIC ALEJADO
User List
User created successfully!
Add New User
#ID
Name
Email
Actions
1
MARJOVIC ALEJADO
marjovicalejado123@gmail.com
6
Marjovic Alejado
marjovicalejado32@gmail.com
7
Marjovic Alejado
marjovicalejad02332@gmail.com
8
Aslainie Maruhom
maruhomaslainie@gmail.com
9
Aslainie Maruhom
aslainiemaruhom8@gmail.com
10
Geraldine Michelle Ablitado
gerald.ablitado5@gmail.com
View
Edit
Delete
27°C Haze
8:25 PM
3/12/2025

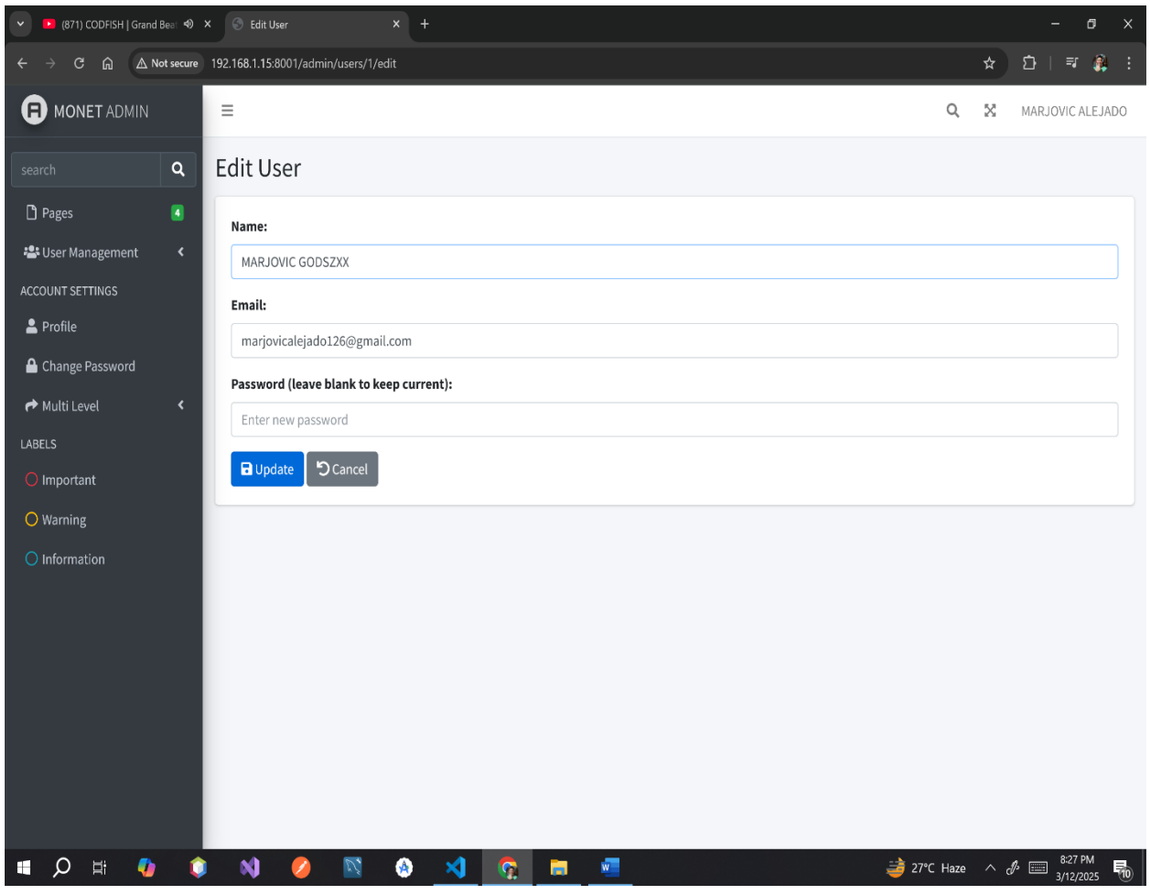

(871) CODFISH | Grand Bea
Users
Not secure
192.168.1.15:8001/admin/users
36
Marjovic
alejadomarjocic+58@gmail.com
37
Marjovic Alejado
alejadomarjocic+1426@gmail.com
38
Marjovic Alejado
alejadomarjocic+14264@gmail.com
39
aslainie
aslainie@gmail.com
40
FAHAD
GWAPO123@GMAIL.COM
41
MarjovicSad
alejadomarjocic123@gmail.com
42
Marjovic God
marjovicalejado256@gmail.com
43
Marjovic Alejado
alejadomarjocic444@gmail.com
44
Marjovic Godz
alejadomarjovic123@gmail.com
View
Edit
Delete
27°C Haze
8:25 PM
3/12/2025

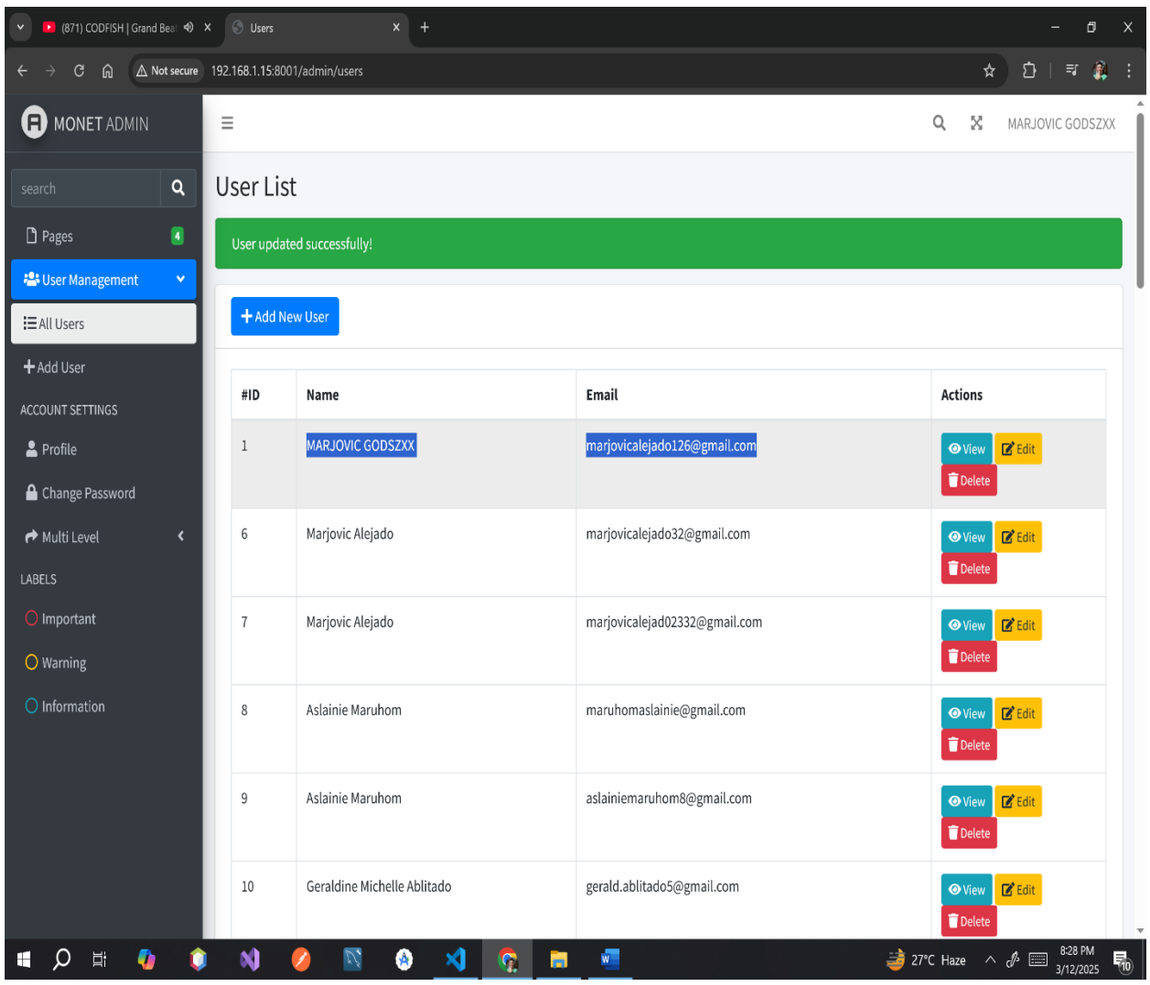
READ USER


View User
127.0.0.1:8001/admin/users/43
MONET ADMIN
MARJOVIC ALEJADO
search
Pages
User Management
ACCOUNT SETTINGS
Profile
Change Password
Multi Level
LABELS
Important
Warning
Information
View User Details
User Information
UUID: 78e95863-a2a5-4164-9bf0-8a5884f93a9b
ID: 43
Name: Marjovic Alejado
Email: alejadomarjocic444@gmail.com
Email Verified At:
Created At: 2025-03-12 11:10:26
Updated At: 2025-03-12 11:10:26
Back to List
29°C Haze
7:28 PM
3/12/2025

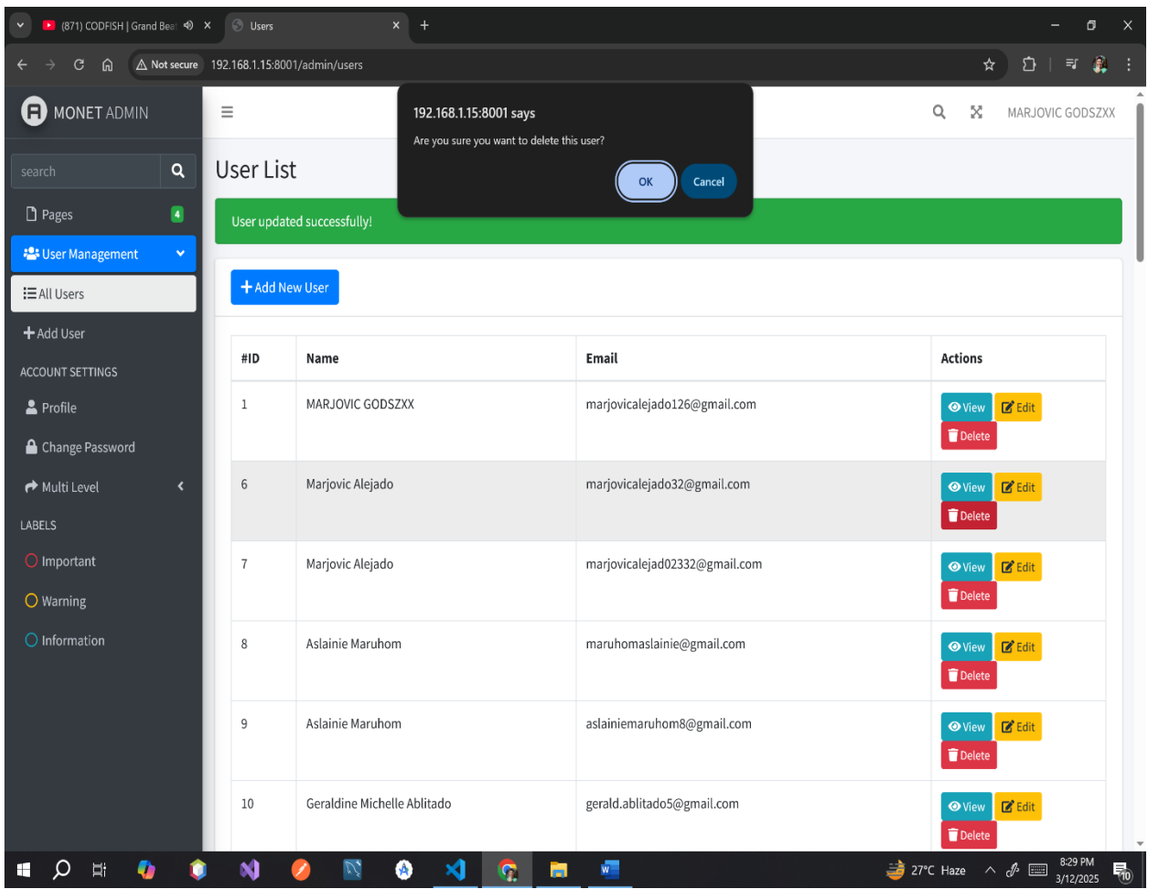

View User
127.0.0.1:8001/admin/users/10
MONET ADMIN
MARJOVIC ALEJADO
search
Pages
User Management
ACCOUNT SETTINGS
Profile
Change Password
Multi Level
LABELS
Important
Warning
Information
View User Details
User Information
UUID: 28d7be9b-4a59-49f7-9733-091639e836ca
ID: 10
Name: Geraldine Michelle Ablitado
Email: gerald.ablitado5@gmail.com
Email Verified At:
Created At: 2025-03-02 08:04:25
Updated At: 2025-03-12 11:29:00
Back to List
29°C Haze
7:29 PM
3/12/2025

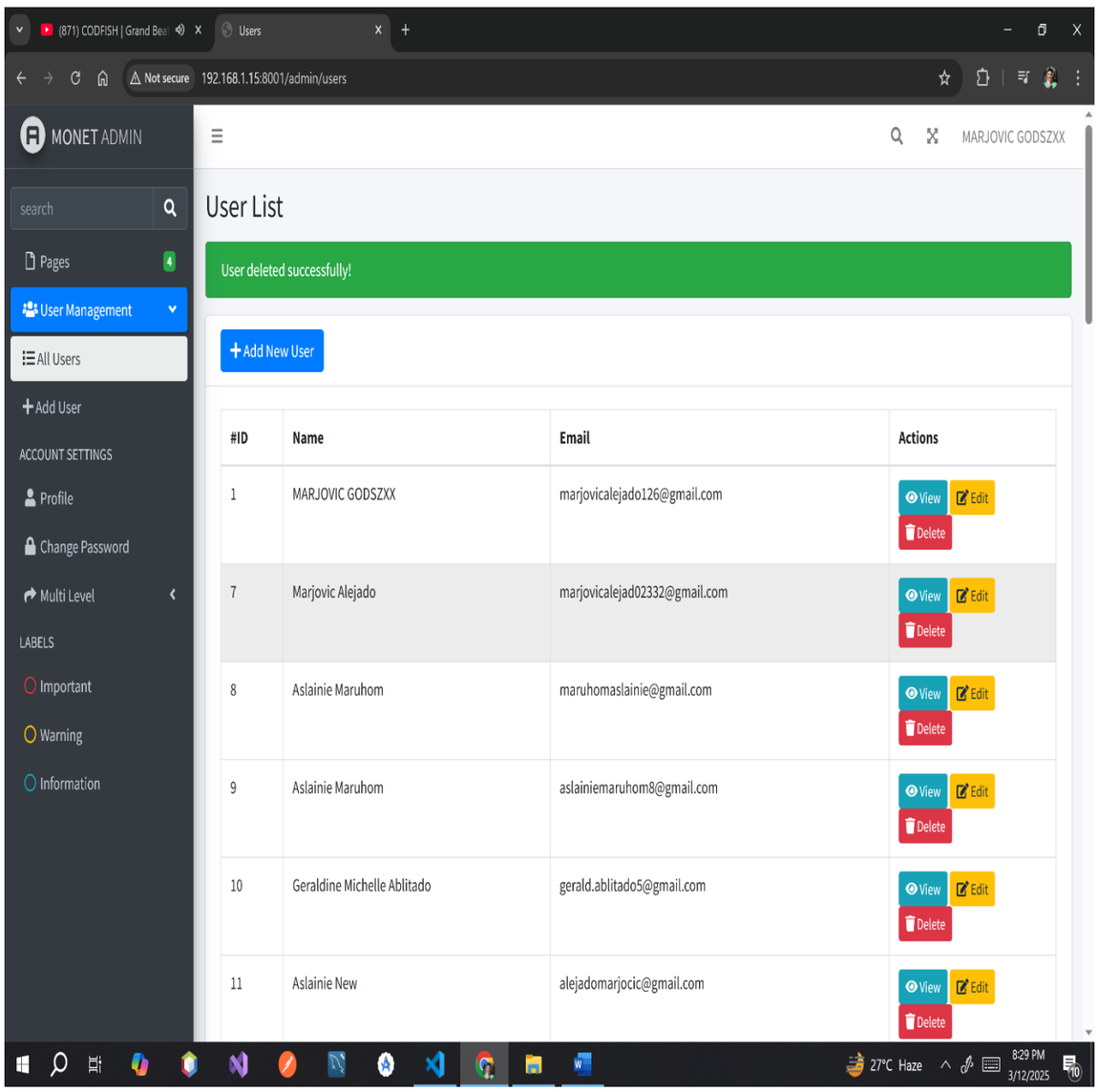
UPDATE USER


(871) CODFISH | Grand Bea
Edit User
Not secure
192.168.1.15:8001/admin/users/1/edit
MONET ADMIN
MARJOVIC ALEJADO
search
Pages
User Management
ACCOUNT SETTINGS
Profile
Change Password
Multi Level
LABELS
Important
Warning
Information
Edit User
Name:
MARJOVIC GODSZXX
Email:
marjovicalejado126@gmail.com
Password (leave blank to keep current):
Enter new password
Update
Cancel
27°C Haze
8:27 PM
3/12/2025

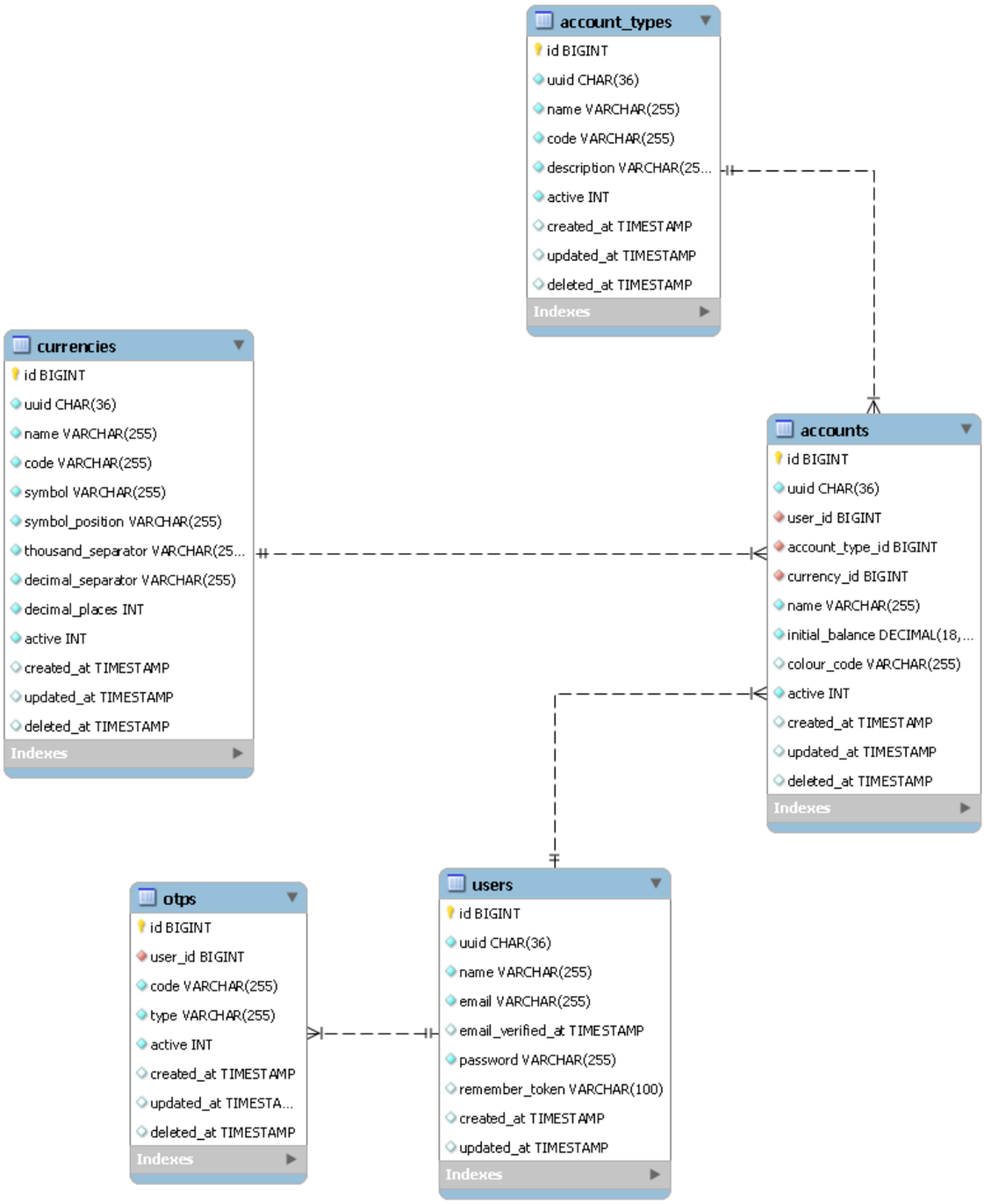

(871) CODFISH | Grand Bea
Users
Not secure
192.168.1.15:8001/admin/users
MONET ADMIN
MARJOVIC GODSZXX
search
Pages
User Management
All Users
Add User
ACCOUNT SETTINGS
Profile
Change Password
Multi Level
LABELS
Important
Warning
Information
User List
User updated successfully!
Add New User
#ID
Name
Email
Actions
1
MARJOVIC GODSZXX
marjovicalejado126@gmail.com
View
Edit
Delete
6
Marjovic Alejado
marjovicalejado32@gmail.com
7
Marjovic Alejado
marjovicalejad02332@gmail.com
8
Aslainie Maruhom
maruhomaslainie@gmail.com
9
Aslainie Maruhom
aslainiemaruhom8@gmail.com
10
Geraldine Michelle Ablitado
gerald.ablitado5@gmail.com
27°C Haze
8:28 PM
3/12/2025

DELETE USER


192.168.1.15:8001 says
Are you sure you want to delete this user?
OK
Cancel
MONET ADMIN
MARJOVIC GODSZXX
User List
User updated successfully!
Add New User
#ID
Name
Email
Actions
1
MARJOVIC GODSZXX
marjovicalejado126@gmail.com
6
Marjovic Alejado
marjovicalejado32@gmail.com
7
Marjovic Alejado
marjovicalejad02332@gmail.com
8
Aslainie Maruhom
maruhomaslainie@gmail.com
9
Aslainie Maruhom
aslainiemaruhom8@gmail.com
10
Geraldine Michelle Ablitado
gerald.ablitado5@gmail.com
View
Edit
Delete


MONET ADMIN
MARJOVIC GODSZXX
User List
User deleted successfully!
Add New User
#ID
Name
Email
Actions
1
MARJOVIC GODSZXX
marjovicalejado126@gmail.com
7
Marjovic Alejado
marjovicalejad02332@gmail.com
8
Aslainie Maruhom
maruhomaslainie@gmail.com
9
Aslainie Maruhom
aslainiemaruhom8@gmail.com
10
Geraldine Michelle Ablitado
gerald.ablitado5@gmail.com
11
Aslainie New
alejadomarjocic@gmail.com
View
Edit
Delete

DATABASE SCHEMA DIAGRAM



CODE SNIPPETS

UserController

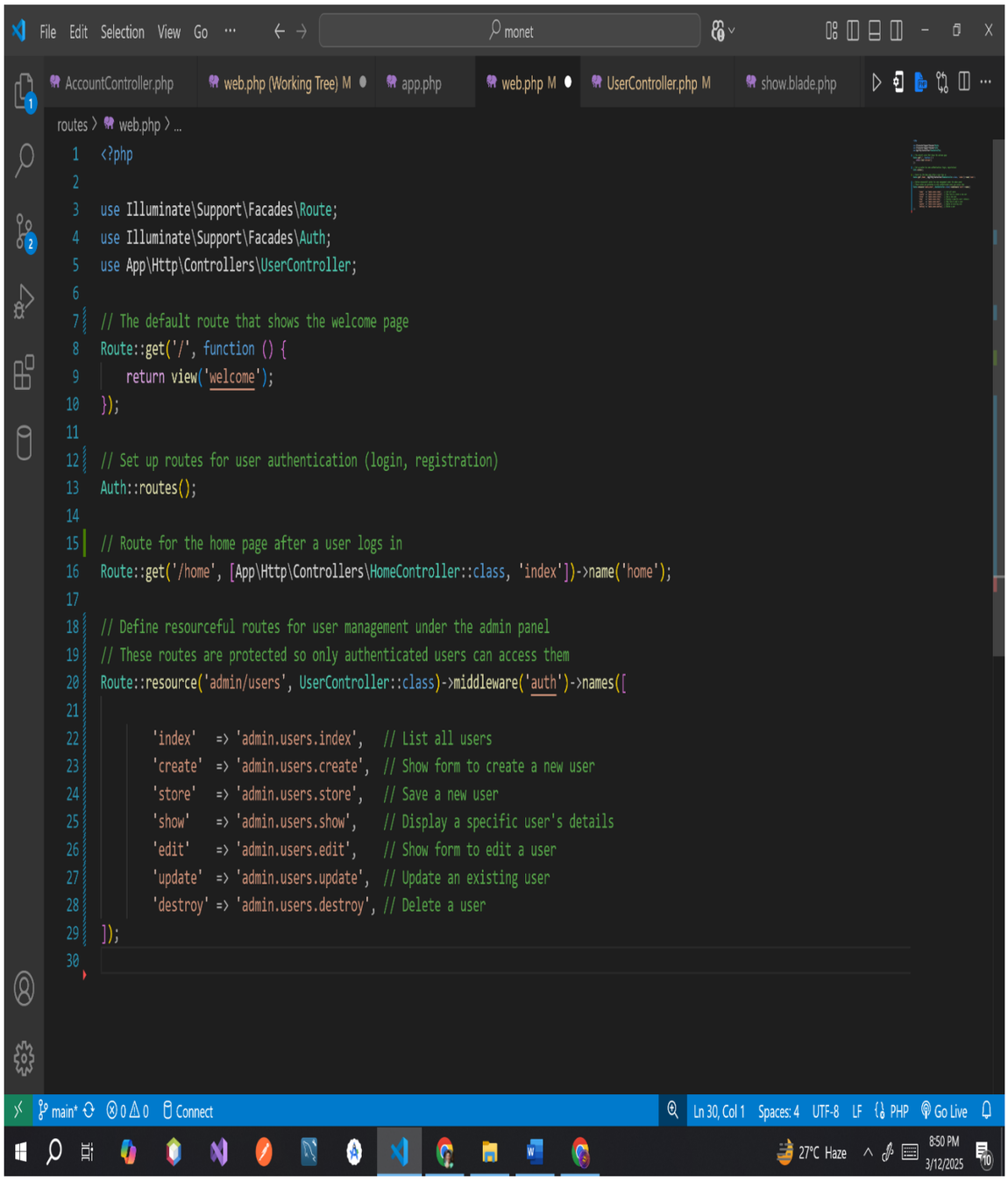
AccountController.php | web.php (Working Tree) | app.php | web.php

app > Http > Controllers > UserController.php > UserController

```php
<?php

namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Http\Request;

class UserController extends Controller
{
    // Only allow logged-in users to access these actions.
    public function __construct()
    {
        $this->middleware('auth');
    }

    // List all users.
    public function index()
    {
        // Get every user from the database.
        $all = User::all();

        // Show the users list in the AdminLTE-styled view.
        return view('admin.users.index', compact('all'));
    }

    // Show details for a single user.
    public function show(User $user)
    {
        // Display a view with this user's details.
        return view('admin.users.show', compact('user'));
    }

    // Display the form to create a new user.
    public function create()
    {
        // Show the form for adding a new user.
        return view('admin.users.create');
    }
```

```php

    // Save a new user to the database.
    public function store(Request $request)
    {
        // Check that the input data is valid.
        $request->validate([
            'name'     => 'required|string|max:255',
            'email'    => 'required|string|email|unique:users,email',
            'password' => 'required|string|min:6',
        ]);

        // Add a new user using the validated data.
        User::create([
            'name'     => $request->name,
            'email'    => $request->email,
            'password' => bcrypt($request->password),
        ]);

        // Redirect to the user list with a success message.
        return redirect()->route('admin.users.index')->with('success', 'User created successfully!');
    }

    // Display the form to edit an existing user.
    public function edit(User $user)
    {
        // Show the form pre-filled with the user's current data.
        return view('admin.users.edit', compact('user'));
    }

    // Update an existing user's data.
    public function update(Request $request, User $user)
    {
        // Check that the updated data is valid.
        $request->validate([
            'name'     => 'required|string|max:255',
            'email'    => 'required|string|email|unique:users,email,'.$user->id,
            'password' => 'nullable|string|min:6',
        ]);

        // Update the user's name and email.
        $user->name  = $request->name;
        $user->email = $request->email;

        // If a new password is provided, update it.
        if ($request->password) {
            $user->password = bcrypt($request->password);
        }
        $user->save();

        // Redirect to the user list with a success message.
        return redirect()->route('admin.users.index')->with('success', 'User updated successfully!');
    }

    // Delete a user.
    public function destroy(User $user)
    {
        // Remove the user from the database.
        $user->delete();

        // Redirect back to the user list with a success message.
        return redirect()->route('admin.users.index')->with('success', 'User deleted successfully!');
    }
}

```

Web Routes


File Edit Selection View Go ···
monet
AccountController.php
web.php (Working Tree) M
app.php
web.php M
UserController.php M
show.blade.php
routes > web.php > ...
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use Illuminate\Support\Facades\Auth;
5 use App\Http\Controllers\UserController;
6
7 // The default route that shows the welcome page
8 Route::get('/', function () {
9 return view('welcome');
10 });
11
12 // Set up routes for user authentication (login, registration)
13 Auth::routes();
14
15 // Route for the home page after a user logs in
16 Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])->name('home');
17
18 // Define resourceful routes for user management under the admin panel
19 // These routes are protected so only authenticated users can access them
20 Route::resource('admin/users', UserController::class)->middleware('auth')->names([
21
22 'index' => 'admin.users.index', // List all users
23 'create' => 'admin.users.create', // Show form to create a new user
24 'store' => 'admin.users.store', // Save a new user
25 'show' => 'admin.users.show', // Display a specific user's details
26 'edit' => 'admin.users.edit', // Show form to edit a user
27 'update' => 'admin.users.update', // Update an existing user
28 'destroy' => 'admin.users.destroy', // Delete a user
29]);
30
main* 0 0 Connect
Ln 30, Col 1 Spaces: 4 UTF-8 LF PHP Go Live
27°C Haze
8:50 PM
3/12/2025

Index page

index.blade.php ● | create.blade.php | edit.blade.php | adminlte.php | User.php

resources > views > admin > users > index.blade.php

```blade
@extends('adminlte::page')

// Set the page title
@section('title', 'Users')

// Define the header section of the page
@section('content_header')
    // Display the main heading
    <h1>User List</h1>
@endsection

// Define the main content section
@section('content')
    // Check if there is a success message in the session and show it
    @if(session('success'))
        <div class="alert alert-success">{{ session('success') }}</div>
    @endif

    // Start a card component to hold the user table
    <div class="card">
        <div class="card-header">
            // Button to add a new user
            <a href="{{ route('admin.users.create') }}" class="btn btn-primary">
                <i class="fas fa-plus"></i> Add New User
            </a>
        </div>
        <div class="card-body">
            // Start a table to list users
            <table class="table table-bordered table-hover">
                <thead>
                    <tr>
                        // Table header for the user ID
                        <th>#ID</th>
                        // Table header for the user name
                        <th>Name</th>
                        // Table header for the user email
                        <th>Email</th>
                        // Table header for action buttons with fixed width
                        <th width="220">Actions</th>
                    </tr>
                </thead>
                <tbody>
```

```
                    // Loop through each user in the collection
                    @foreach($all as $user)
                        <tr>
                            // Show the user's ID
                            <td>{{ $user->id }}</td>
                            // Show the user's name
                            <td>{{ $user->name }}</td>
                            // Show the user's email
                            <td>{{ $user->email }}</td>
                            <td>
                                // Link to view the user's details
                                <a href="{{ route('admin.users.show', $user->id) }}" class="btn btn-sm btn-info">
                                    <i class="fas fa-eye"></i> View
                                </a>

                                // Link to edit the user's info
                                <a href="{{ route('admin.users.edit', $user->id) }}" class="btn btn-sm btn-warning">
                                    <i class="fas fa-edit"></i> Edit
                                </a>

                                // Form to delete the user
                                <form action="{{ route('admin.users.destroy', $user->id) }}" method="POST" style="display: inline-block;">
                                    @csrf
                                    @method('DELETE')
                                    // Delete button with a confirmation prompt
                                    <button class="btn btn-sm btn-danger" onclick="return confirm('Are you sure you want to delete this user?')">
                                        <i class="fas fa-trash"></i> Delete
                                    </button>
                                </form>
                            </td>
                        </tr>
                    @endforeach
                </tbody>
            </table>
        </div>
    </div>
@endsection

```

Show page

show.blade.php M | 0001_01_01_000000_create_users_table.php | index.blade.php | create.blade.php

resources > views > admin > users > show.blade.php

```blade
@extends('adminlte::page')

// Set the page title
@section('title', 'View User')

// Define the header section
@section('content_header')
    // Display the main heading for user details
    <h1>View User Details</h1>
@endsection

// Define the main content section
@section('content')
    // Start a card to show user information
    <div class="card">
        <div class="card-header">
            // Card title for user information
            <h3 class="card-title">User Information</h3>
        </div>
        <div class="card-body">
            // Show the user's UUID
            <p><strong>UUID:</strong> {{ $user->uuid }}</p>

            // Show the user's ID
            <p><strong>ID:</strong> {{ $user->id }}</p>

            // Show the user's name
            <p><strong>Name:</strong> {{ $user->name }}</p>

            // Show the user's email
            <p><strong>Email:</strong> {{ $user->email }}</p>

            // Show the date the user's email was verified
            <p><strong>Email Verified At:</strong> {{ $user->email_verified_at }}</p>

            // Show the date the user was created
            <p><strong>Created At:</strong> {{ $user->created_at }}</p>

            // Show the date the user was last updated
            <p><strong>Updated At:</strong> {{ $user->updated_at }}</p>
        </div>
        <div class="card-footer">
            // Link to go back to the user list
            <a href="{{ route('admin.users.index') }}" class="btn btn-secondary">
                <i class="fas fa-arrow-left"></i> Back to List
            </a>
        </div>
    </div>
@endsection

```

Edit page

show.blade.php M | 0001_01_01_000000_create_users_table.php | index.blade.php | create.blade.php | edit.b

resources > views > admin > users > edit.blade.php

```
@extends('adminlte::page')

@section('title', 'Edit User')

@section('content_header')
    // Display the page header

    <h1>Edit User</h1>
@endsection

@section('content')
    <div class="card">
        <div class="card-body">

            // Check if there are any errors
            @if ($errors->any())
                <div class="alert alert-danger">
                    <strong>Whoops!</strong> Please correct the following errors.<br><br>
                    <ul>
                        @foreach ($errors->all() as $error)
                            <li>{{ $error }}</li>
                        @endforeach
                    </ul>
                </div>
            @endif

            // Form to update the user
            <form action="{{ route('admin.users.update', $user->id) }}" method="POST">
                @csrf
                @method('PUT')

                // Input field for the user's name
                <div class="form-group">
                    <label for="name">Name:</label>
                    <input type="text" name="name"
                           class="form-control"
                           value="{{ old('name', $user->name) }}"
                           placeholder="Enter name">
                </div>

```

```
40
41                  // Input field for the user's email
42                  <div class="form-group">
43                      <label for="email">Email:</label>
44                      <input type="email" name="email"
45                             class="form-control"
46                             value="{{ old('email', $user->email) }}"
47                             placeholder="Enter email">
48                  </div>
49
50                  // Input field for the user's password (leave blank to keep current)
51                  <div class="form-group">
52                      <label for="password">Password (leave blank to keep current):</label>
53                      <input type="password" name="password"
54                             class="form-control"
55                             placeholder="Enter new password">
56                  </div>
57
58                  // Submit button to update the user
59                  <button type="submit" class="btn btn-primary">
60                      <i class="fas fa-save"></i> Update
61                  </button>
62
63                  // Cancel button to go back to the user list
64                  <a href="{{ route('admin.users.index') }}" class="btn btn-secondary">
65                      <i class="fas fa-undo"></i> Cancel
66                  </a>
67              </form>
68          </div>
69      </div>
70  @endsection
71
```

Create page

show.blade.php M | 0001_01_01_000000_create_users_table.php | index.blade.php | create.blade.php M

resources > views > admin > users > create.blade.php

```
@extends('adminlte::page')

// Set the page title
@section('title', 'Create User')

// Define the header section
@section('content_header')
    // Display the main header for the page
    <h1>Create New User</h1>
@endsection

// Define the main content section
@section('content')
    <div class="card">
        <div class="card-body">

            // If there are any errors, show them
            @if ($errors->any())
                <div class="alert alert-danger">
                    <strong>Whoops!</strong> Please correct the following errors<br><br>
                    <ul>
                        @foreach ($errors->all() as $error)
                            <li>{{ $error }}</li>
                        @endforeach
                    </ul>
                </div>
            @endif

            // Form to create a new user
            <form action="{{ route('admin.users.store') }}" method="POST">
                @csrf

                // Form group for the user's name
                <div class="form-group">
                    <label for="name">Name:</label>
                    <input type="text" name="name"
                           class="form-control"
                           value="{{ old('name') }}"
                           placeholder="Enter name">
                </div>

```

```
                // Form group for the user's email
                <div class="form-group">
                    <label for="email">Email:</label>
                    <input type="email" name="email"
                           class="form-control"
                           value="{{ old('email') }}"
                           placeholder="Enter email">
                </div>

                // Form group for the user's password
                <div class="form-group">
                    <label for="password">Password:</label>
                    <input type="password" name="password"
                           class="form-control"
                           placeholder="Enter password">
                </div>

                // Button to submit the form and save the user
                <button type="submit" class="btn btn-success">
                    <i class="fas fa-save"></i> Save
                </button>

                // Button to cancel and go back to the user list
                <a href="{{ route('admin.users.index') }}" class="btn btn-secondary">
                    <i class="fas fa-undo"></i> Cancel
                </a>
            </form>

        </div>
    </div>
@endsection

```

CHALLENGES AND SOLUTIONS

Challenge 1: Setting Up the Layout

Our Problem:

We needed to include the AdminLTE template in our Laravel project, but linking all the CSS and JavaScript files correctly was a bit challenging.

Our Solution:

We used a package like Laravel-AdminLTE to quickly install and configure the template. This package creates a master layout that we can extend in our Blade files.

Challenge 2: Routing and Controllers

Our Problem:

Creating separate routes for every CRUD action (create, read, update, delete) felt repetitive.

Our Solution:

We leveraged Laravel's resource routes. For example, we added this line in our routes/web.php file:

Route::resource('items', ItemController::class);

Challenge 3: Form Handling and Validation

Our Problem:

We needed to ensure that our user input was valid and free of errors, and writing validation logic in each controller method was cumbersome.

Our Solution:

We took advantage of Laravel's built-in validation. We even used Form Request classes to keep our code clean. For example:

```
$request->validate([
    'name'  => 'required|max:255',
    'email' => 'required|email'
]);
```

Challenge 4: Asset Management with Vite and Sass

Our Problem:

Since we are using Laravel 10 with Vite and Sass, integrating AdminLTE's assets (CSS/JS) didn't work out-of-the-box.

Our Solution:

We made sure to properly link the AdminLTE CSS and JS files in our Vite configuration. This involved updating our vite.config.js to include these files and ensuring our Sass compiles correctly. Reviewing the package's documentation helped us sort out the file paths and dependencies.

Challenge 5: Customizing the UI

Our Problem:

The default AdminLTE design didn’t always fit our needs, and adjusting colors or layouts without clear guidance was difficult.

Our Solution:

We used the customization options provided by AdminLTE. By changing the Sass variables or adding our own styles in our Laravel project, we could easily adjust the design to better suit our requirements.

Challenge 6: Handling JavaScript Plugins

Our Problem:

AdminLTE uses various JavaScript plugins (like charts and data tables), and sometimes these plugins didn’t work well with our Laravel setup if they weren’t loaded correctly.

Our Solution:

We ensured that we imported and compiled all the necessary JavaScript files correctly. By using Vite’s module system to load these files and testing each plugin separately, we made sure everything worked as expected.