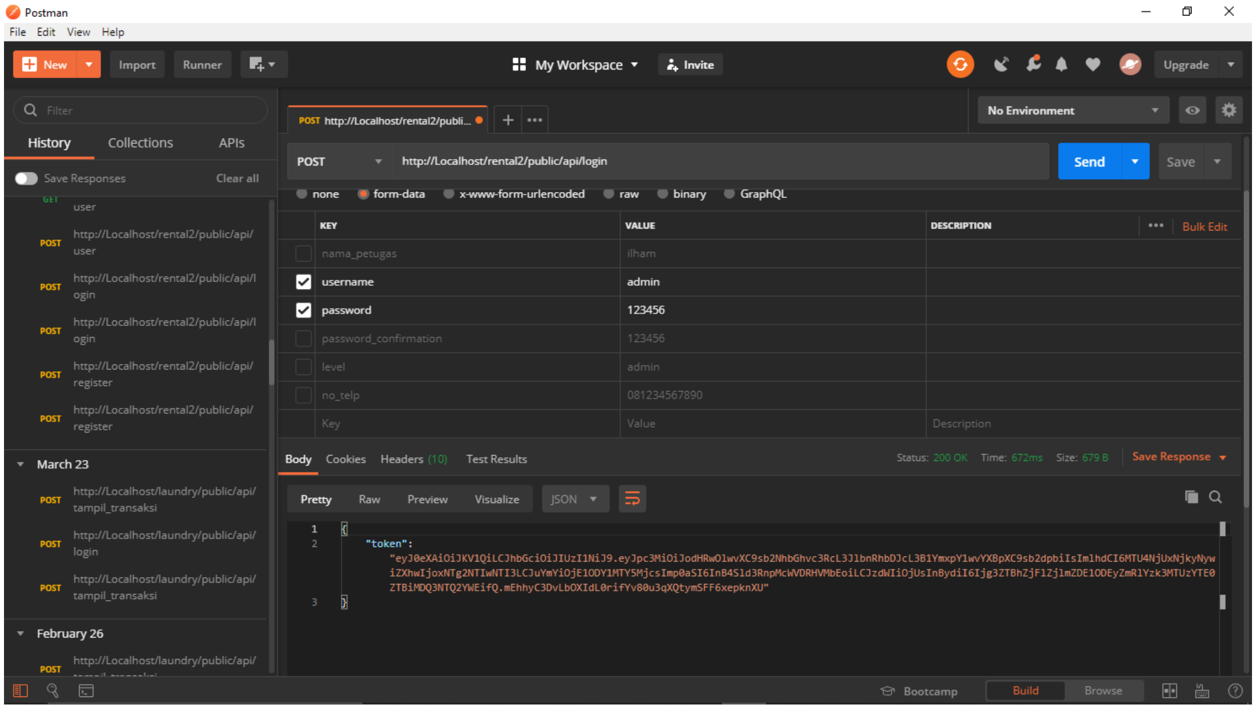
Login



POST http://Localhost/rental2/public/api/login

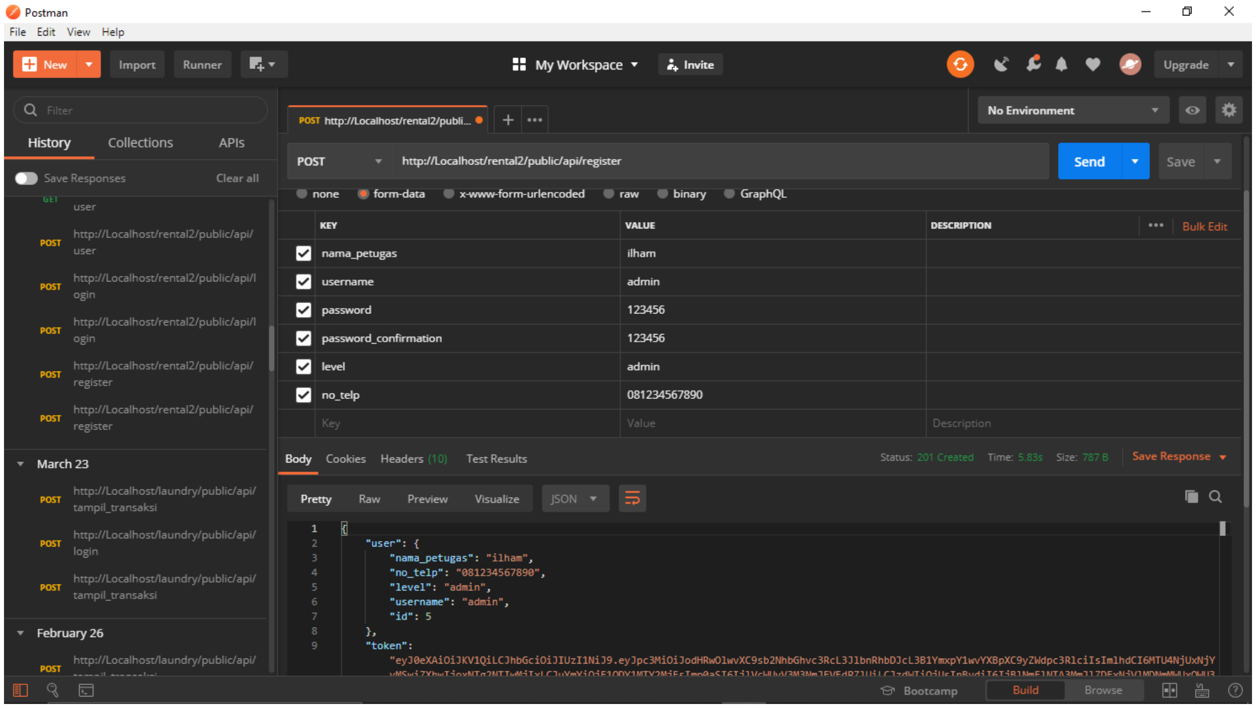
none [✓ form-data] x-www-form-urlencoded raw binary GraphQL

	KEY	VALUE	DESCRIPTION
	nama_petugas	ilham	
✓	username	admin	
✓	password	123456	
	password_confirmation	123456	
	level	admin	
	no_telp	081234567890	

Body Cookies Headers (10) Test Results — Status: 200 OK Time: 672ms Size: 679 B

```
{
    "token":
        "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3RcL3JlbnRhbDJcL3B1YmxpY1wvYXBpXC9sb2dpbiIsImlhdCI6MTU4NjUxNjkyNywiZXhwIjoxNTg2NTIwNTI3LCJuYmYiOjE1ODY1MTY5MjcsImp0aSI6InB4S1d3RnpMcWVDRHVMbEoiLCJzdWIiOjUsInBydiI6Ijg3ZTBhZjFlZjlmZDE1ODEyZmRlYzk3MTUzYTE0ZTBiMDQ3NTQ2YWEifQ.mEhhyC3DvLbOXIdL0rifYv80u3qXQtymSFF6xepknXU"
}
```

Register



POST http://Localhost/rental2/public/api/register

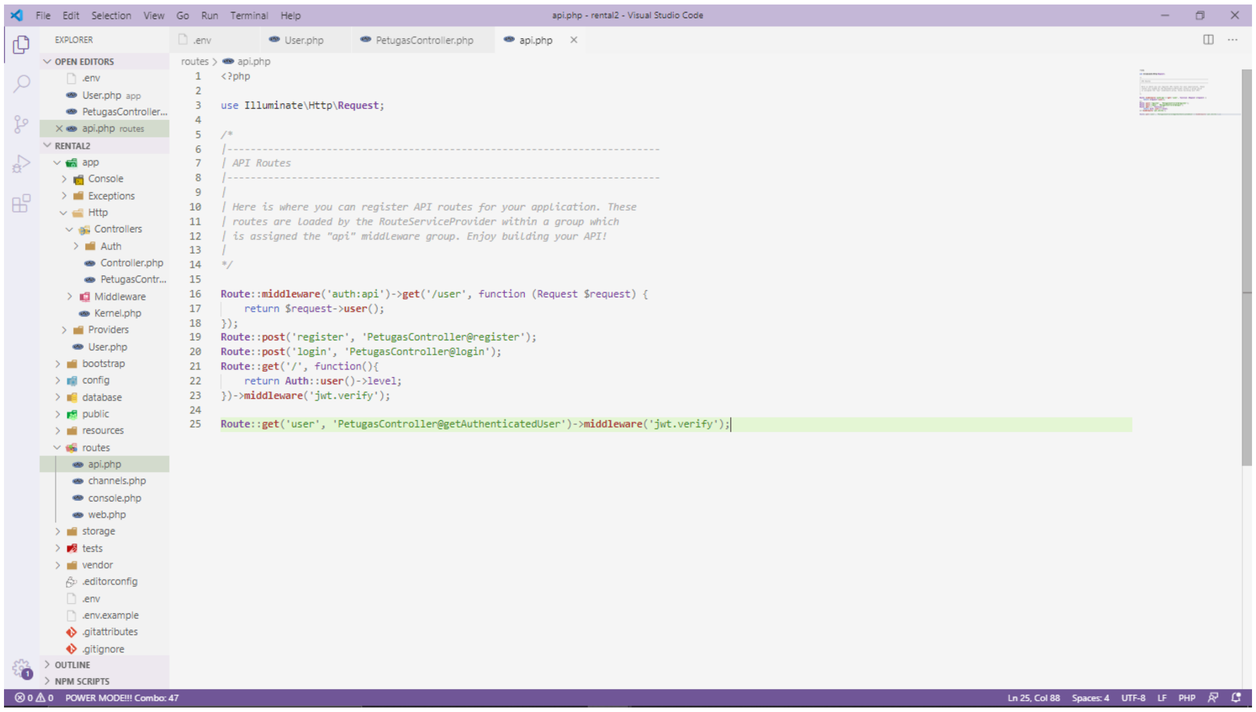
none [✓ form-data] x-www-form-urlencoded raw binary GraphQL

	KEY	VALUE	DESCRIPTION
✓	nama_petugas	ilham	
✓	username	admin	
✓	password	123456	
✓	password_confirmation	123456	
✓	level	admin	
✓	no_telp	081234567890	

Body Cookies Headers (10) Test Results — Status: 201 Created Time: 5.83s Size: 787 B

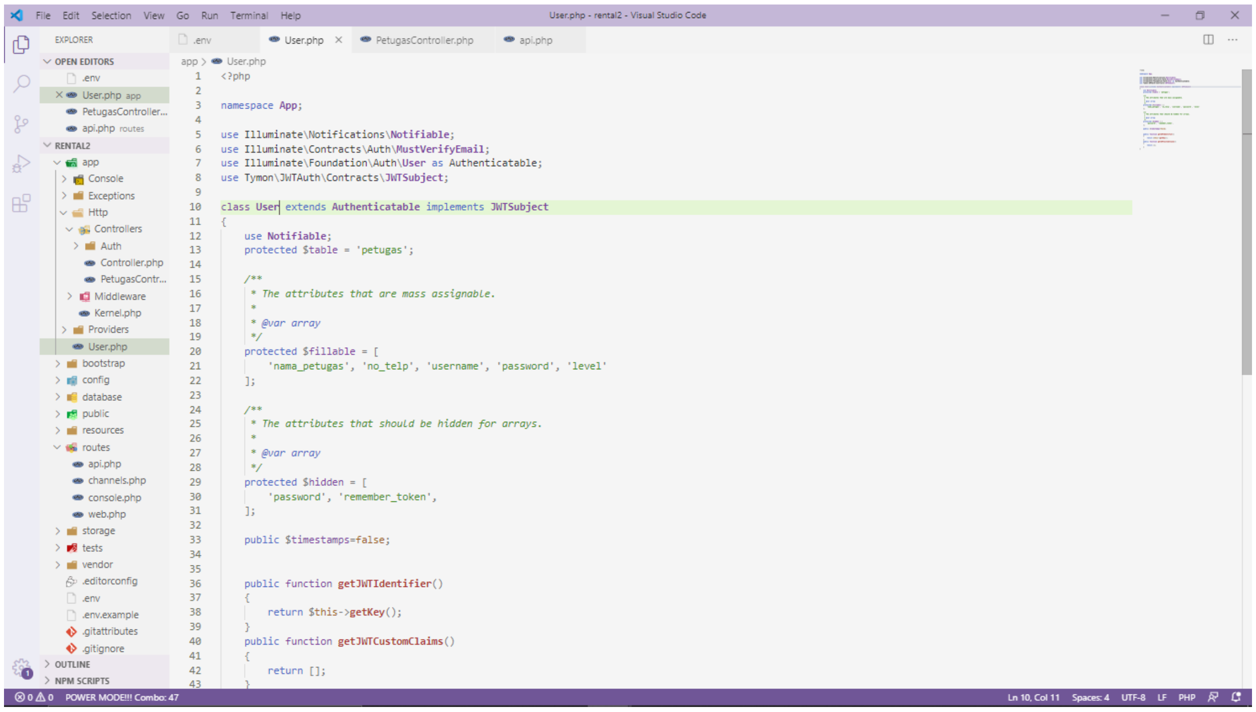
```
{
    "user": {
        "nama_petugas": "ilham",
        "no_telp": "081234567890",
        "level": "admin",
        "username": "admin",
        "id": 5
    },
    "token":
        "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9sb2NhbGhvc3RcL3JlbnRhbDJcL3B1YmxpY1wvYXBpXC9yZWdpc3RlciIsImlhdCI6MTU4NjUxNjY
```

Api.php



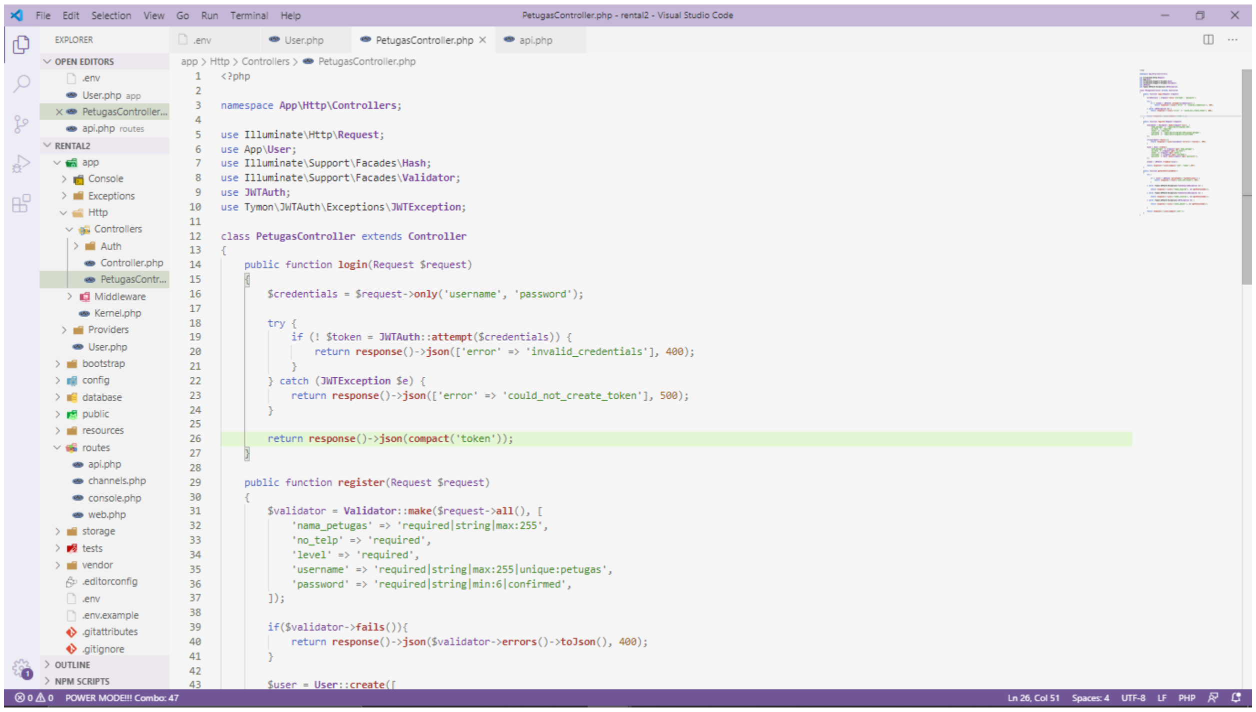
```php
<?php

use Illuminate\Http\Request;

/*
|--------------------------------------------------------------------------
| API Routes
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/

Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});
Route::post('register', 'PetugasController@register');
Route::post('login', 'PetugasController@login');
Route::get('/', function(){
    return Auth::user()->level;
})->middleware('jwt.verify');

Route::get('user', 'PetugasController@getAuthenticatedUser')->middleware('jwt.verify');
```

Model



```php
<?php

namespace App;

use Illuminate\Notifications\Notifiable;
use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Tymon\JWTAuth\Contracts\JWTSubject;

class User extends Authenticatable implements JWTSubject
{
    use Notifiable;
    protected $table = 'petugas';

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'nama_petugas', 'no_telp', 'username', 'password', 'level'
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];

    public $timestamps=false;


    public function getJWTIdentifier()
    {
        return $this->getKey();
    }
    public function getJWTCustomClaims()
    {
        return [];
    }
```

Controller



```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\User;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use JWTAuth;
use Tymon\JWTAuth\Exceptions\JWTException;

class PetugasController extends Controller
{
    public function login(Request $request)
    {

        $credentials = $request->only('username', 'password');

        try {
            if (! $token = JWTAuth::attempt($credentials)) {
                return response()->json(['error' => 'invalid_credentials'], 400);
            }
        } catch (JWTException $e) {
            return response()->json(['error' => 'could_not_create_token'], 500);
        }

        return response()->json(compact('token'));
    }

    public function register(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'nama_petugas' => 'required|string|max:255',
            'no_telp' => 'required',
            'level' => 'required',
            'username' => 'required|string|max:255|unique:petugas',
            'password' => 'required|string|min:6|confirmed',
        ]);

        if($validator->fails()){
            return response()->json($validator->errors()->toJson(), 400);
        }

        $user = User::create([
```