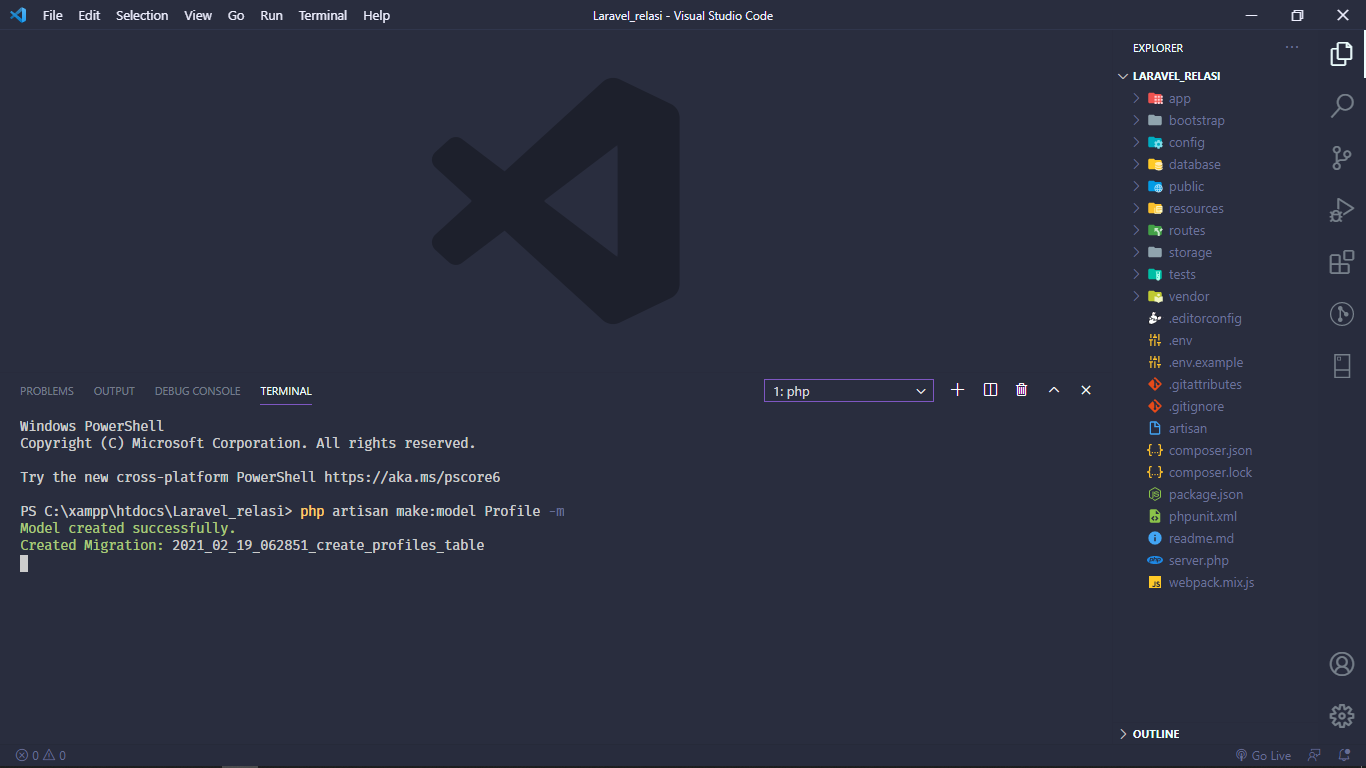
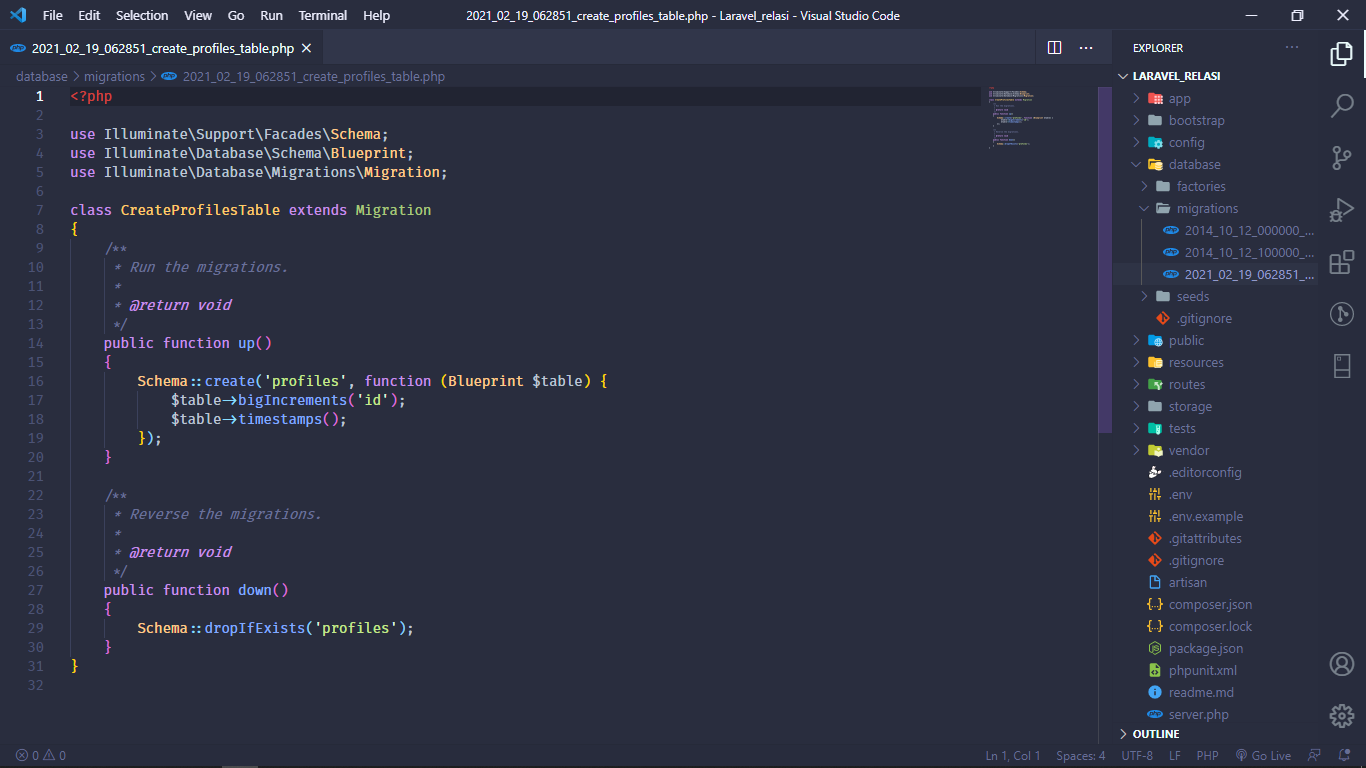
**I. Bagian 1 – Setup Model Relasi One To One**

Pertama, Bukalah terminal dari folder Laravel yang sudah di extract (Visual Studio Code) dengan *shortcut* CTRL + SHIFT + ` , teruntuk kalian yang tidak menggunakan Visual Studio Code bisa menggunakan CMD (Command Prompt).

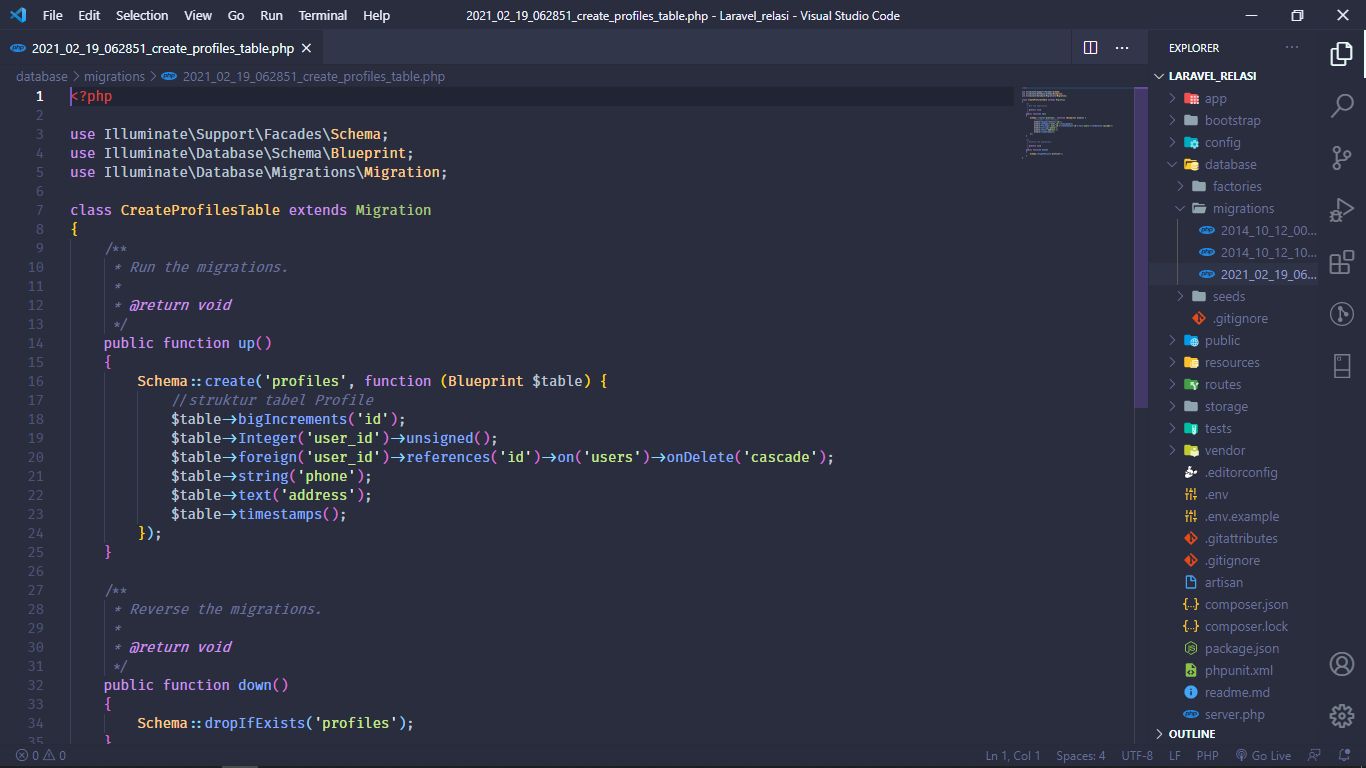


Lalu, ketiklah “php artisan make:model Profile -m” (tanpa symbol “”). Apabila tampilan sudah seperti gambar yang menampilkan “Model Created Successfully” maka model dan tabel blueprint sudah dibuat dan dapat di migrasi.

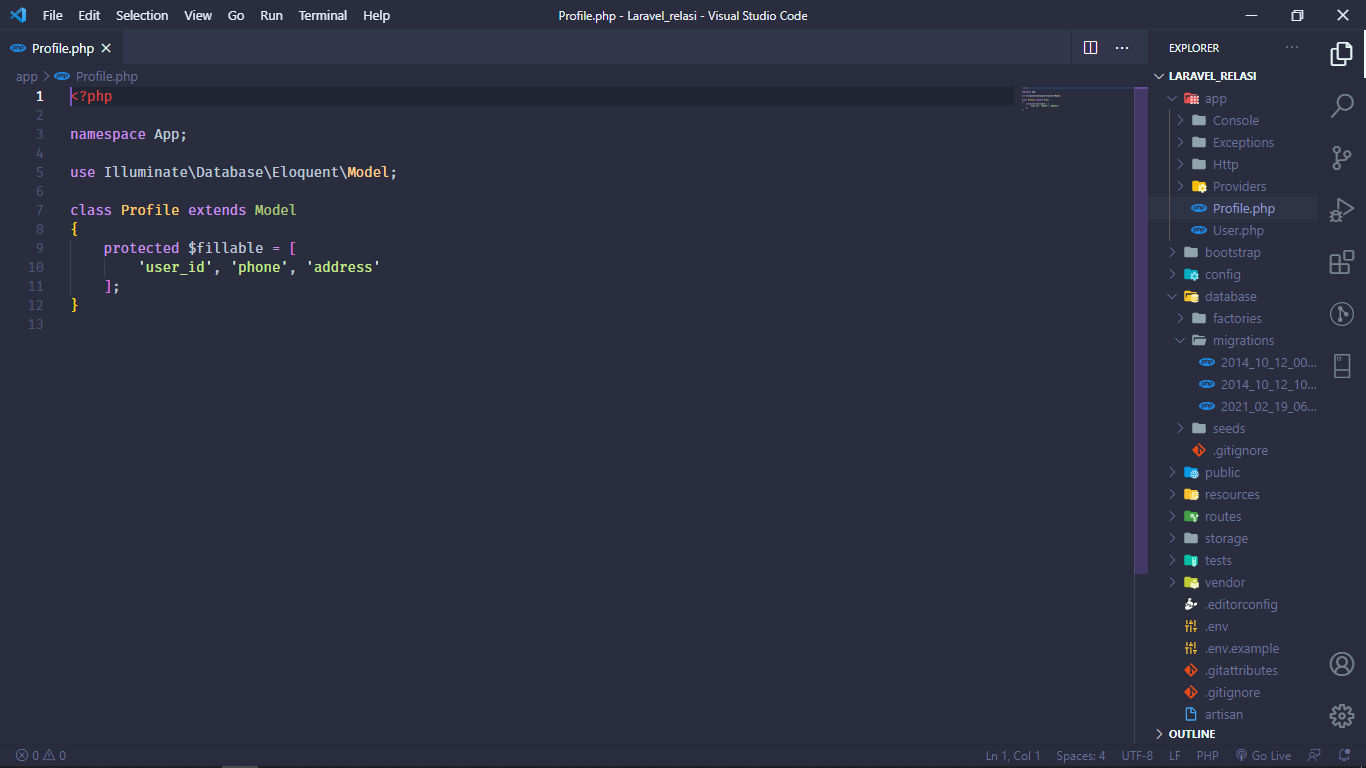
Kemudian, masuk ke folder database > migrations > dan cari file yang bernama xxxx\_xx\_xx\_xxxxxx\_create\_profiles\_table.php seperti gambar berikut ini.



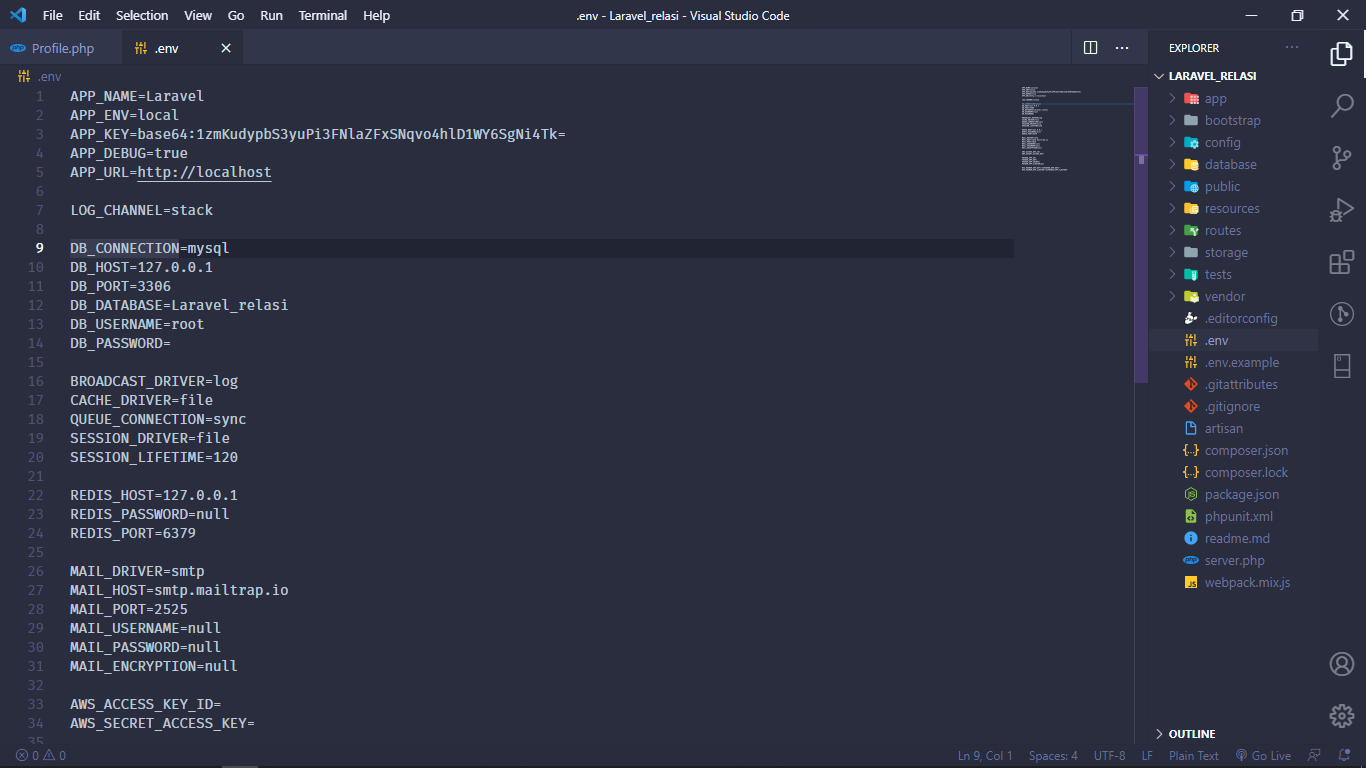
Setelah itu tambahkan struktur tabel seperti, Id User, nomor telepon, dan alamat sebagai identitas User seperti gambar dibawah ini.



Kemudian, masuk ke folder app dengan file yang bernama Profile.php dan tambahkan beberapa properti kedalam class tersebut seperti berikut.



Jika sudah, kalian bisa cari file yang bernama .env untuk mengkoneksikan database kalian dengan projek Laravel yang sedang dibuat.



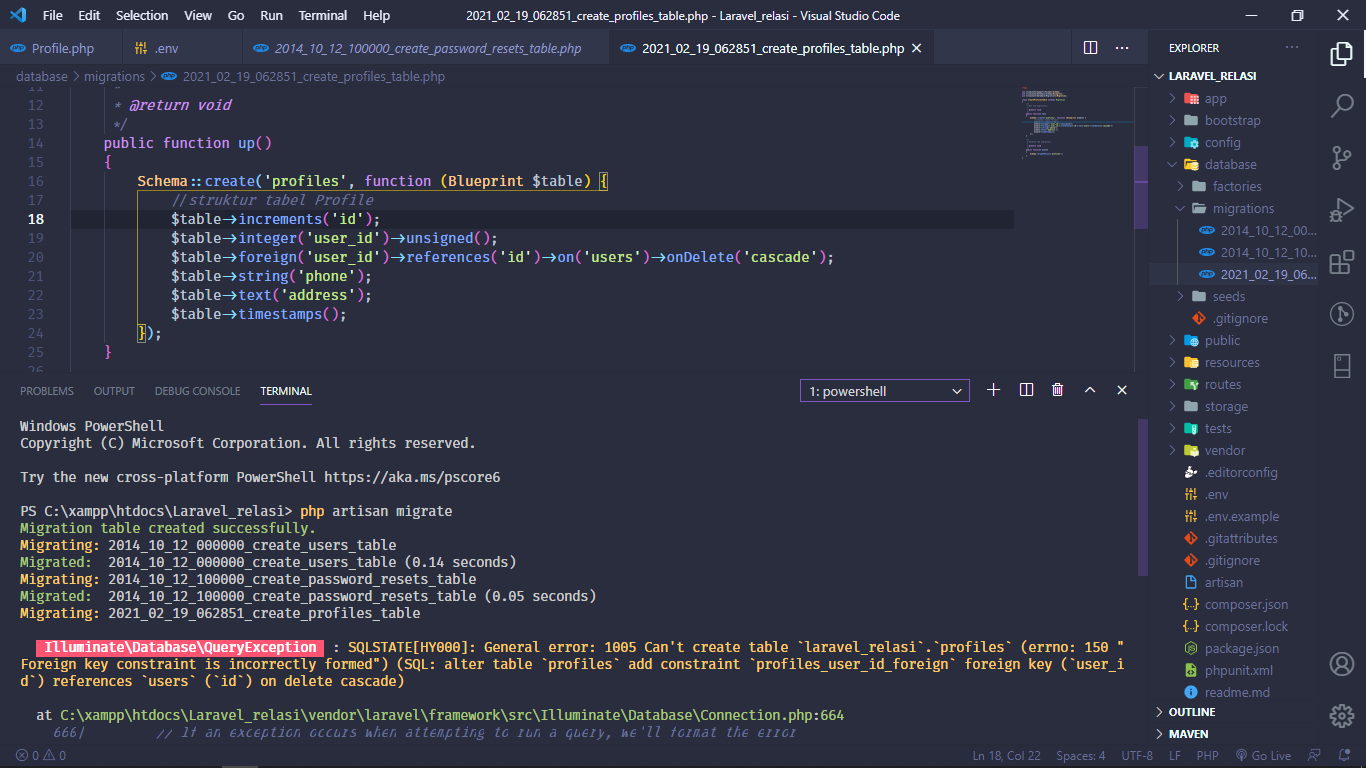
Untuk pengguna MySQL :

DB\_DATABASE, database yang kita gunakan untuk projek,

DB\_USERNAME, username DBMS yang kita gunakan yaitu “root”, untuk cloud server memiliki username tersendiri.

DB\_PASSWORD, password DBMS yang kita gunakan, bisa dikosongkan, dan juga untuk cloud server terkadang terdapat password yang telah didaftarkan.

Jika sudah terdaftar, kalian bisa buka terminal kembali atau bisa menggunakan Command Prompt, dengan mengetikkan “php artisan migrate” (tanpa symbol “”) Pastikan xampp Control Panel telah aktif.



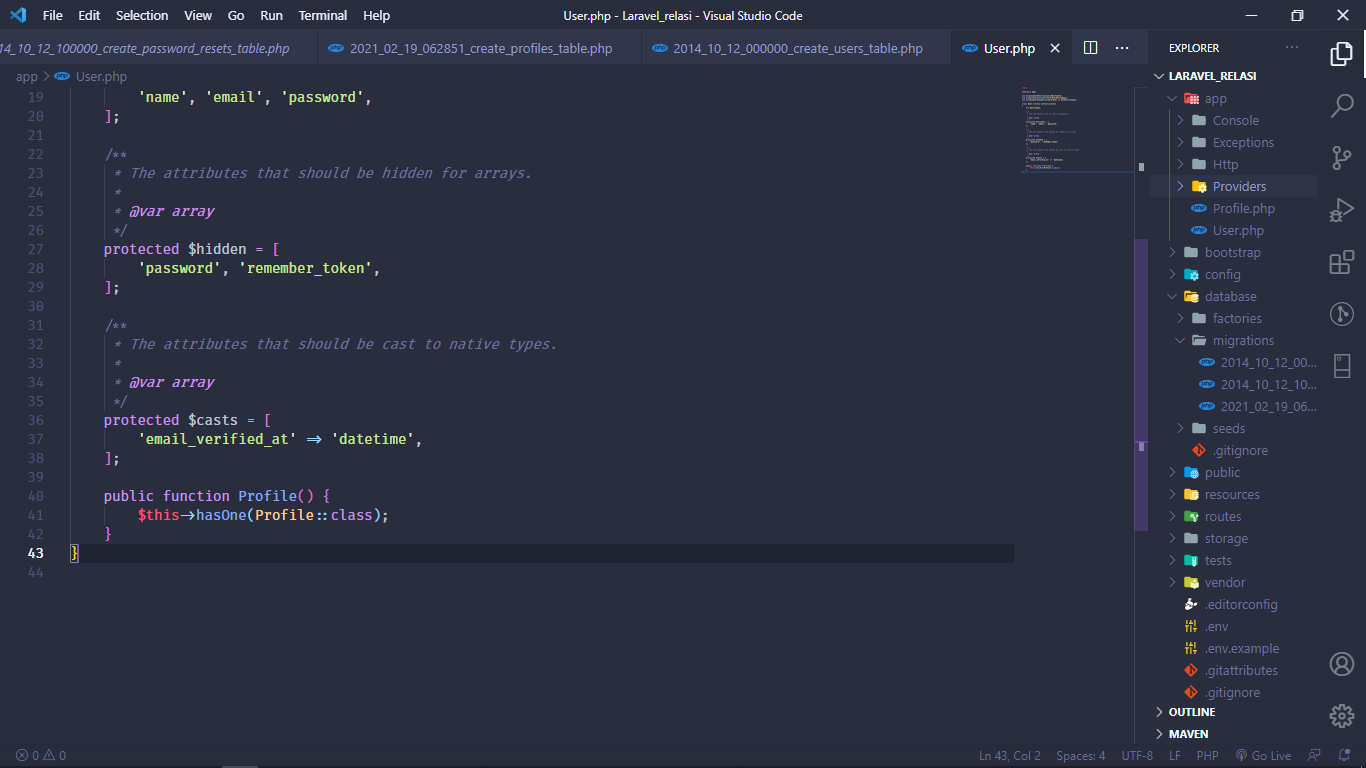
Untuk menkofigurasi relasi One To One terhadap Profile, bukalah file User.php pada folder App dan tambahkan script dibawah ini :

public function Profile() {

        return $this->hasOne(Profile::class);

    }

Sehingga menjadi seperti ini :



**II. Bagian 2 - Membuat Create Data User**

Masuk ke file web.php untuk membuat route pembuatan data user secara manual (tanpa menggunakan *form*) dengan itu, tambahkan script seperti dibawah ini :

use App\User;

Route::get('/create\_user', function() {

    $user = User::create([

        'name' => 'admin',

        'email' => 'admin@gmail.com',

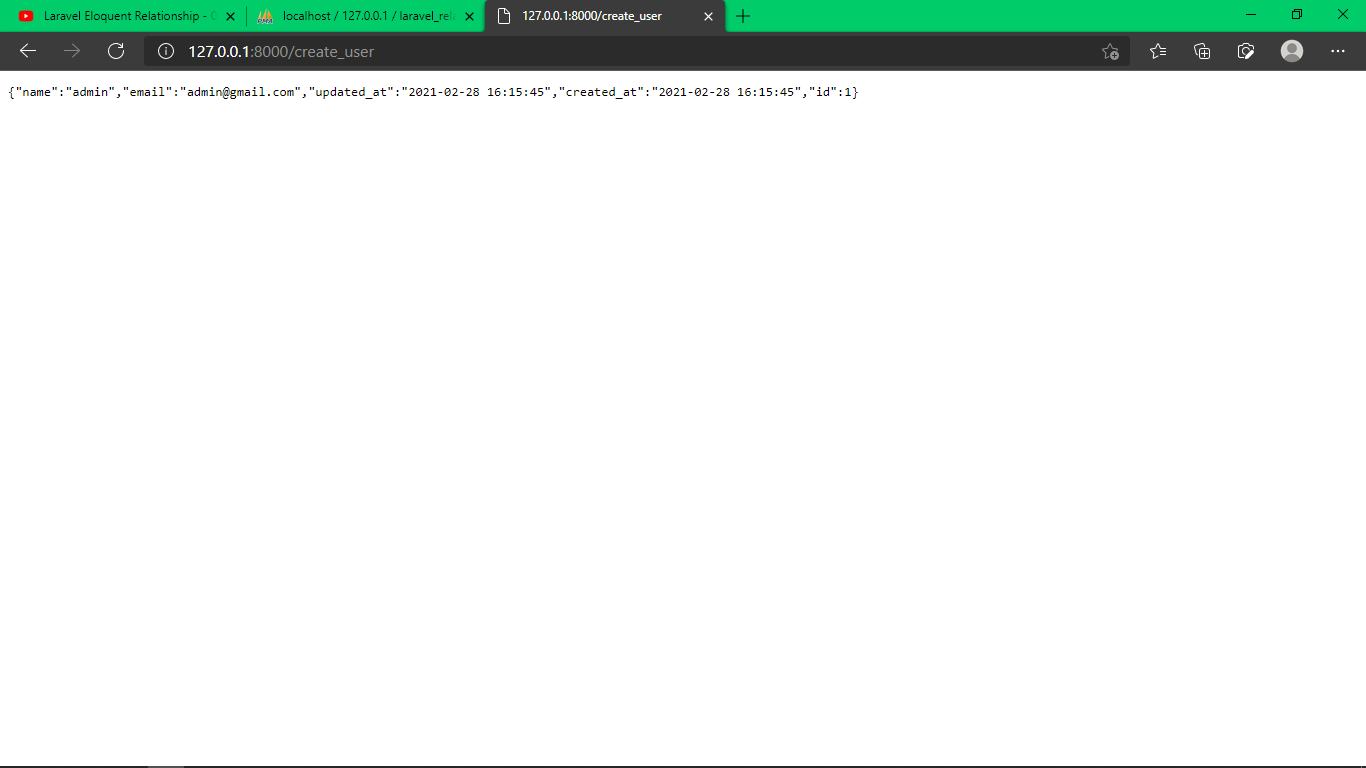
        'password' => bcrypt(12345)

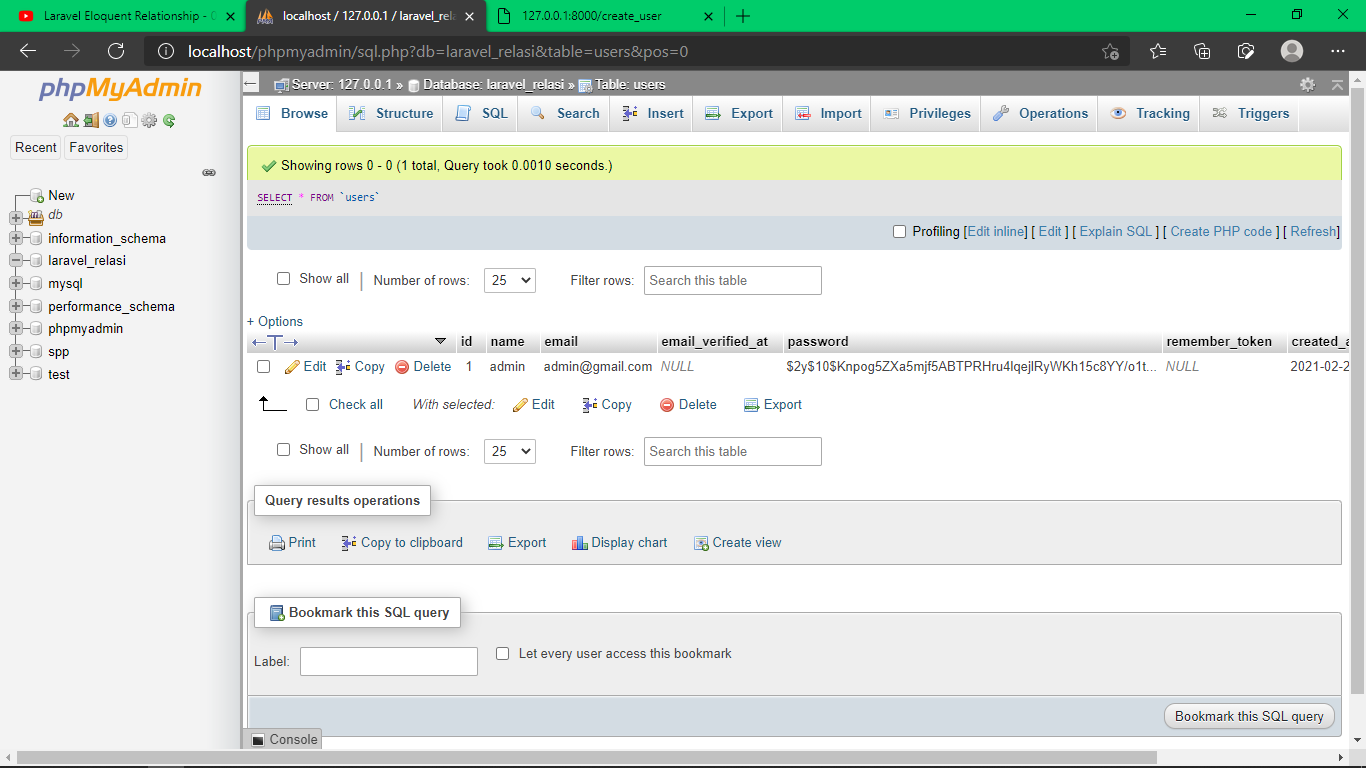
    ]);

    return $user;

});

Setelah itu masuk ke route yang telah dibuat sebelumnya (/create\_user) sampai muncul data-data yang sudah di daftarkan seperti contoh dibawah ini :





Apabila datanya sudah terdaftar pada tabel “User”, lalu sekarang kita bisa tambahkan data untuk tabel “Profile” dengan cara yang sama seperti langkah sebelumnya.

use App\Profile;

Route::get('/create\_profile', function() {

    $profile = Profile::create([

        'user\_id' => 1,

        'phone' => '08921231413',

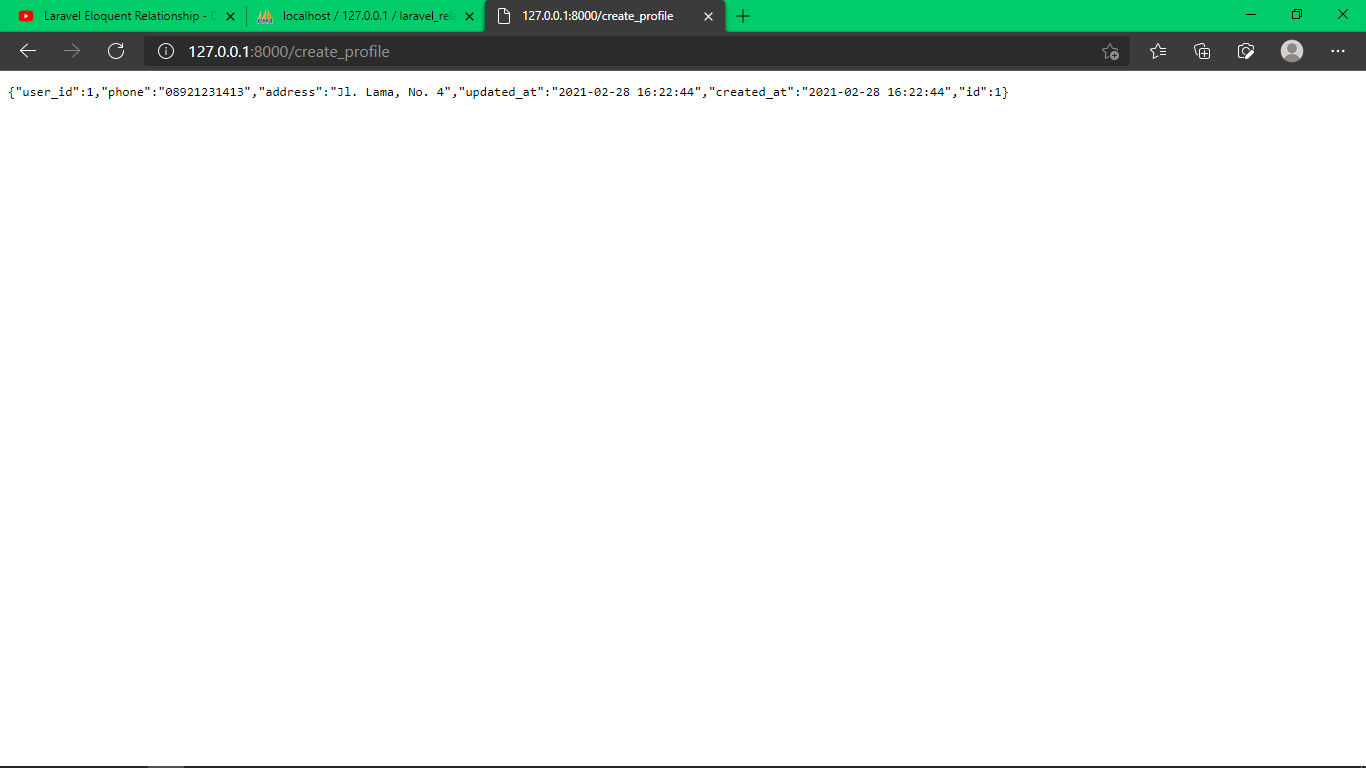
        'address' => 'Jl. Lama, No. 4'

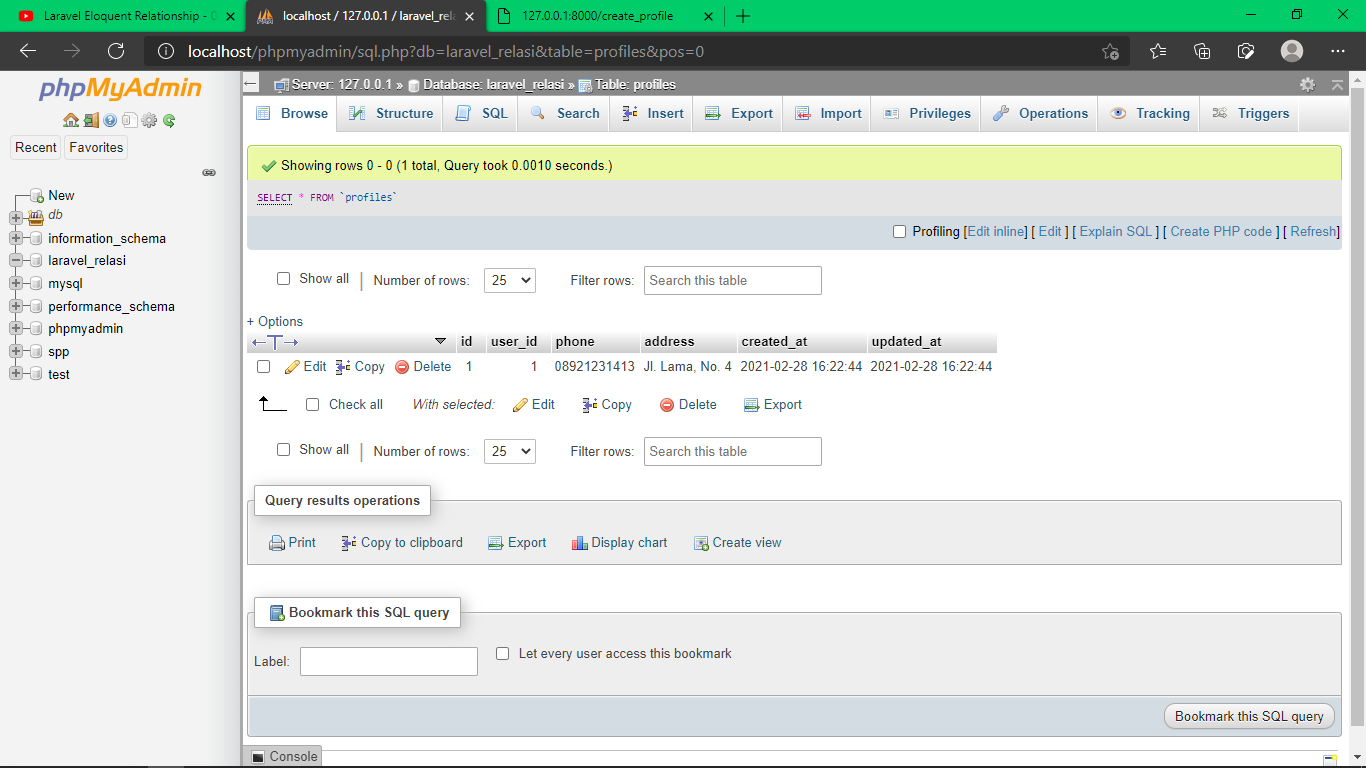
    ]);

    return $profile;

});

Setelah itu, Jalankan route yang telah kita buat (/create\_profile), sampai tampilannya seperti dibawah ini (sesuai data yang telah di daftarkan) :





Kemudian, Tambahkan script dibawah ini dalam file web.php :

Route::get('/create\_user\_profile', function() {

    $user = User::find(2);

    $profile = new Profile([

        'phone' => '0892123124151',

        'address' => 'Jl. Lama, No. 6'

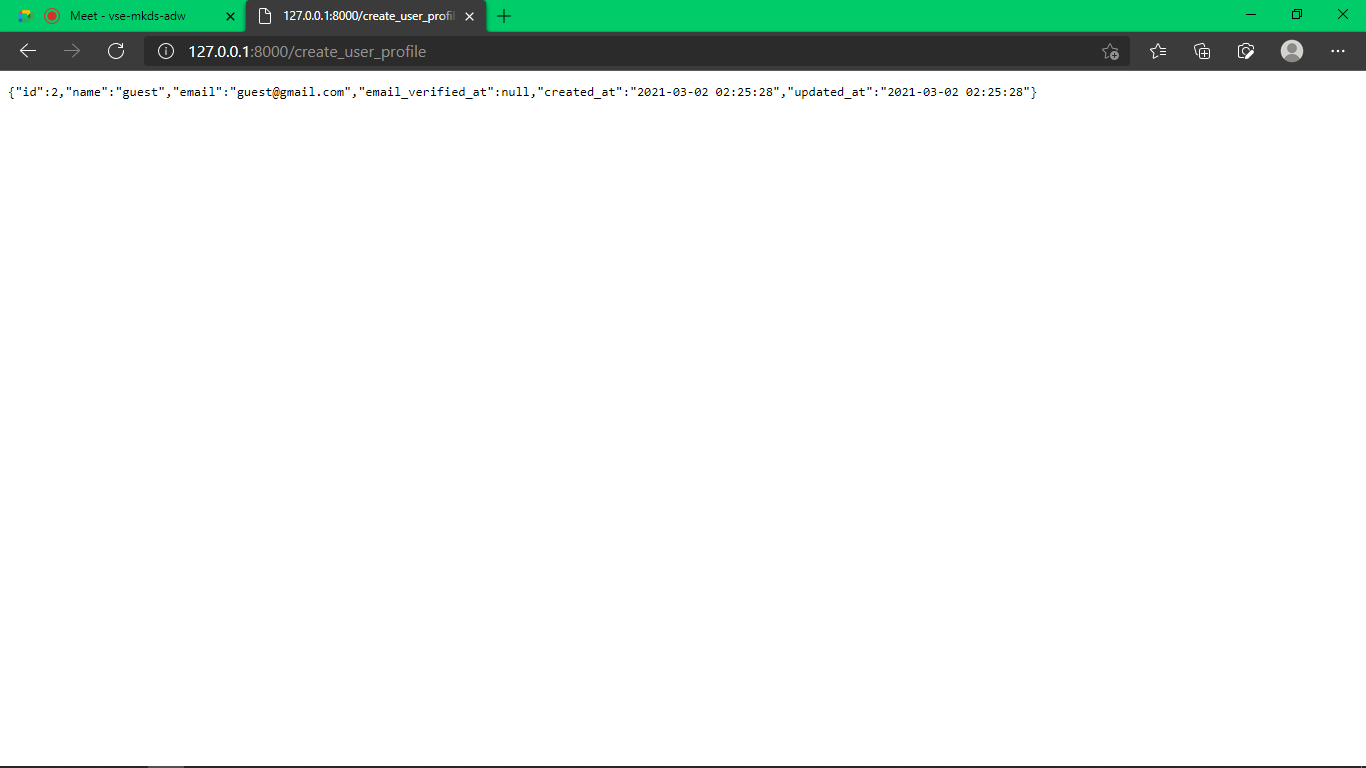
    ]);

    $user->profile()->save($profile);

    return $user;

});

Dan kemudian, jalankan route yang sudah dibuat sebelumnya (/create\_user\_profile) sampai hasilnya seperti gambar dibawah ini :



**III. Bagian 3 - Membuat Read Data User**

Lalu, tambahkan script dibawah ini pada file web.php

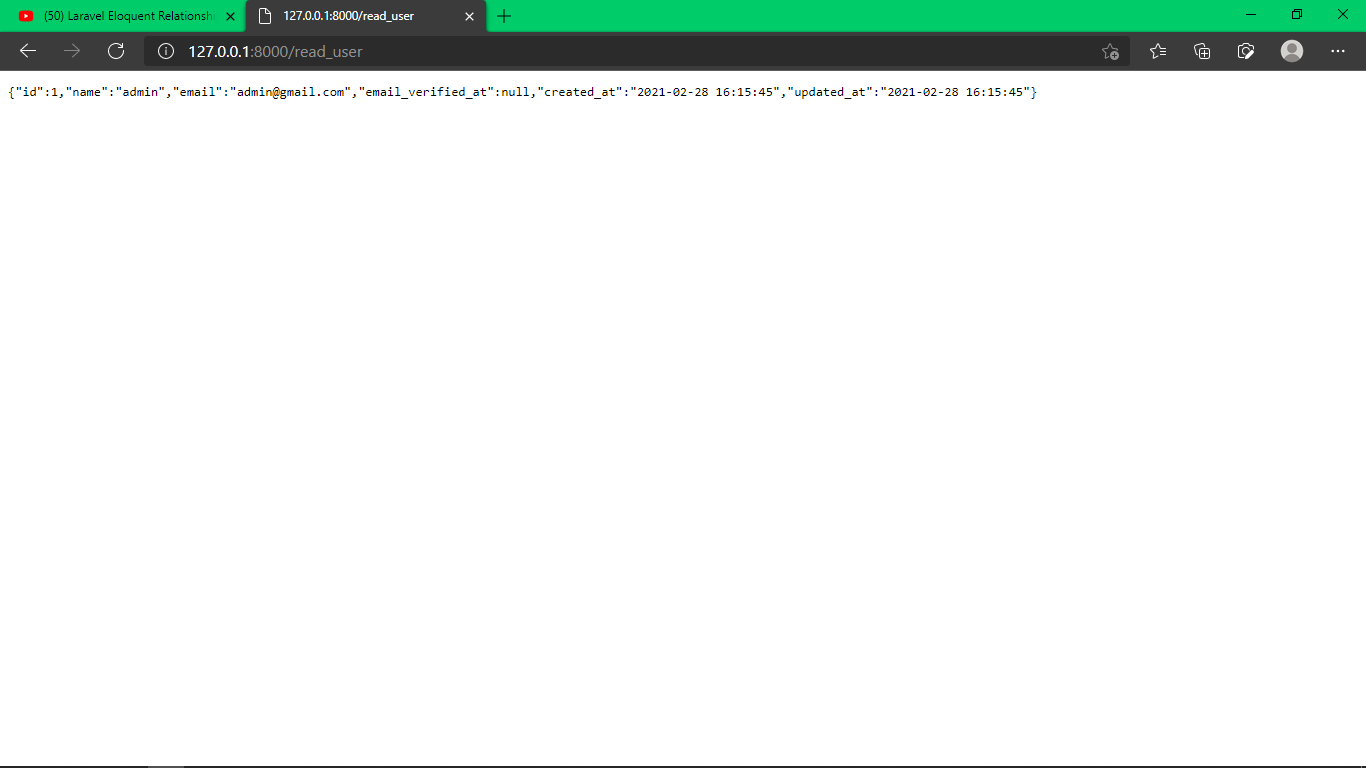
Route::get('/read\_user', function () {

    $user = User::find(1);

    return $user;

});

Sehingga tampilannya saat dijalankan seperti gambar dibawah ini :



Setelah itu, isi dari Route::get(‘/read\_user’, function () diubah menjadi :

Route::get('/read\_user', function () {

    $user = User::find(1);

    $data = [

        'name' => $user->name,

        'phone' => $user->profile->phone,

        'address' => $user->profile->address

    ];

    return $data;

});

Dan tambahkan beberapa script dibawah ini pada file Profile.php (model)

public function user() {

        return $this->belongsTo(User::class);

    }

Setelah itu, kita membuat route baru untuk membaca data-data Profile yang ada pada database dengan menambahkan script dibawah ini pada file web.php :

Route::get('/read\_profile', function () {

    $profile = Profile::where('phone', '0892123124151')->first();

    $data = [

        'name' => $profile->user->name,

        'email' => $profile->user->email,

        'phone' => $profile->phone,

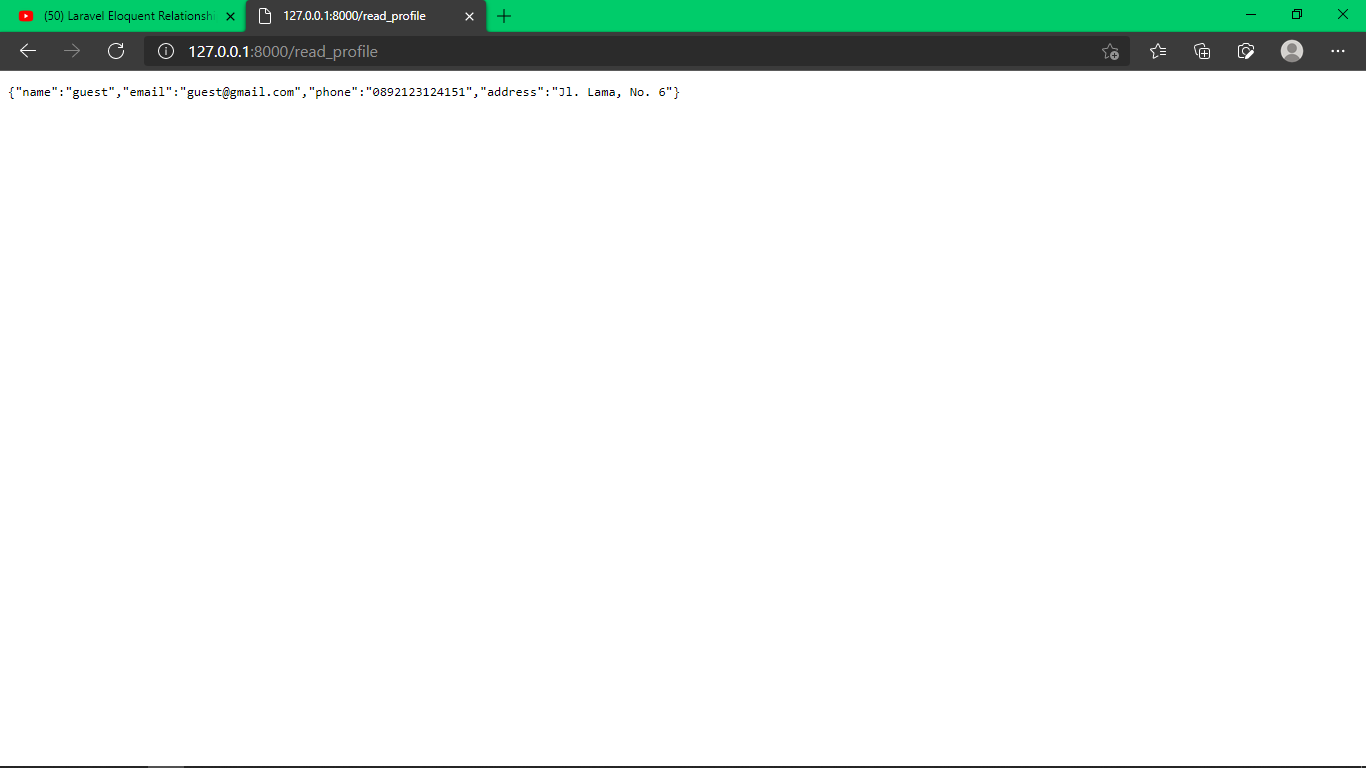
        'address' => $profile->address

    ];

    return $data;

});

Dan saat dijalankan, tampilannya seperti gambar berikut :



**IV. Bagian 4 - Membuat Update Data User**

Pertama, pergi ke file web.php dan tambahkan script dibawah ini untuk membuat route baru :

Route::get('/update\_profile', function () {

    $user = User::find(2);

    $user->profile()->update([

        'phone' => '0895325237',

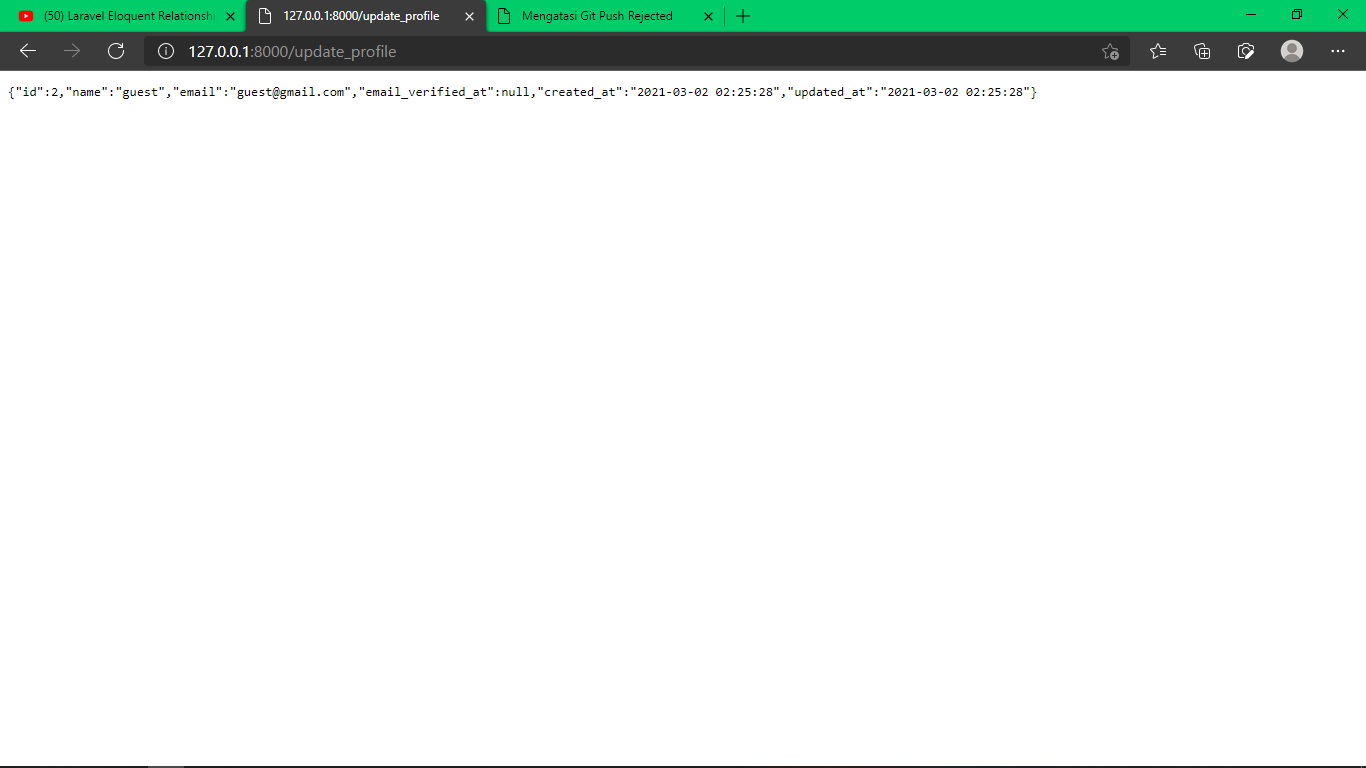
        'address' => 'Jl. Inovasi, No. 23'

    ]);

    return $user;

});

Ketika dijalankan di website, tampilannya akan seperti dibawah ini dan data yang dituju akan terubah sesuai apa yang dituliskan pada web.php



**V. Bagian 5 - Membuat Delete Data User**

Dalam mendelete Data User, kita bisa menambahkan Route yang baru pada web.php seperti script dibawah ini :

Route::get('/delete\_profile', function () {

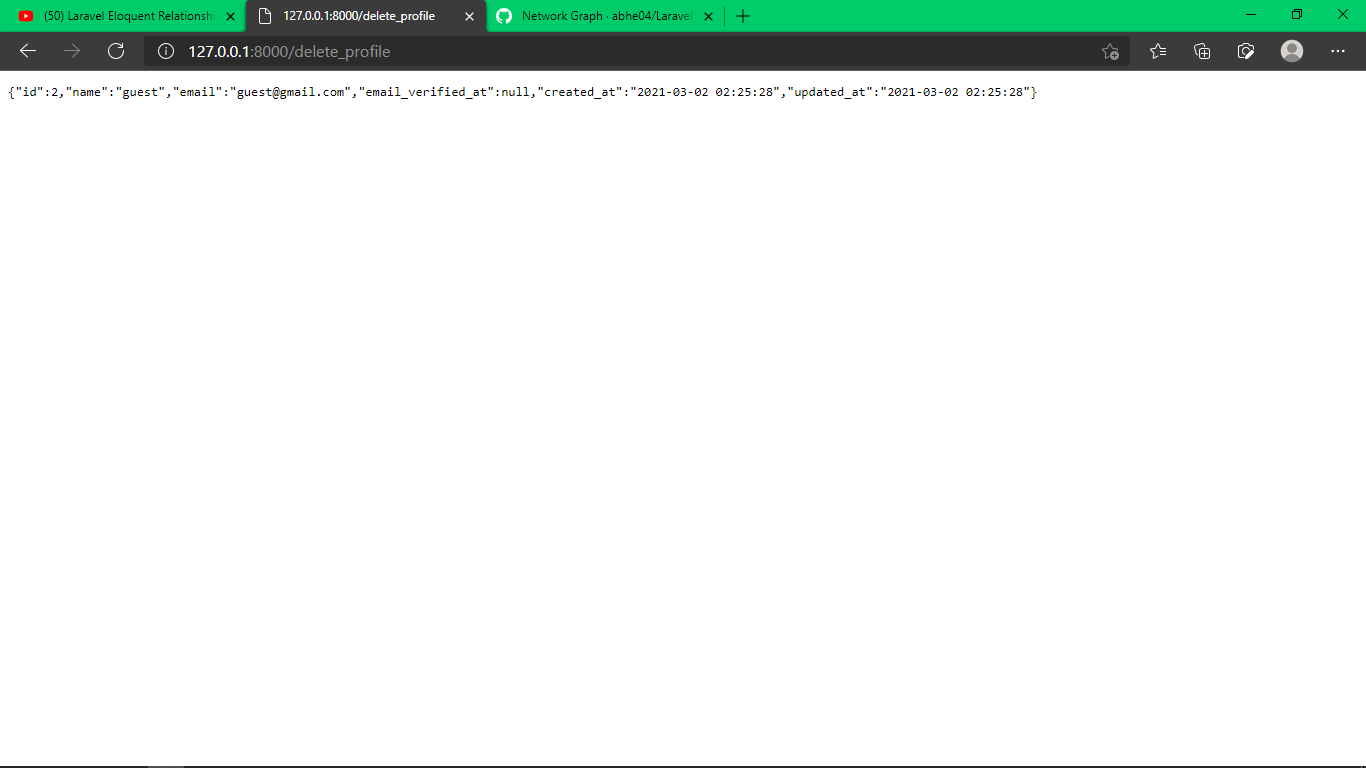
    $user = User::find(2);

    $user->profile()->delete();

    return $user;

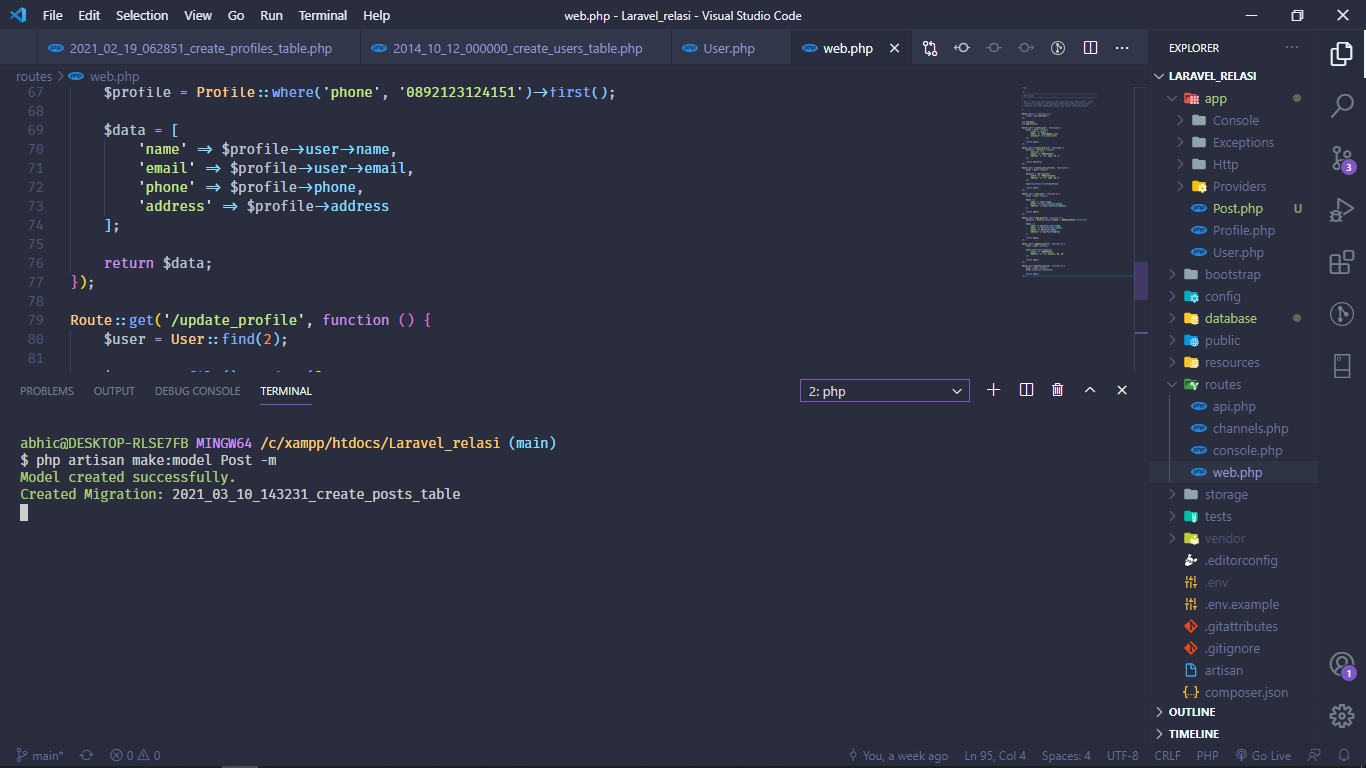
});

Yang setelah dieksekusi di website local Laravel, tampilannya seperti gambar berikut ini :

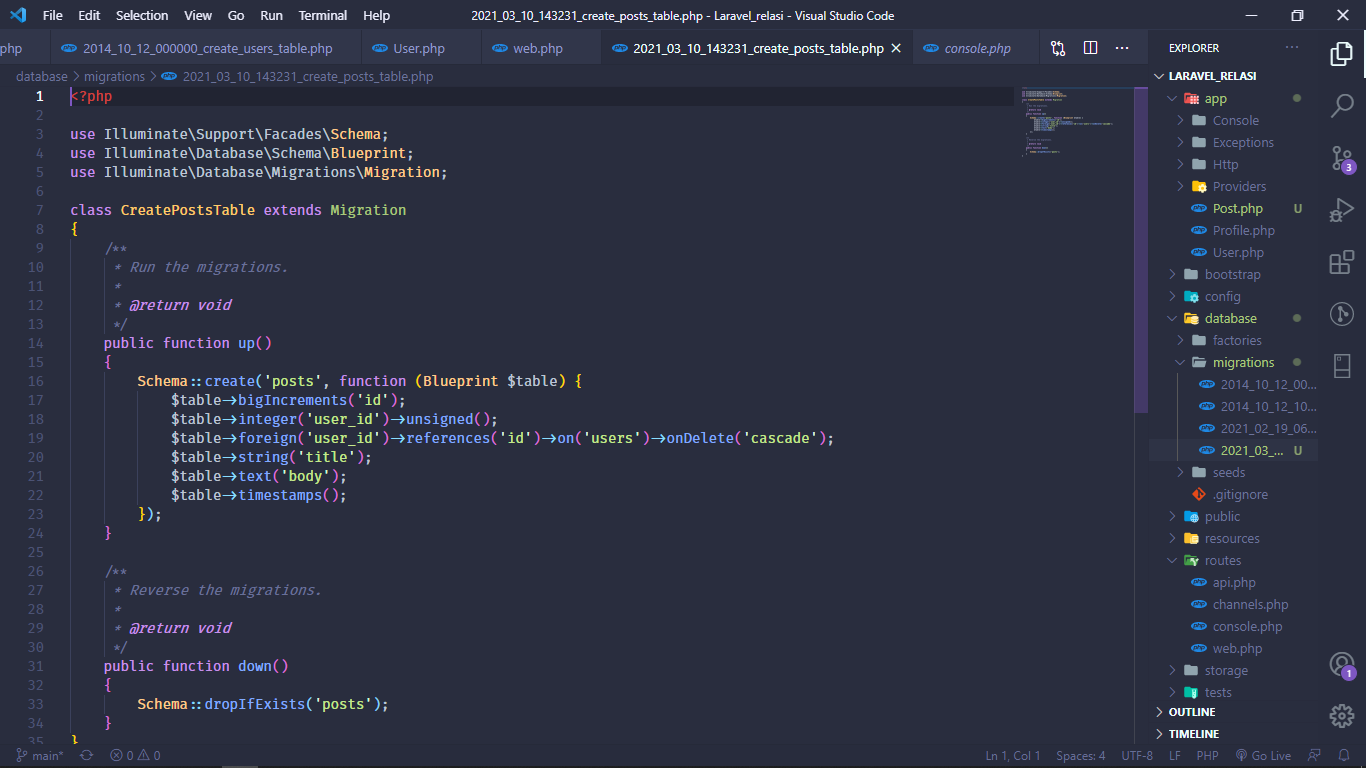


**VI. Bagian 6 – Setup Model Relasi One To Many**

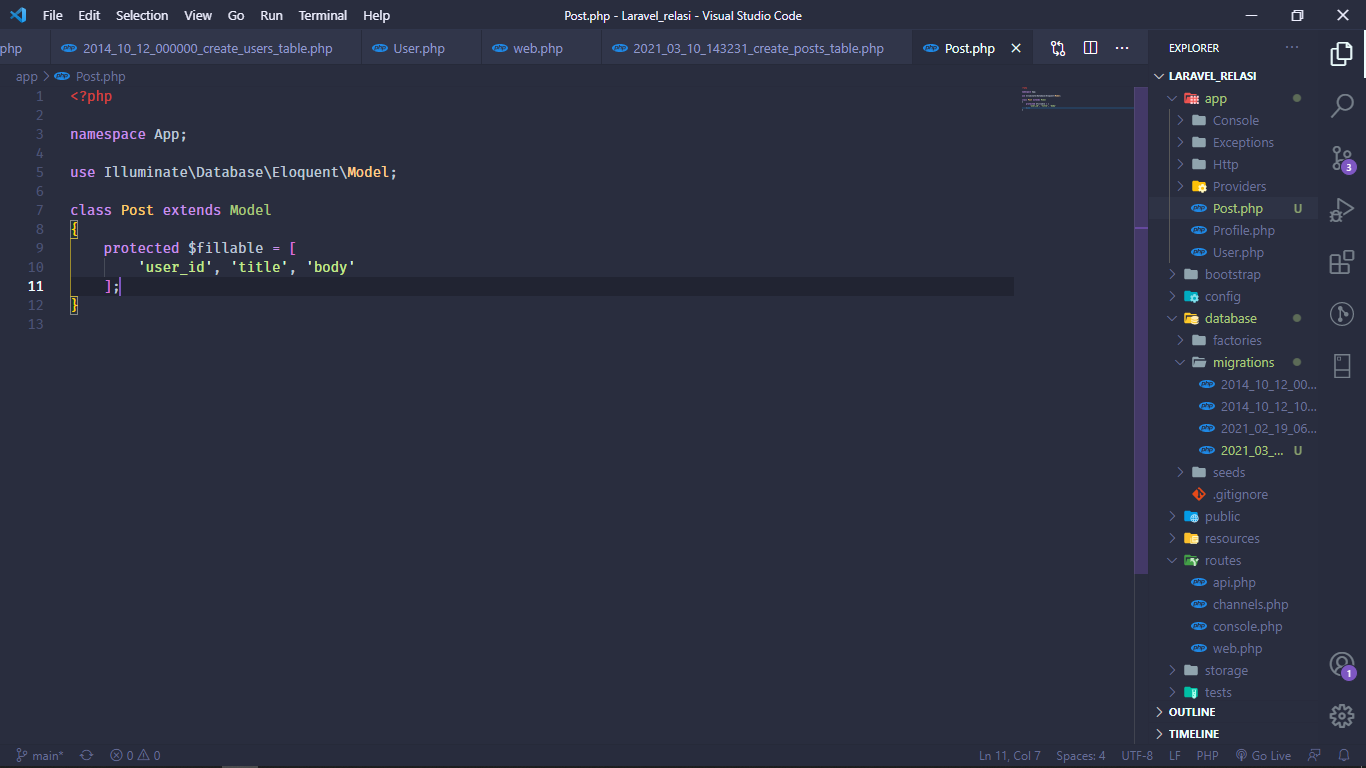
Bukalah terminal yang sudah disediakan Visual Studio Code atau kita juga bisa membuka command prompt dengan mendirect folder Laravel kita terlebih dahulu, lalu ketik command di bawah ini :



Kemudian, tambahkan code berikut ini pada function up() pada file schema yang telah dibuat sebelumnya (model berikut schema) dalam folder database > migrations.



Pada file model yang bernama Post.php, kita bisa menambahkan beberapa atribut dibawah ini :

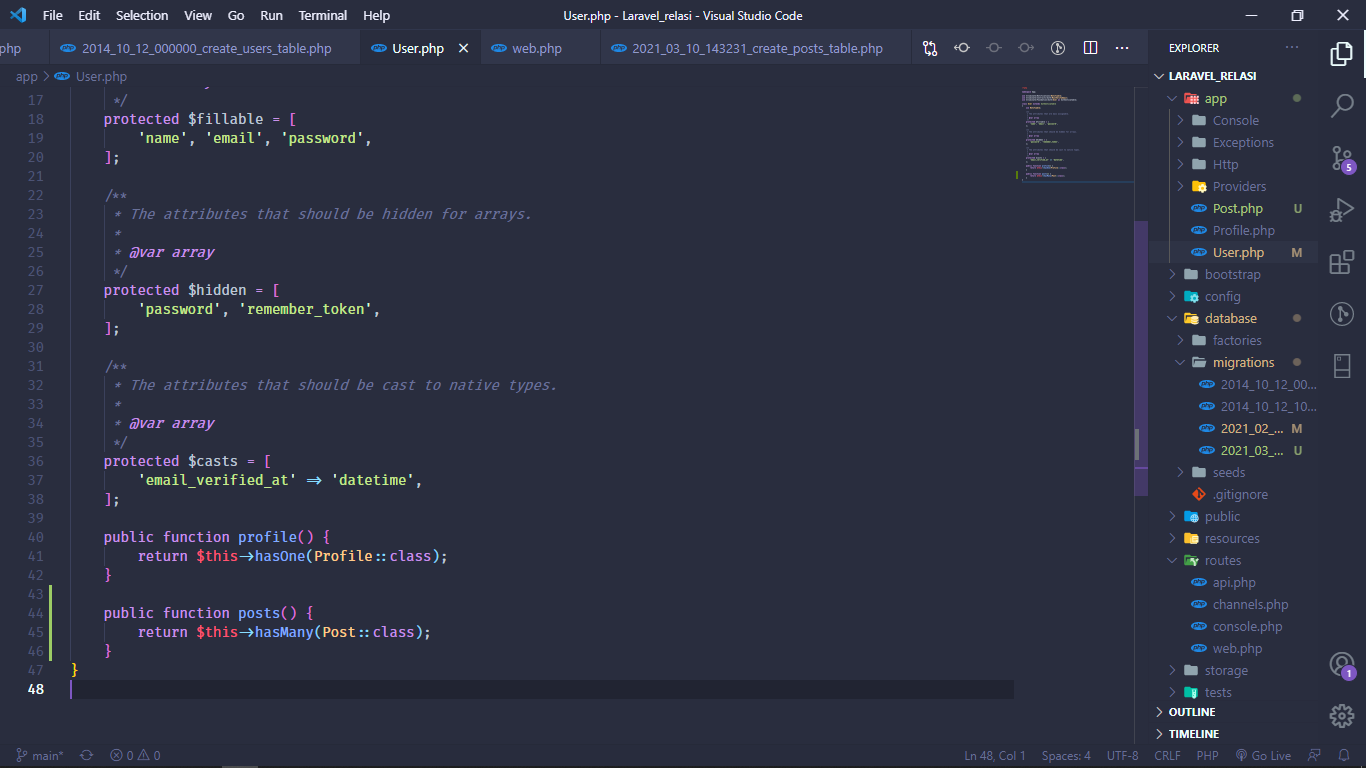


protected $fillable = [

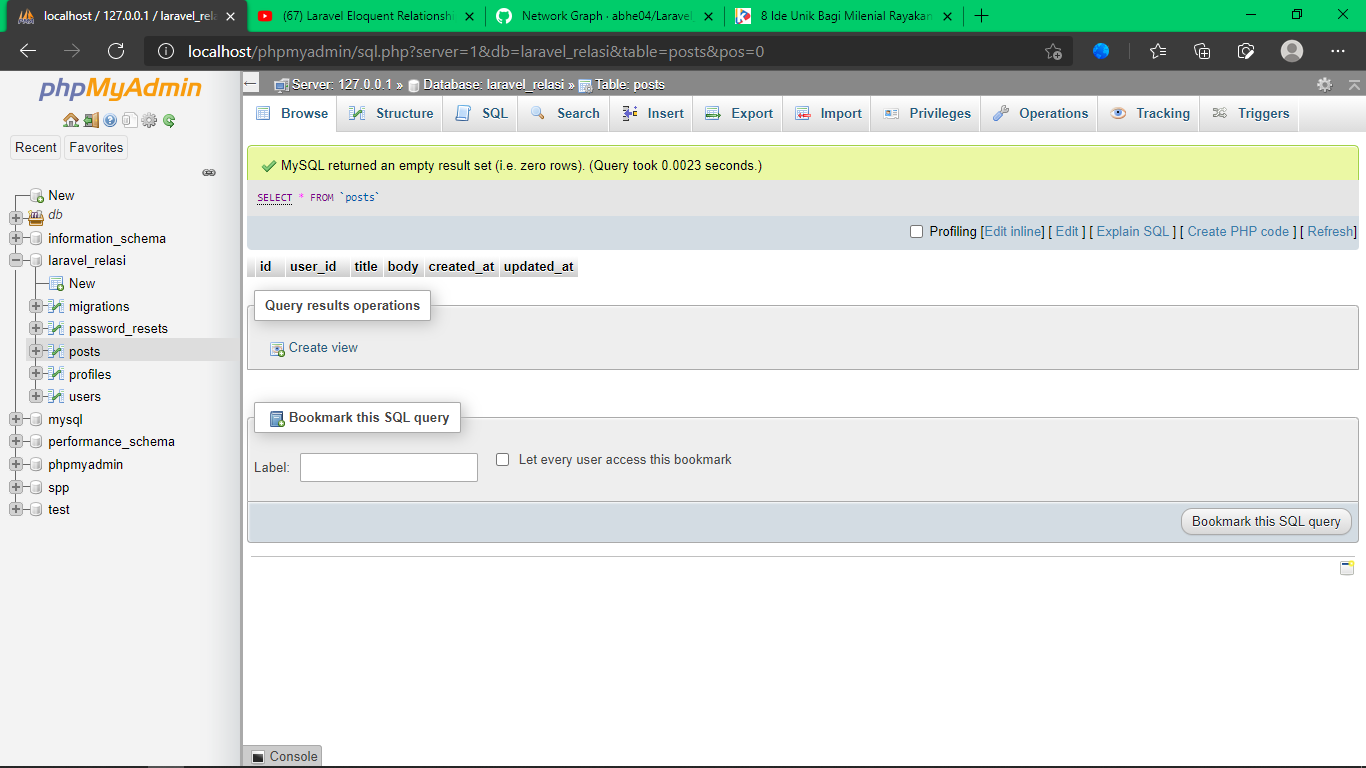
        'user\_id', 'title', 'body'

    ];

Setelah itu, kita bisa migrate dengan menggunakan command “php artisan migrate” pada terminal yang telah di sediakan untuk menambahkan tabel Post pada Database. Lalu, masuk ke file User.php dan tambahkan function posts() seperti dibawah ini :



Setelah di migrate, maka tampilan pada Server Database seperti gambar berikut :



**VII. Bagian 7 - Membuat Read Data Posts**

Pertama, kita masuk ke file web.php untuk menambahkan Route dibawah ini agar kita bisa mengakses pembacaan data posts dengan url /read\_posts.

Route::get('/read\_posts', function () {

    $user = User::find(1);

    $posts = $user->posts()->get();

    foreach ($posts as $post) {

        $data[] = [

            'name' => $post->user->name,

            'post\_id' => $post->id,

            'title' => $post->title,

            'body' => $post->body

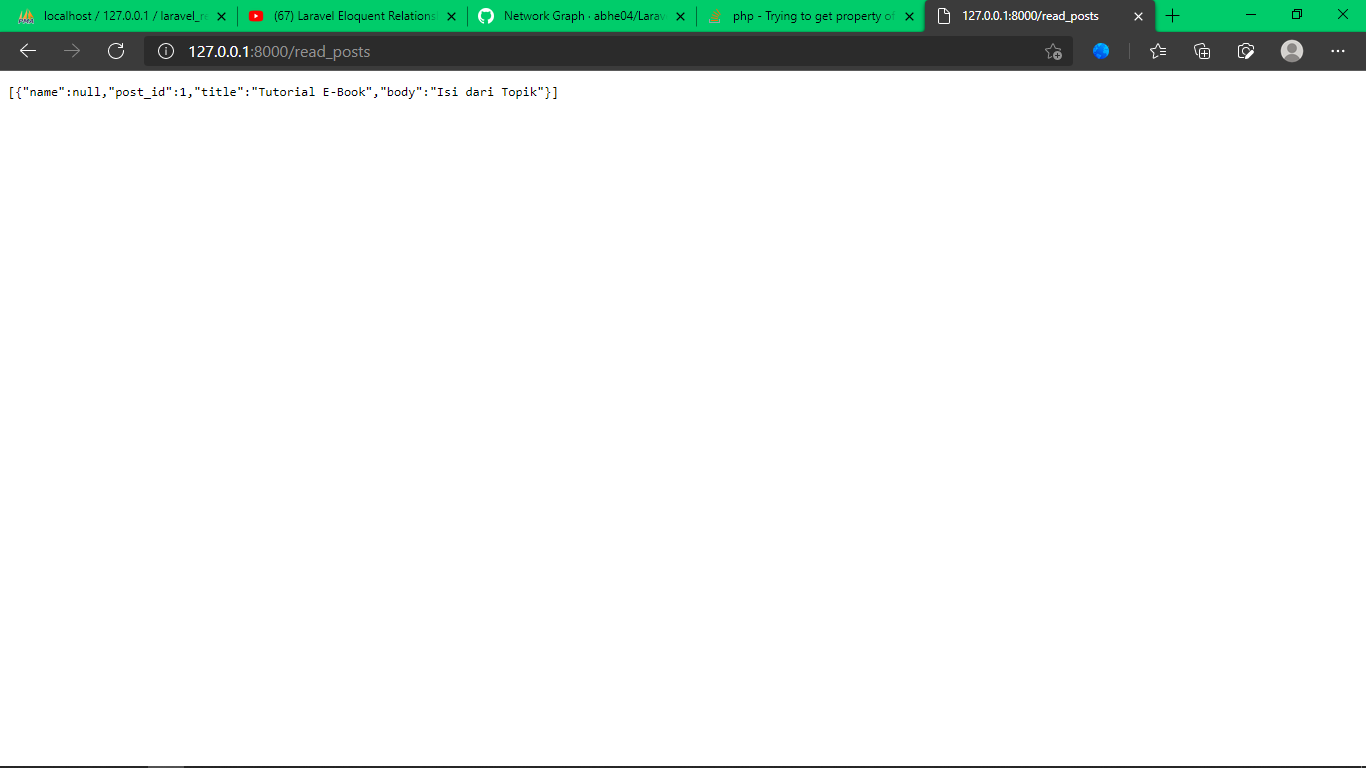
        ];

    }

    return $data;

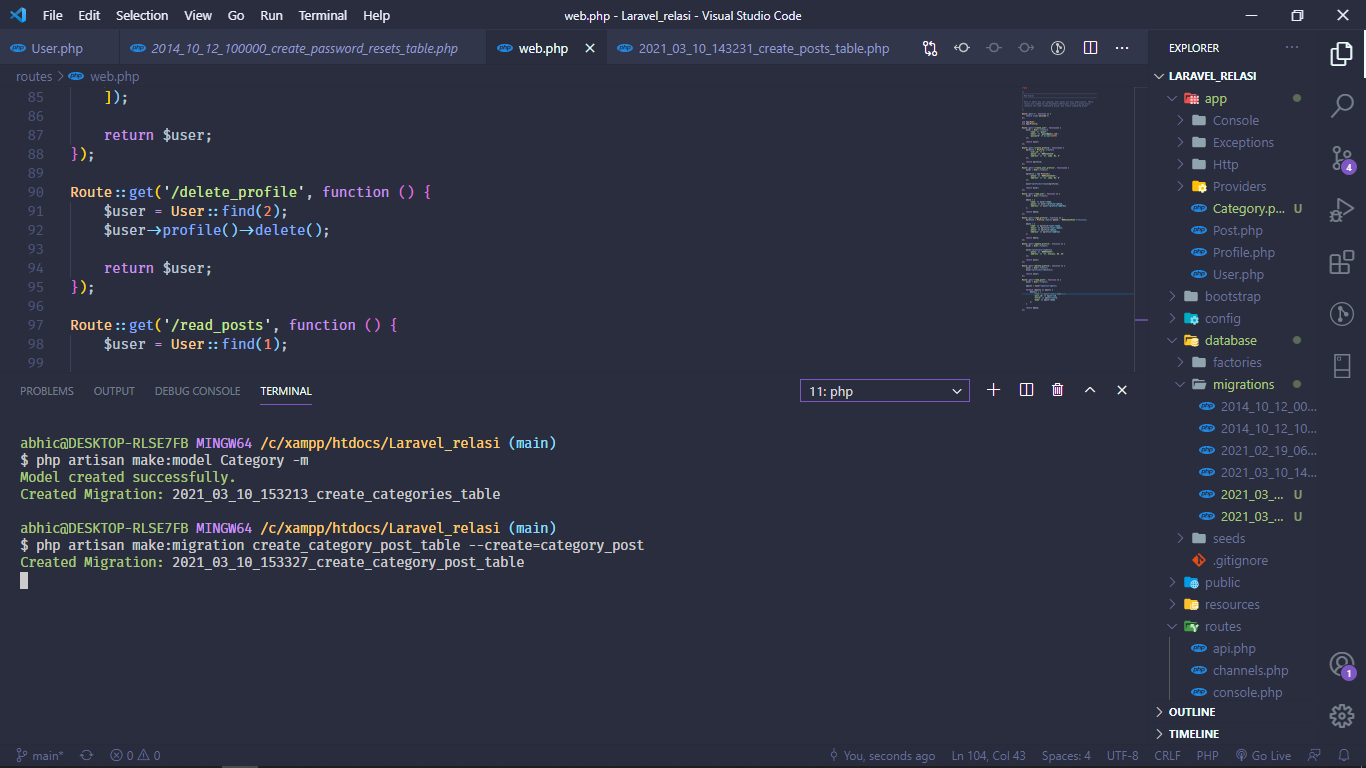
});

Setelah itu, kita bisa membuka di browser mana saja (pastikan xampp control panel telah dinyalakan) sehingga tampilannya seperti dibawah ini :

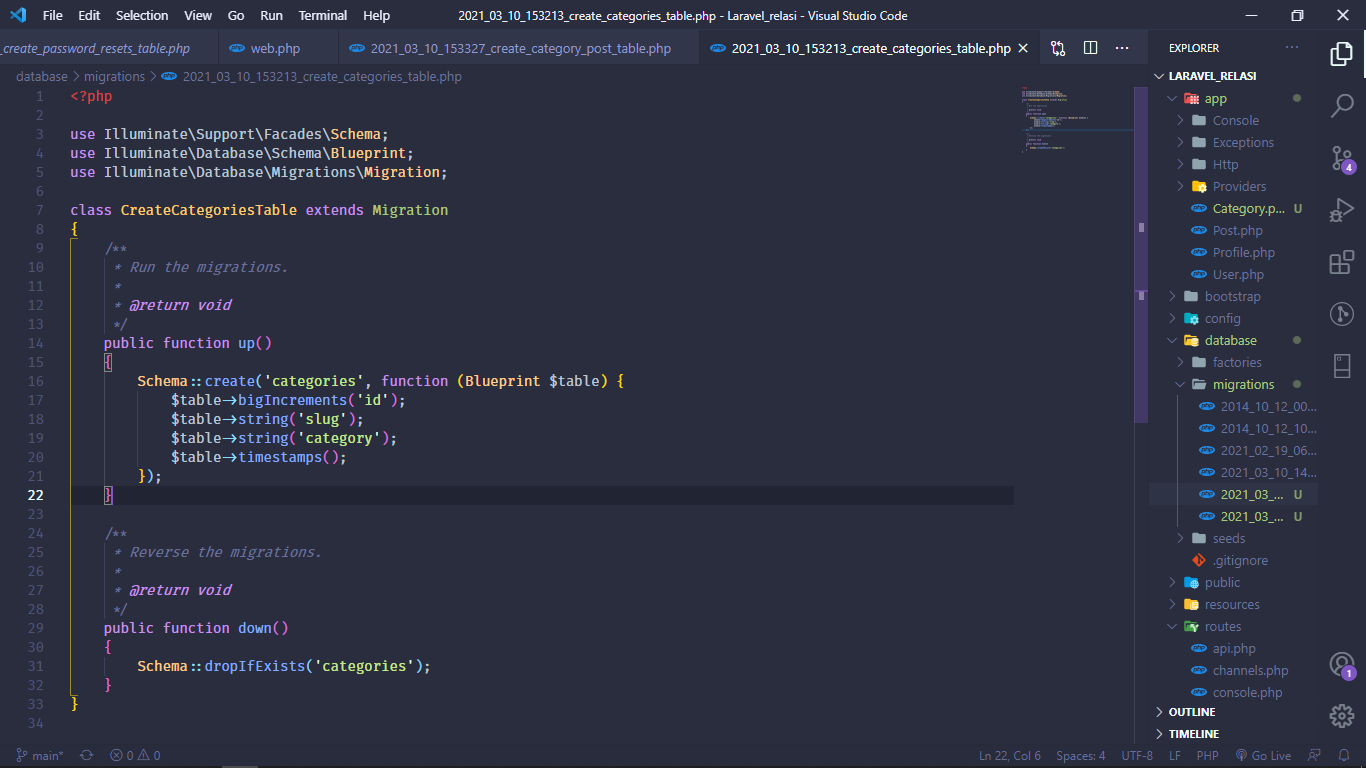


**VIII. Bagian 8 - Setup Model Relasi Many To Many**

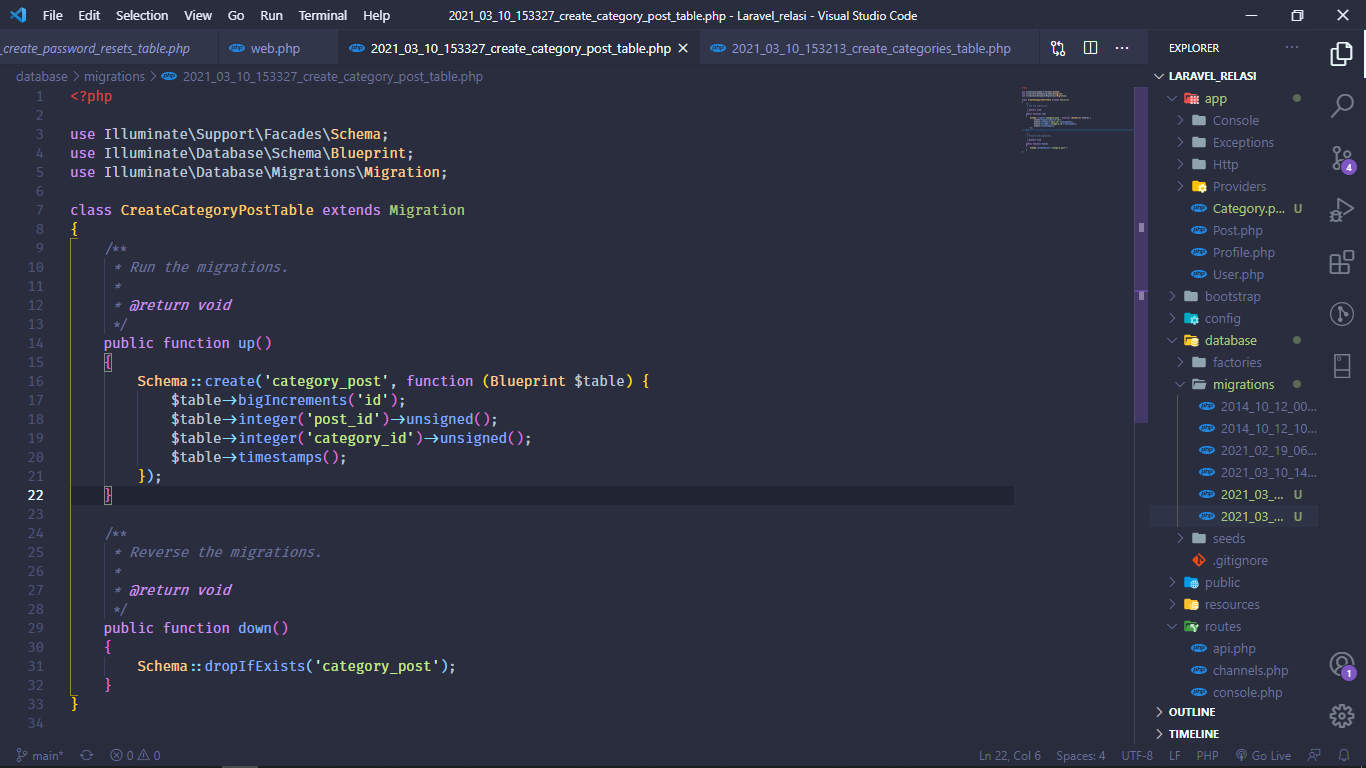
Pertama, bukalah terminal yang sudah disediakan Visual Studio Code atau kita juga bisa membuka command prompt dengan mendirect folder Laravel kita terlebih dahulu, lalu ketik command di bawah ini :



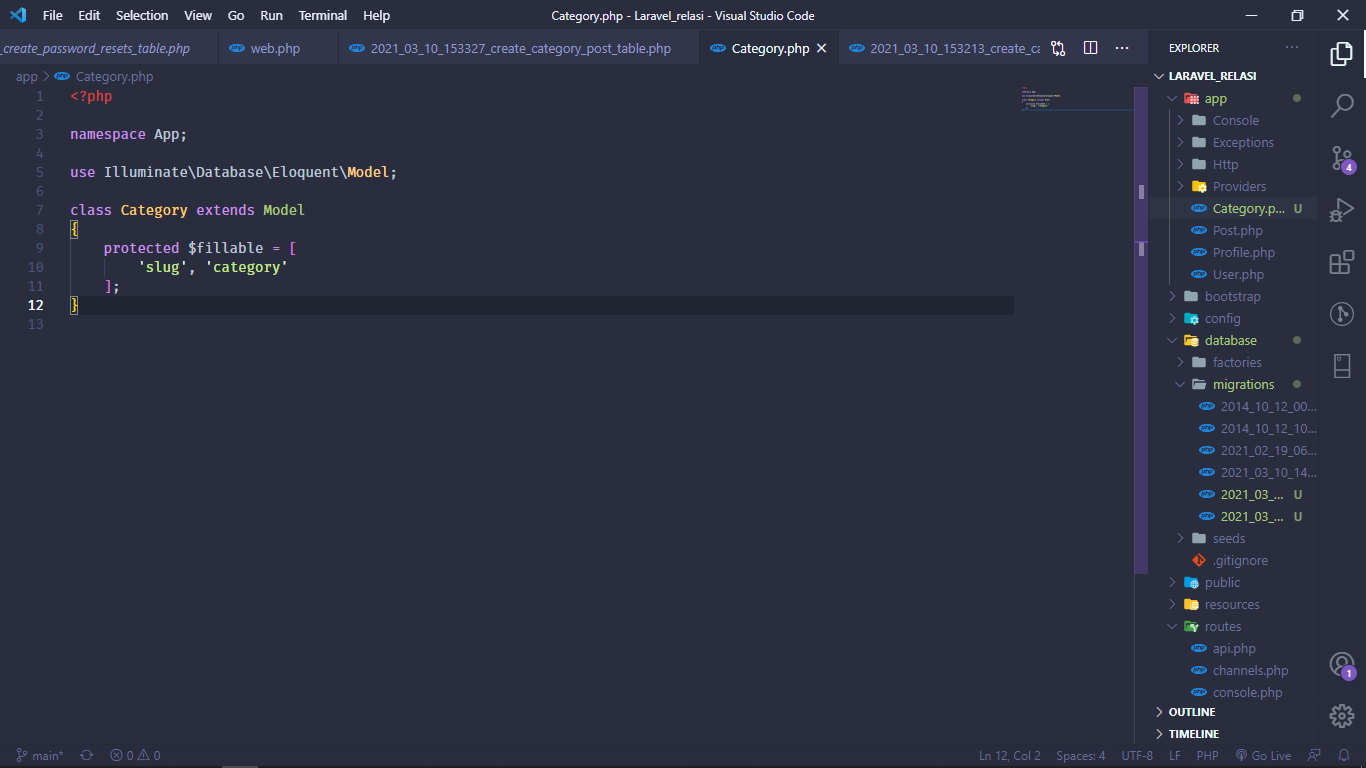
Kemudian, masuk ke folder database > migrations > dan cari file yang bernama xxxx\_xx\_xx\_xxxxxx\_create\_categories\_table.php dan tambahkan beberapa struktur (slug dan category) pada function seperti gambar berikut ini :



Setelah itu, cari file yang bernama xxxx\_xx\_xx\_xxxxxx\_create\_category\_post\_table.php dan tambahkan beberapa struktur pada function seperti gambar berikut ini :



Seperti pembuatan setup One To Many sebelumnya, kita bisa masuk ke file model yang sudah dibuat yaitu Category.php lalu menambahkan beberapa atribut seperti gambar dibawah ini :

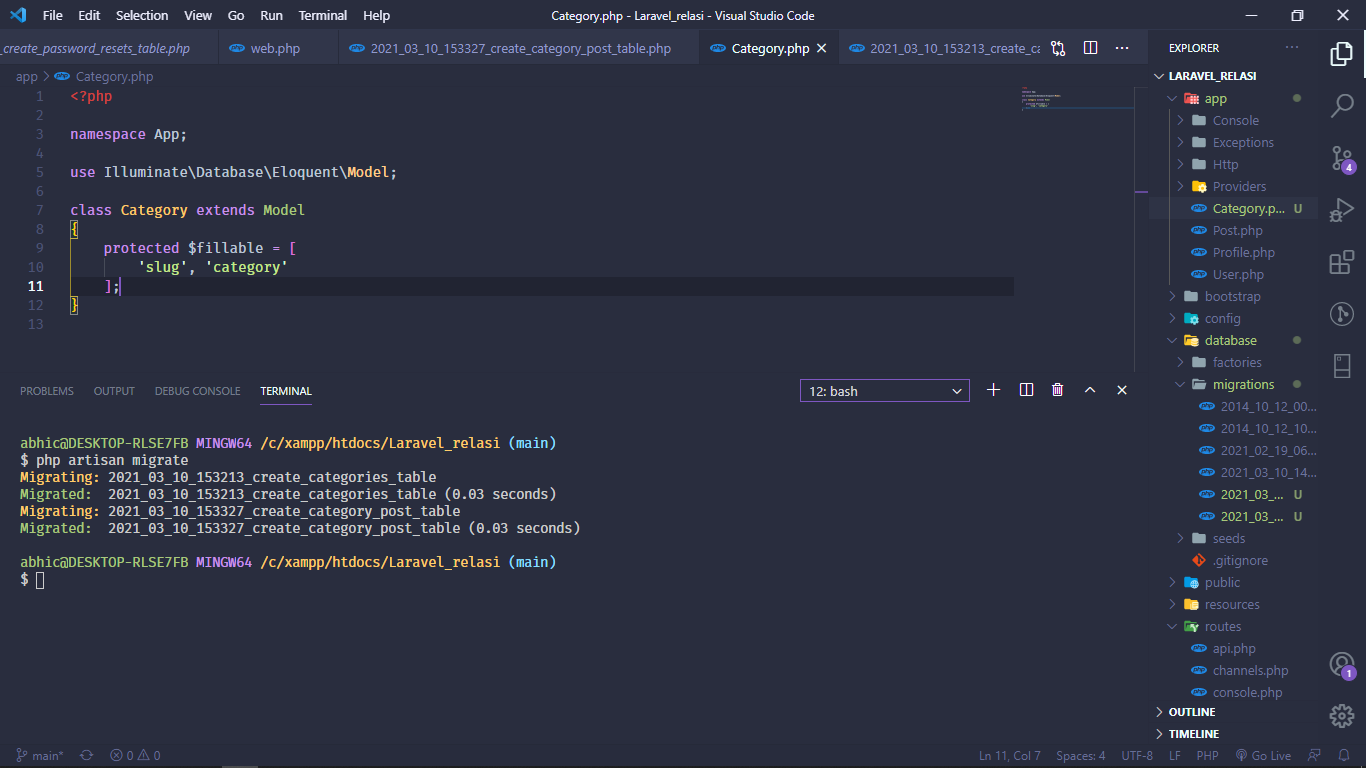


protected $fillable = [

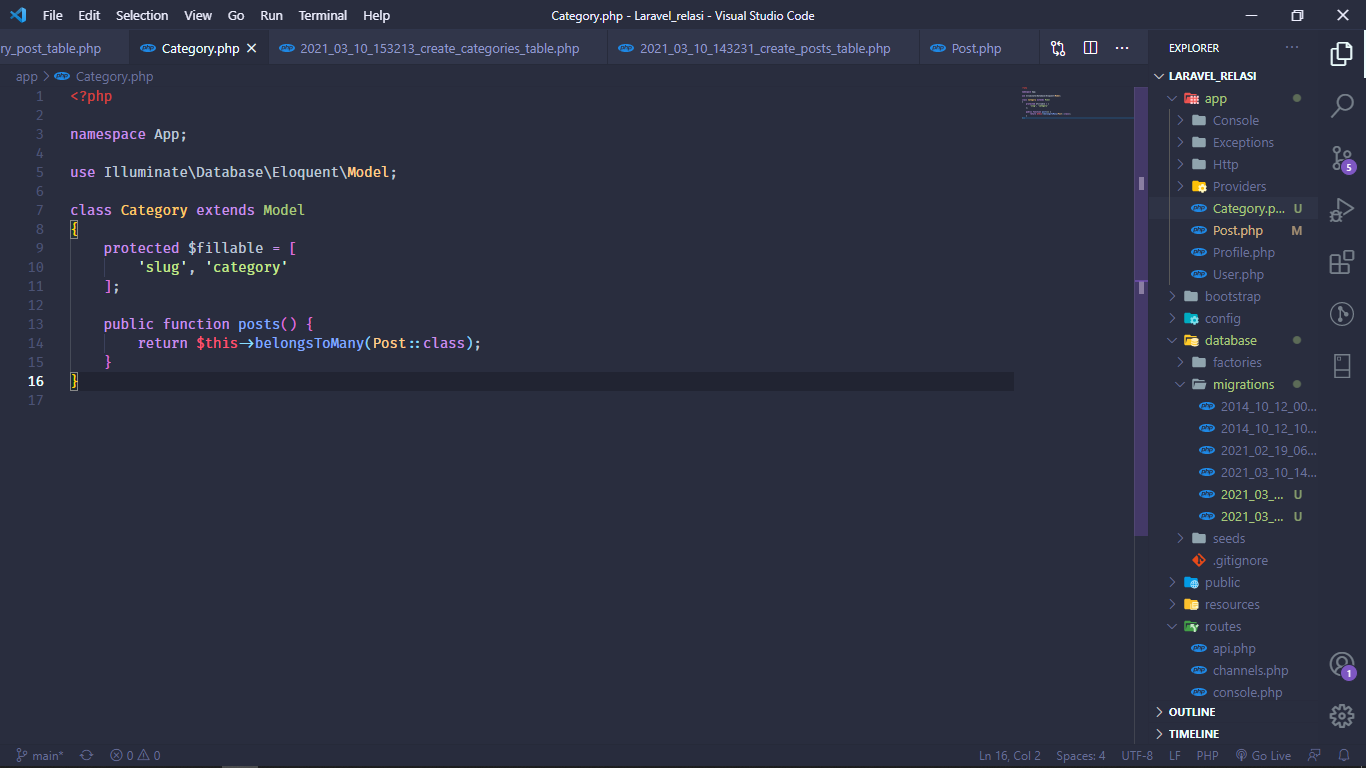
        'slug', 'category'

    ];

Selanjutnya, kita bisa migrate kedua blueprint yang sudah kita konfigurasi dengan command “php artisan migrate”.



Lalu, kita bisa kembali ke file category.php dengan menambahkan sebuah function baru untuk menyatakan adanya relasi Many To Many dengan “belongsToMany”.

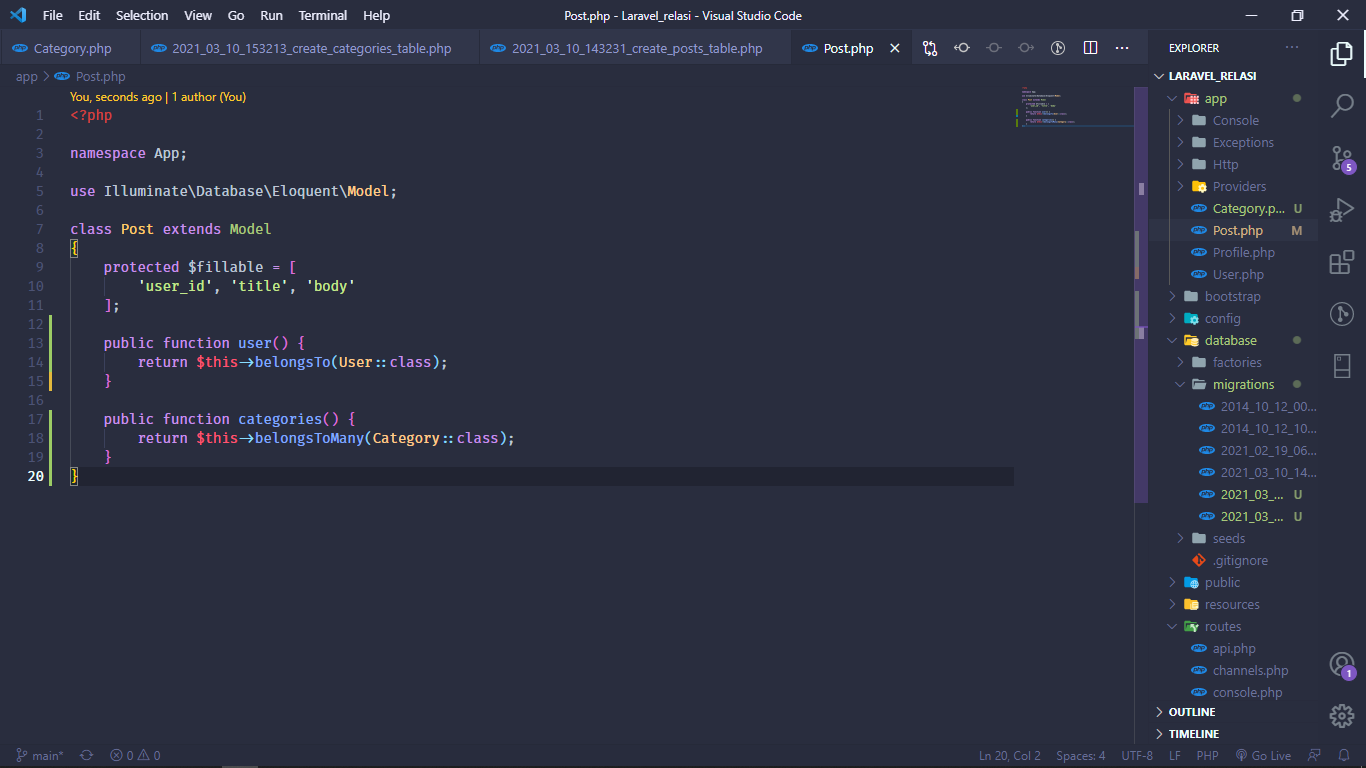


public function posts() {

        return $this->belongsToMany(Post::class);

    }

Setelah itu, kita bisa masuk ke file Post.php dengan memasukkan function baru dibawah ini :



public function categories() {

        return $this->belongsToMany(Category::class);

    }

**IX. Bagian 9 - Membuat Read Data Category**

Kita bisa masuk ke file web.php untuk menambahkan Route dalam mengakses pembacaan data Category dengan url “/read\_category”.

Route::get('/read\_category', function() {

    $post = Post::find(1);

    $categories = $post->categories;

    foreach ($categories as $category) {

        echo $category->slug . '<br/>';

        echo $category->category . '<br/>';

    }

});

Sebelum itu, kita juga harus mendeklarasikan Class Post terlebih dahulu dengan mengetik pada bagian atas ;

use App\Post;

Kemudian, kita bisa mengakses laman tersebut dengan menjalankan Laravel serta url yang telah didaftarkan, maka hasilnya akan seperti gambar berikut ini :

