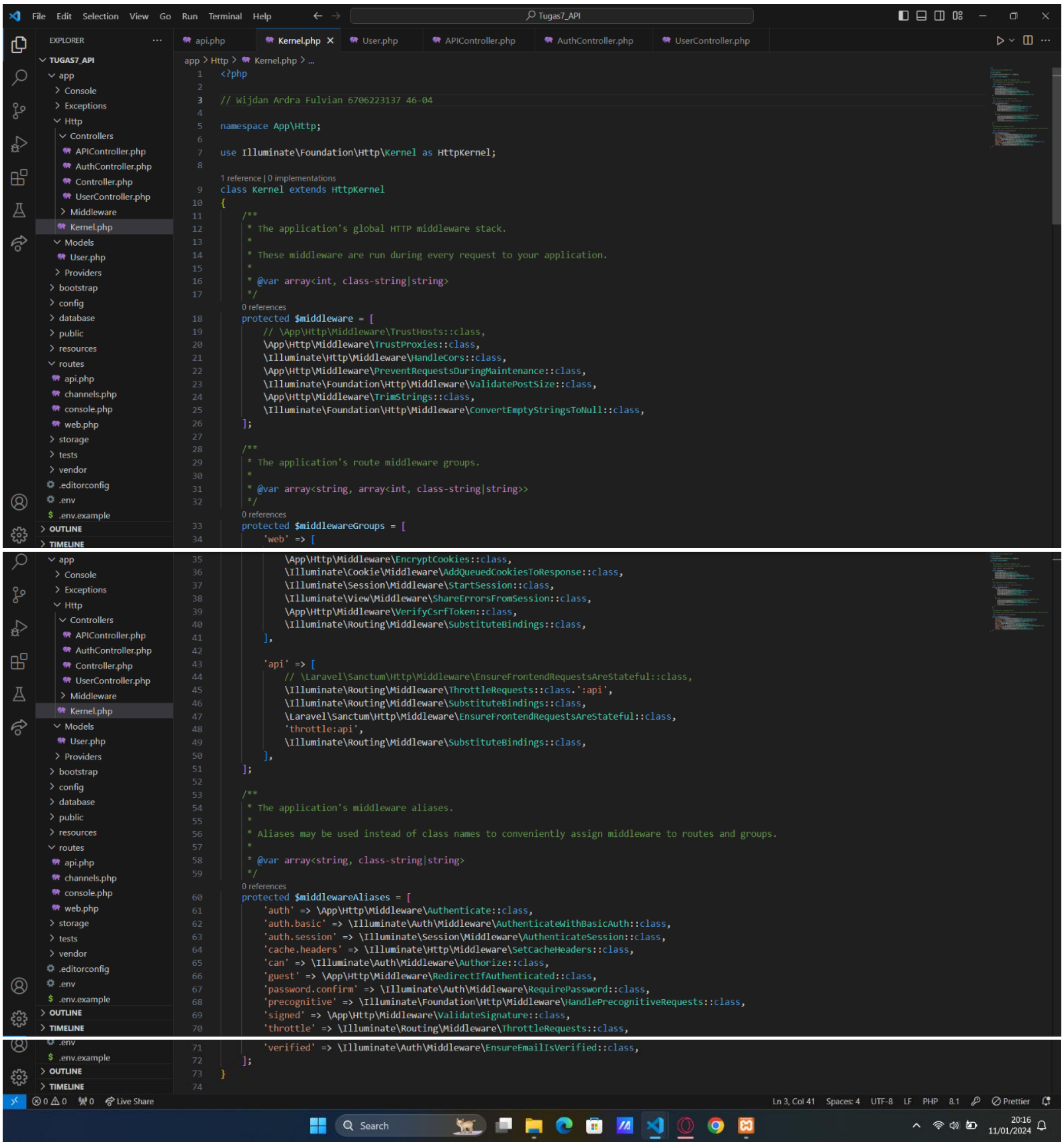
Wijdan Ardra Fulvian

6706223137

D3IF 46-04

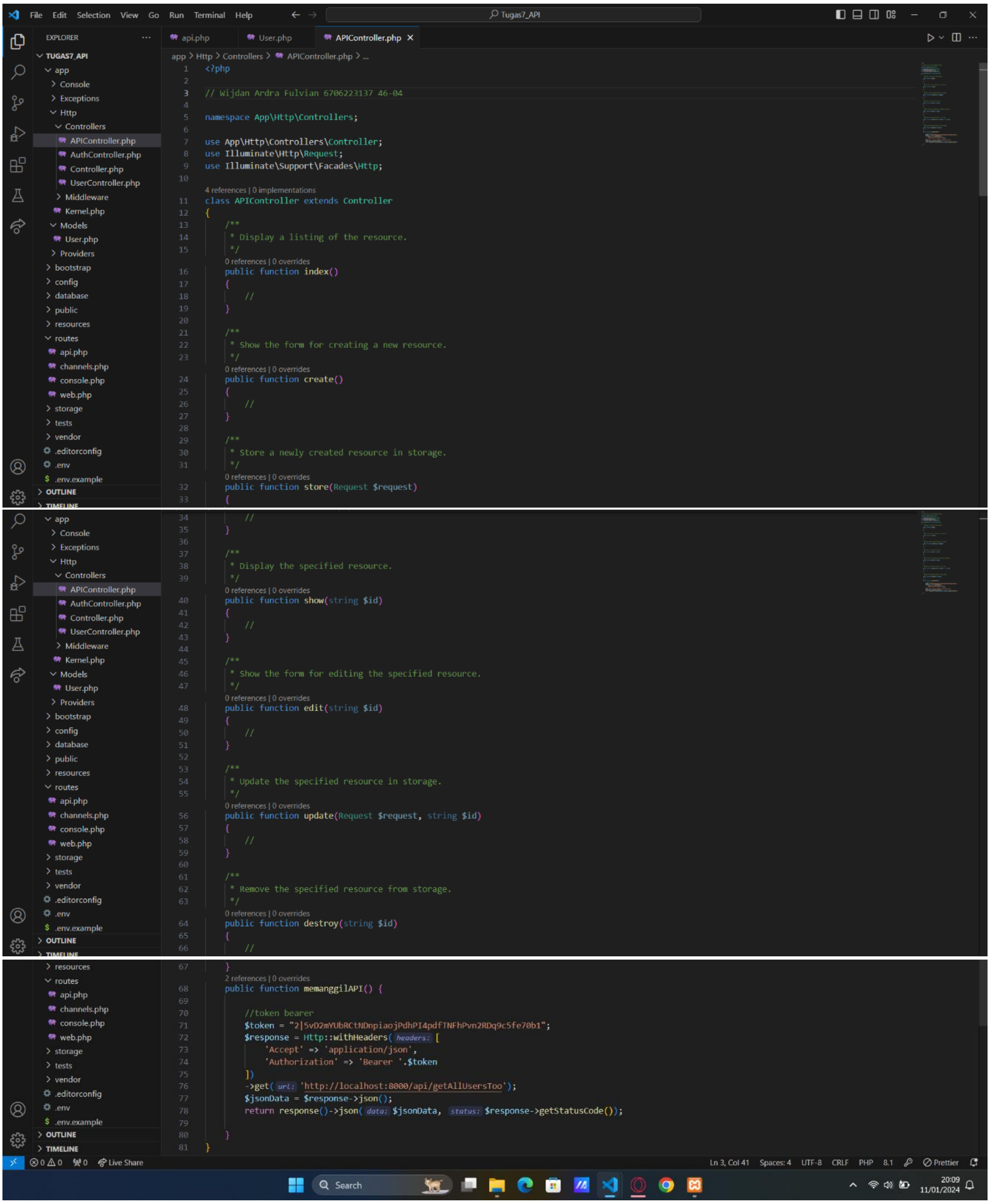
Praktikum 7

- **Http**
    - Kernel.php



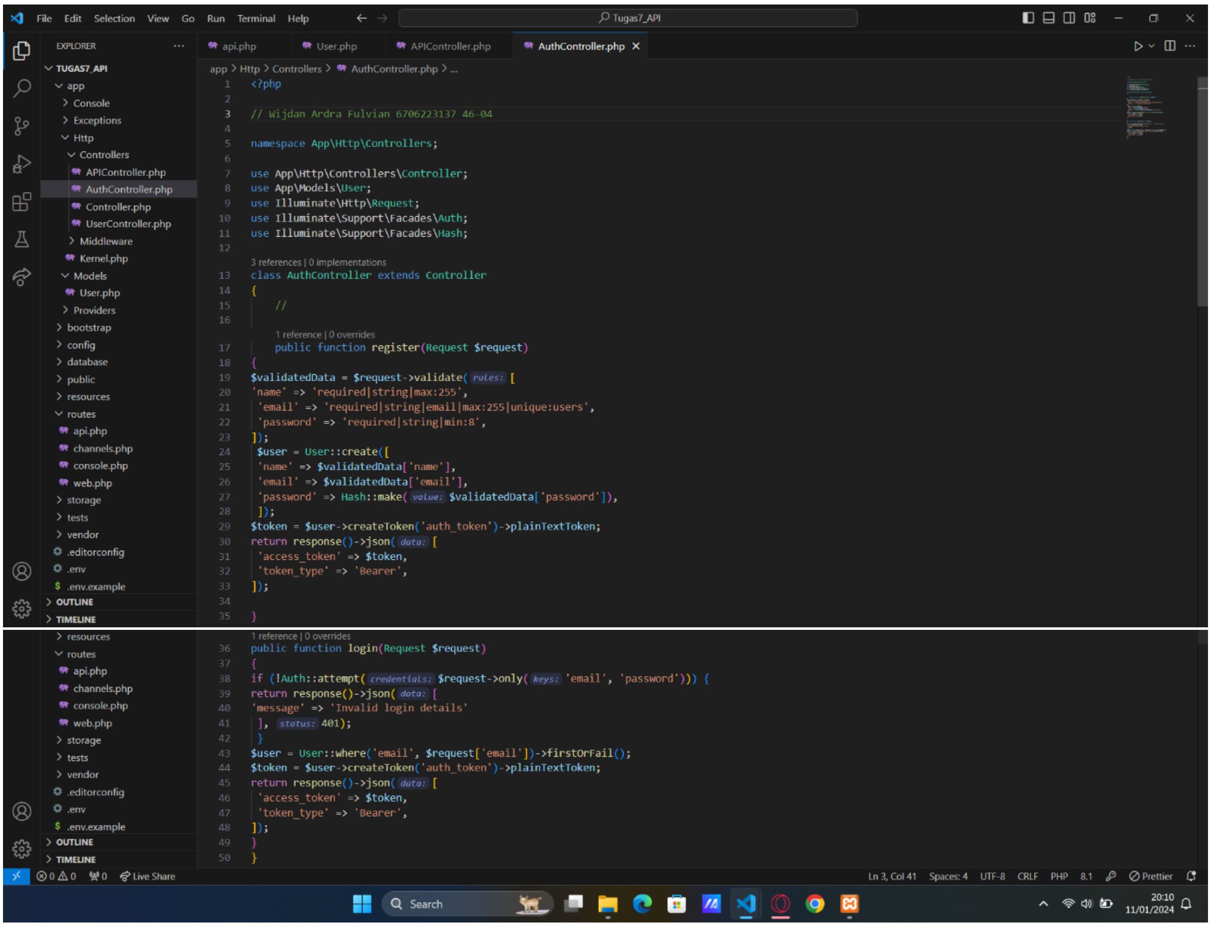
```php
<?php

// Wijdan Ardra Fulvian 6706223137 46-04

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel
{
    /**
     * The application's global HTTP middleware stack.
     *
     * These middleware are run during every request to your application.
     *
     * @var array<int, class-string|string>
     */
    protected $middleware = [
        // \App\Http\Middleware\TrustHosts::class,
        \App\Http\Middleware\TrustProxies::class,
        \Illuminate\Http\Middleware\HandleCors::class,
        \App\Http\Middleware\PreventRequestsDuringMaintenance::class,
        \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
        \App\Http\Middleware\TrimStrings::class,
        \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
    ];

    /**
     * The application's route middleware groups.
     *
     * @var array<string, array<int, class-string|string>>
     */
    protected $middlewareGroups = [
        'web' => [
            \App\Http\Middleware\EncryptCookies::class,
            \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
            \Illuminate\Session\Middleware\StartSession::class,
            \Illuminate\View\Middleware\ShareErrorsFromSession::class,
            \App\Http\Middleware\VerifyCsrfToken::class,
            \Illuminate\Routing\Middleware\SubstituteBindings::class,
        ],

        'api' => [
            // \Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful::class,
            \Illuminate\Routing\Middleware\ThrottleRequests::class.':api',
            \Illuminate\Routing\Middleware\SubstituteBindings::class,
            \Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful::class,
            'throttle:api',
            \Illuminate\Routing\Middleware\SubstituteBindings::class,
        ],
    ];

    /**
     * The application's middleware aliases.
     *
     * Aliases may be used instead of class names to conveniently assign middleware to routes and groups.
     *
     * @var array<string, class-string|string>
     */
    protected $middlewareAliases = [
        'auth' => \App\Http\Middleware\Authenticate::class,
        'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
        'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,
        'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
        'can' => \Illuminate\Auth\Middleware\Authorize::class,
        'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
        'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
        'precognitive' => \Illuminate\Foundation\Http\Middleware\HandlePrecognitiveRequests::class,
        'signed' => \App\Http\Middleware\ValidateSignature::class,
        'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
        'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
    ];
}
```

- **Controllers**
    - APIController.php



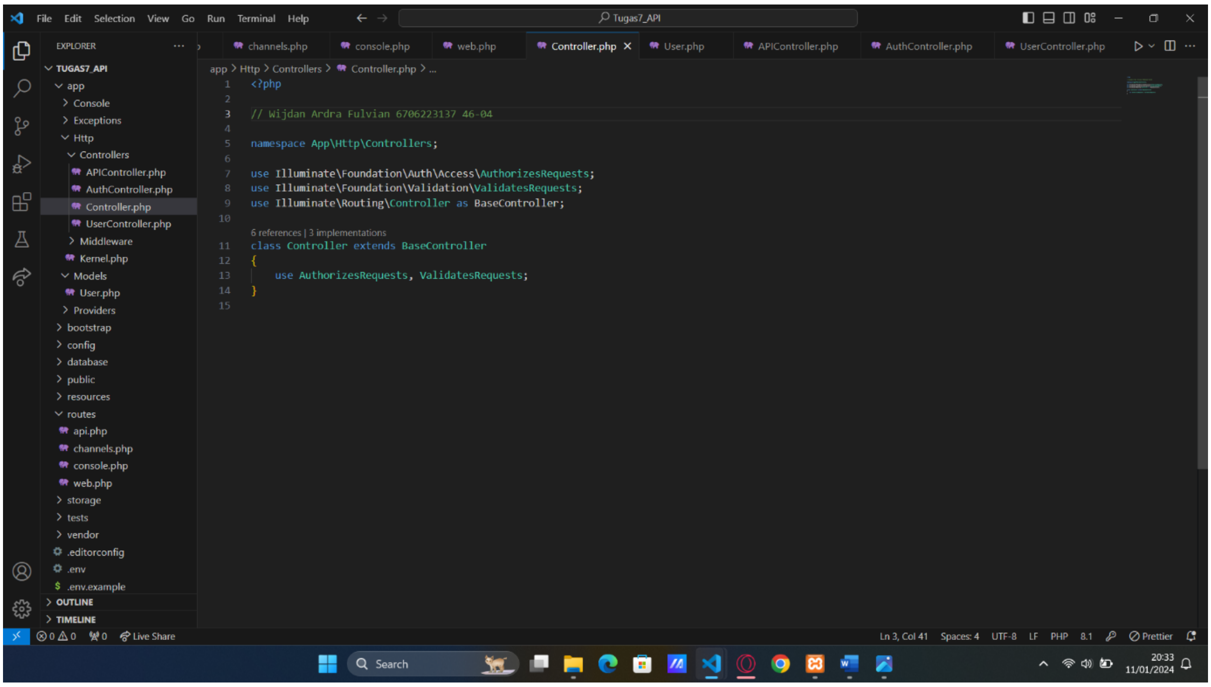
```php
<?php

// Wijdan Ardra Fulvian 6706223137 46-04

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Http;

class APIController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        //
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        //
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        //
    }

    /**
     * Display the specified resource.
     */
    public function show(string $id)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit(string $id)
    {
        //
    }

    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, string $id)
    {
        //
    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy(string $id)
    {
        //
    }
    public function memanggilAPI() {

        //token bearer
        $token = "2|5vD2mYUbRCtNDnpiaojPdhPI4pdfTNFhPvn2RDq9c5fe70b1";
        $response = Http::withHeaders([
            'Accept' => 'application/json',
            'Authorization' => 'Bearer '.$token
        ])
        ->get('http://localhost:8000/api/getAllUsersToo');
        $jsonData = $response->json();
        return response()->json($jsonData, $response->getStatusCode());

    }
}
```

- AuthController.php



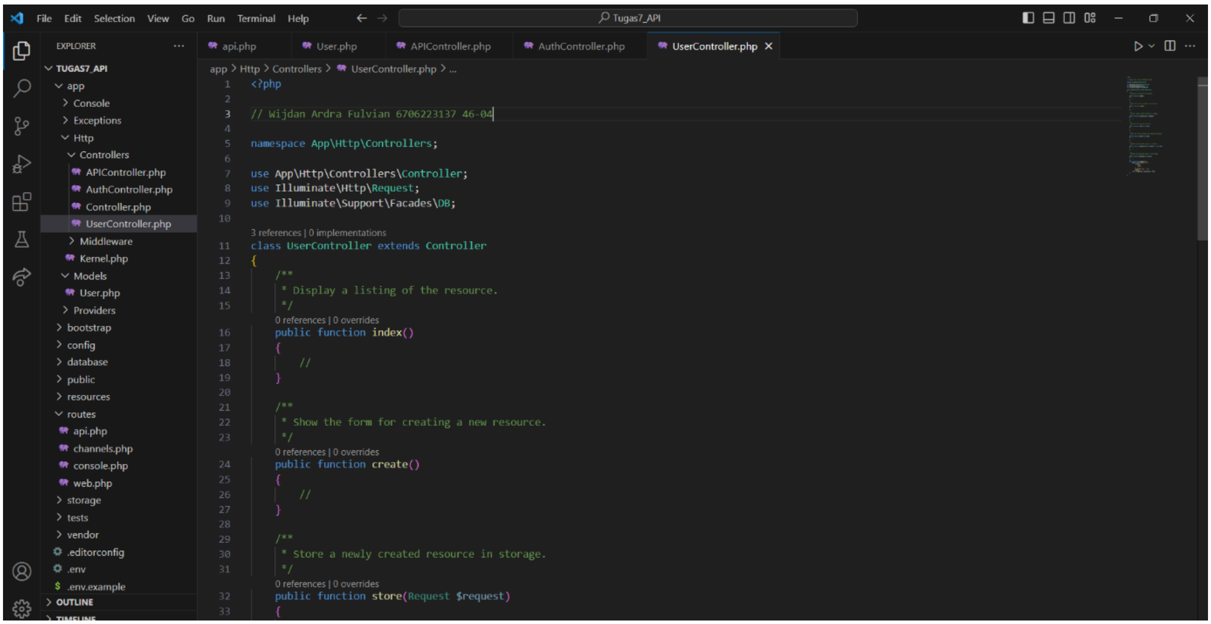
```php
<?php

// Wijdan Ardra Fulvian 6706223137 46-04

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;
use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;

class AuthController extends Controller
{
    //

    public function register(Request $request)
{
$validatedData = $request->validate([
'name' => 'required|string|max:255',
 'email' => 'required|string|email|max:255|unique:users',
 'password' => 'required|string|min:8',
]);
 $user = User::create([
 'name' => $validatedData['name'],
 'email' => $validatedData['email'],
 'password' => Hash::make($validatedData['password']),
 ]);
$token = $user->createToken('auth_token')->plainTextToken;
return response()->json([
 'access_token' => $token,
 'token_type' => 'Bearer',
]);

}
public function login(Request $request)
{
if (!Auth::attempt($request->only('email', 'password'))) {
return response()->json([
'message' => 'Invalid login details'
 ], 401);
 }
$user = User::where('email', $request['email'])->firstOrFail();
$token = $user->createToken('auth_token')->plainTextToken;
return response()->json([
 'access_token' => $token,
 'token_type' => 'Bearer',
]);
}
}
```

- Controller.php

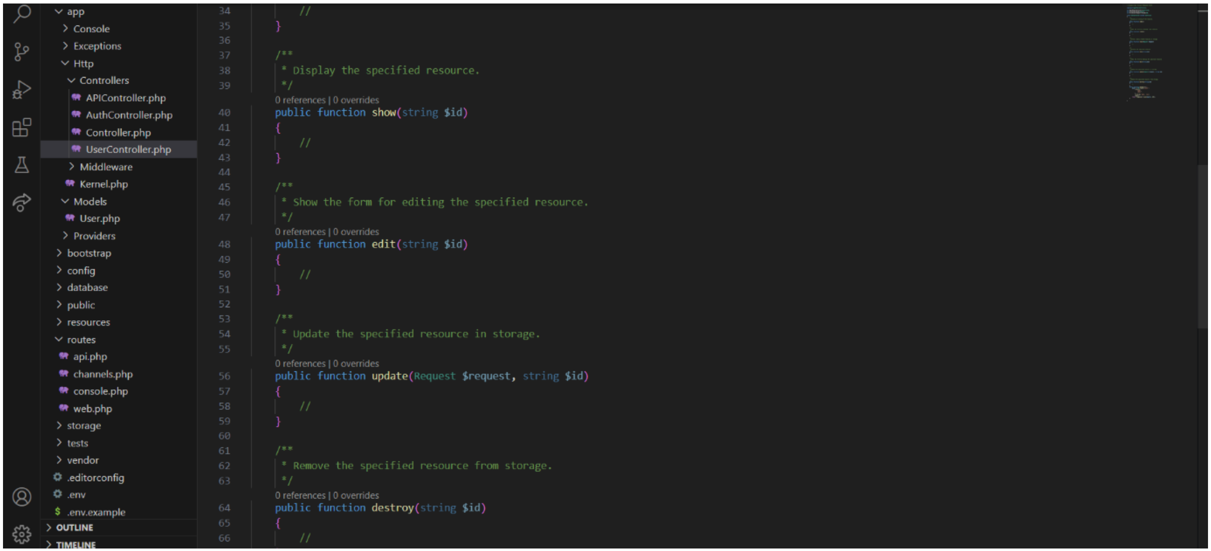


```php
<?php

// Wijdan Ardra Fulvian 6706223137 46-04

namespace App\Http\Controllers;

use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
use Illuminate\Foundation\Validation\ValidatesRequests;
use Illuminate\Routing\Controller as BaseController;

class Controller extends BaseController
{
    use AuthorizesRequests, ValidatesRequests;
}
```

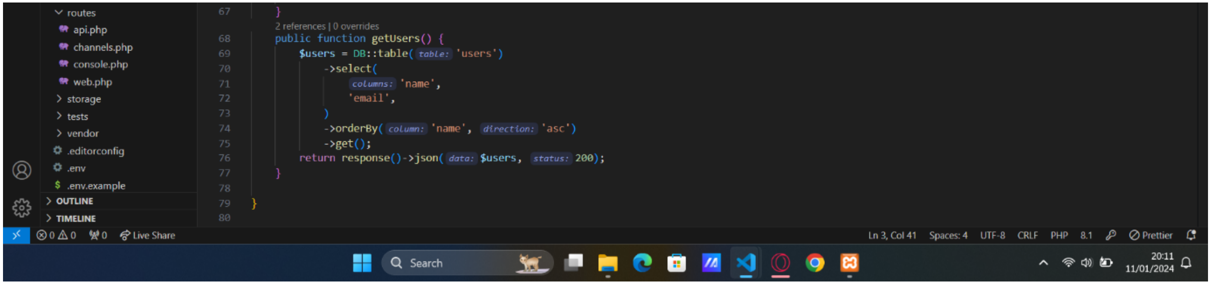
- UserController.php



```php
<?php

// Wijdan Ardra Fulvian 6706223137 46-04

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

3 references | 0 implementations
class UserController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    0 references | 0 overrides
    public function index()
    {
        //
    }

    /**
     * Show the form for creating a new resource.
     */
    0 references | 0 overrides
    public function create()
    {
        //
    }

    /**
     * Store a newly created resource in storage.
     */
    0 references | 0 overrides
    public function store(Request $request)
    {
```

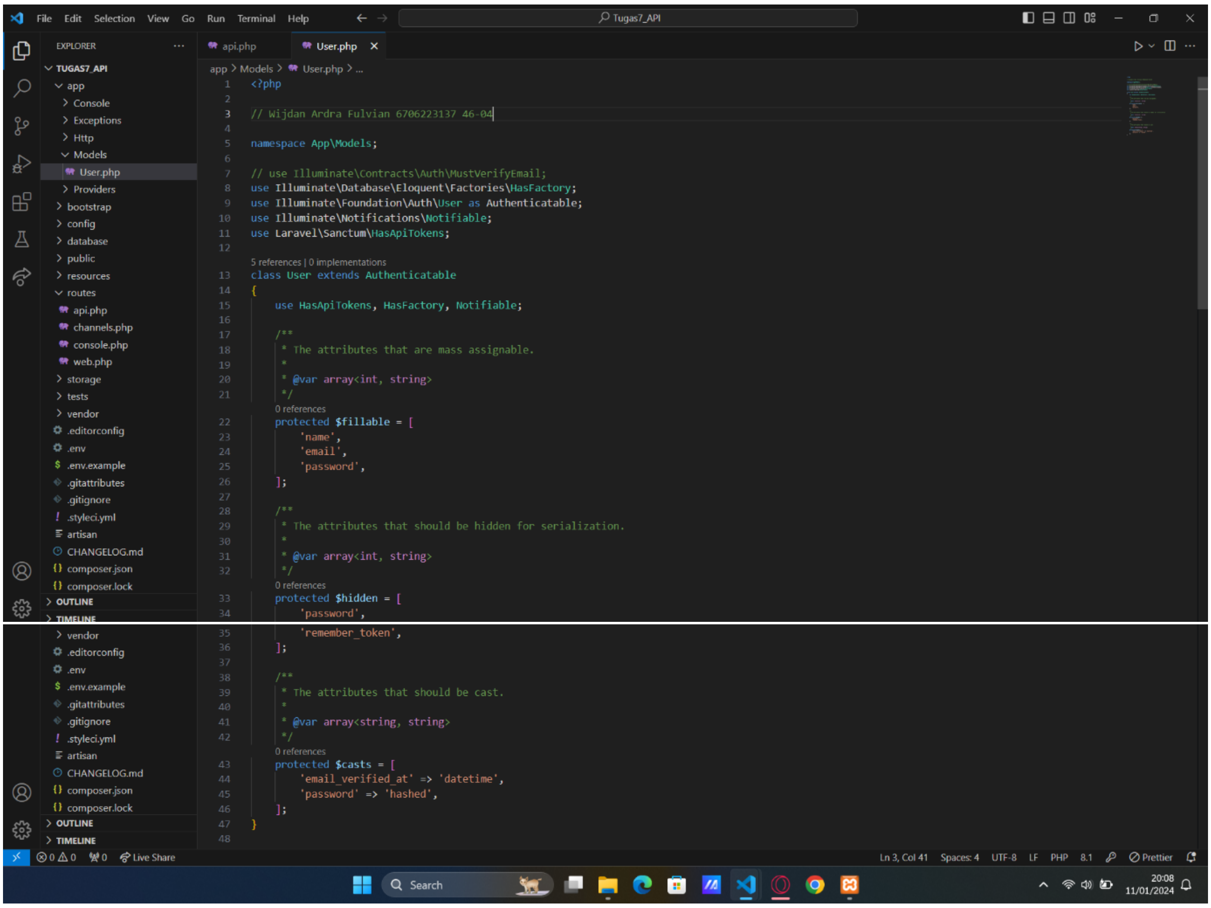


```php
        //
    }

    /**
     * Display the specified resource.
     */
    0 references | 0 overrides
    public function show(string $id)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     */
    0 references | 0 overrides
    public function edit(string $id)
    {
        //
    }

    /**
     * Update the specified resource in storage.
     */
    0 references | 0 overrides
    public function update(Request $request, string $id)
    {
        //
    }

    /**
     * Remove the specified resource from storage.
     */
    0 references | 0 overrides
    public function destroy(string $id)
    {
        //
```



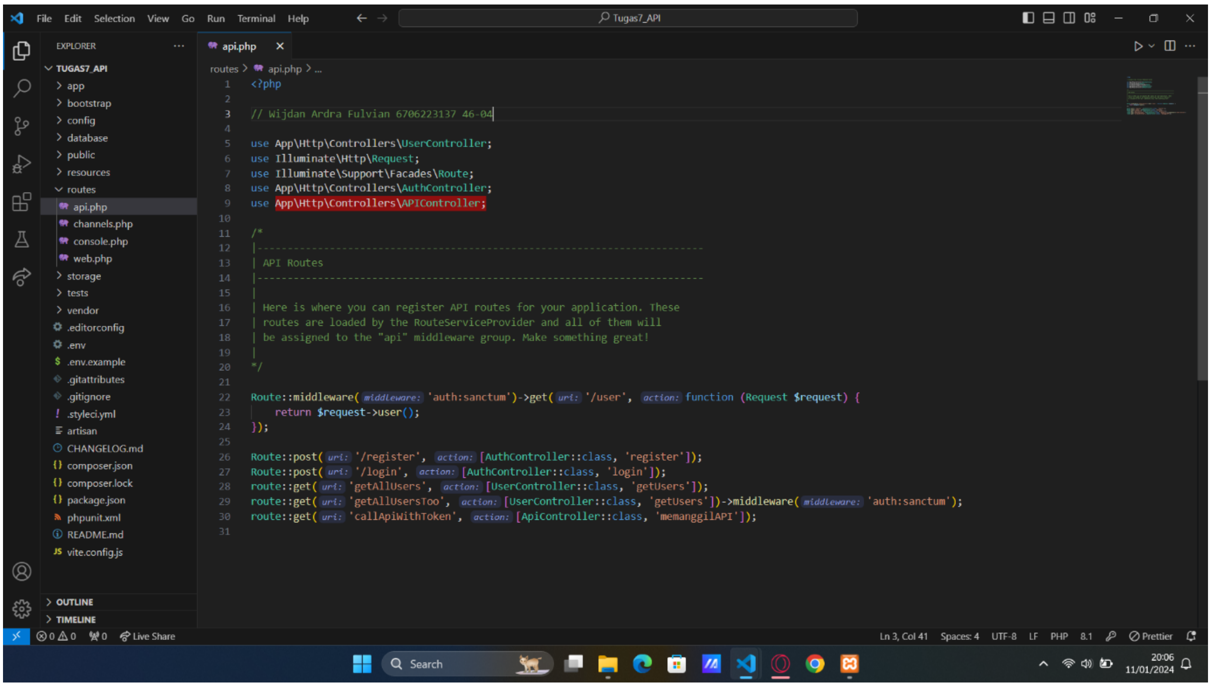
```php
    }
    2 references | 0 overrides
    public function getUsers() {
        $users = DB::table( table: 'users')
            ->select(
                columns: 'name',
                'email',
            )
            ->orderBy( column: 'name', direction: 'asc')
            ->get();
        return response()->json( data: $users, status: 200);
    }

}
```

- **Models**
  - User.php



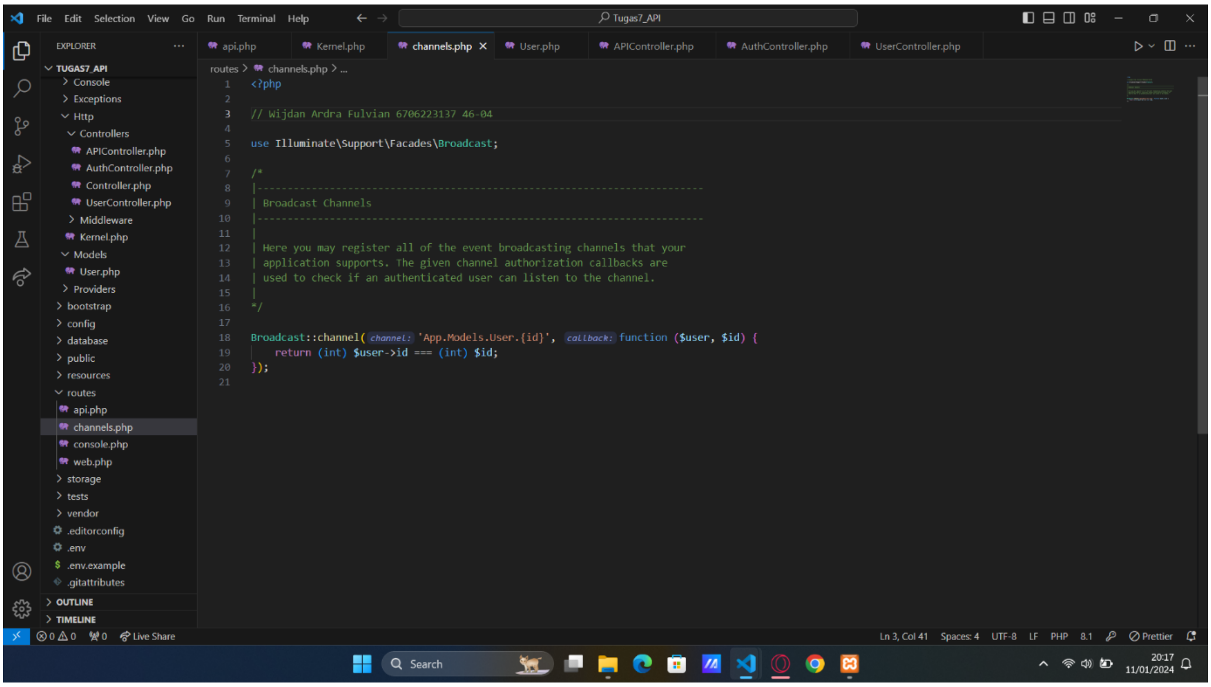
```php
<?php

// Wijdan Ardra Fulvian 6706223137 46-04

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

5 references | 0 implementations
class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    0 references
    protected $fillable = [
        'name',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    0 references
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    0 references
    protected $casts = [
        'email_verified_at' => 'datetime',
        'password' => 'hashed',
    ];
}
```

- **routes**
  - api.php



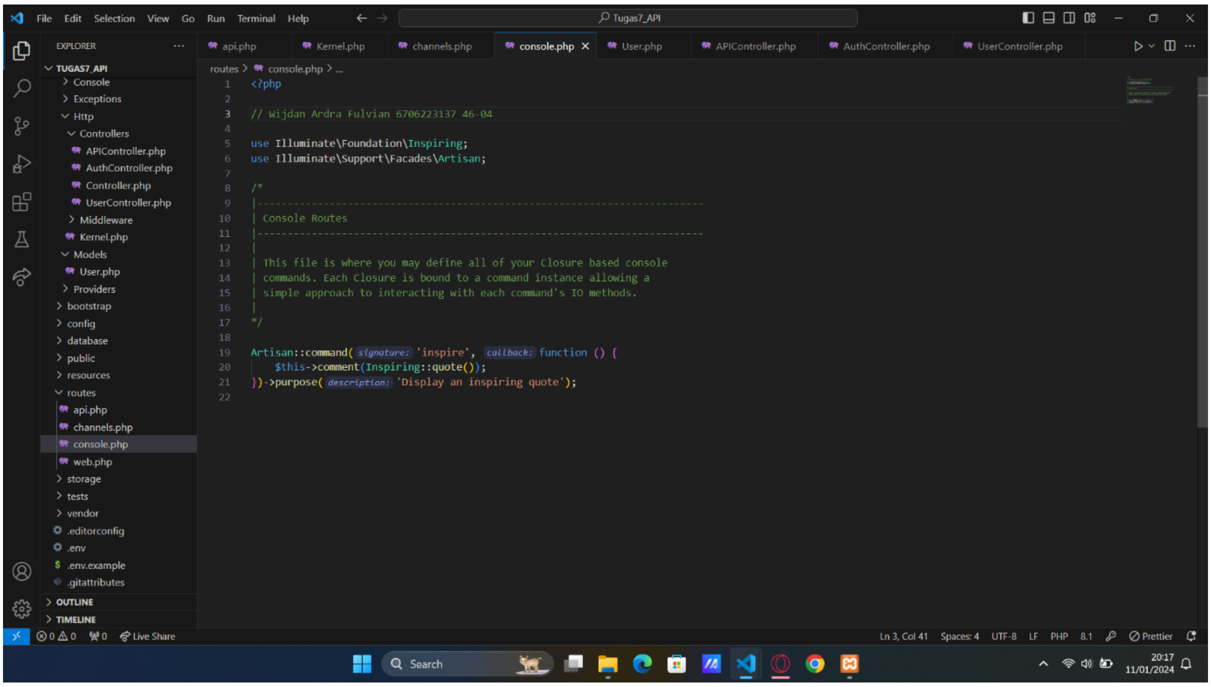
```php
<?php

// Wijdan Ardra Fulvian 6706223137 46-04

use App\Http\Controllers\UserController;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\AuthController;
use App\Http\Controllers\APIController;

/*
|--------------------------------------------------------------------------
| API Routes
|--------------------------------------------------------------------------
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider and all of them will
| be assigned to the "api" middleware group. Make something great!
|
*/

Route::middleware( middleware: 'auth:sanctum')->get( uri: '/user',  action: function (Request $request) {
    return $request->user();
});

Route::post( uri: '/register',  action: [AuthController::class, 'register']);
Route::post( uri: '/login',  action: [AuthController::class, 'login']);
route::get( uri: 'getAllUsers',  action: [UserController::class, 'getUsers']);
route::get( uri: 'getAllUsersToo',  action: [UserController::class, 'getUsers'])->middleware( middleware: 'auth:sanctum');
route::get( uri: 'callApiWithToken',  action: [ApiController::class, 'memanggilAPI']);
```

- channels.php


File Edit Selection View Go Run Terminal Help
Tugas7_API
api.php Kernel.php channels.php User.php APIController.php AuthController.php UserController.php
routes > channels.php > ...
TUGAS7_API
OUTLINE
TIMELINE
Ln 3, Col 41 Spaces: 4 UTF-8 LF PHP 8.1 Prettier
Live Share
20:17
11/01/2024

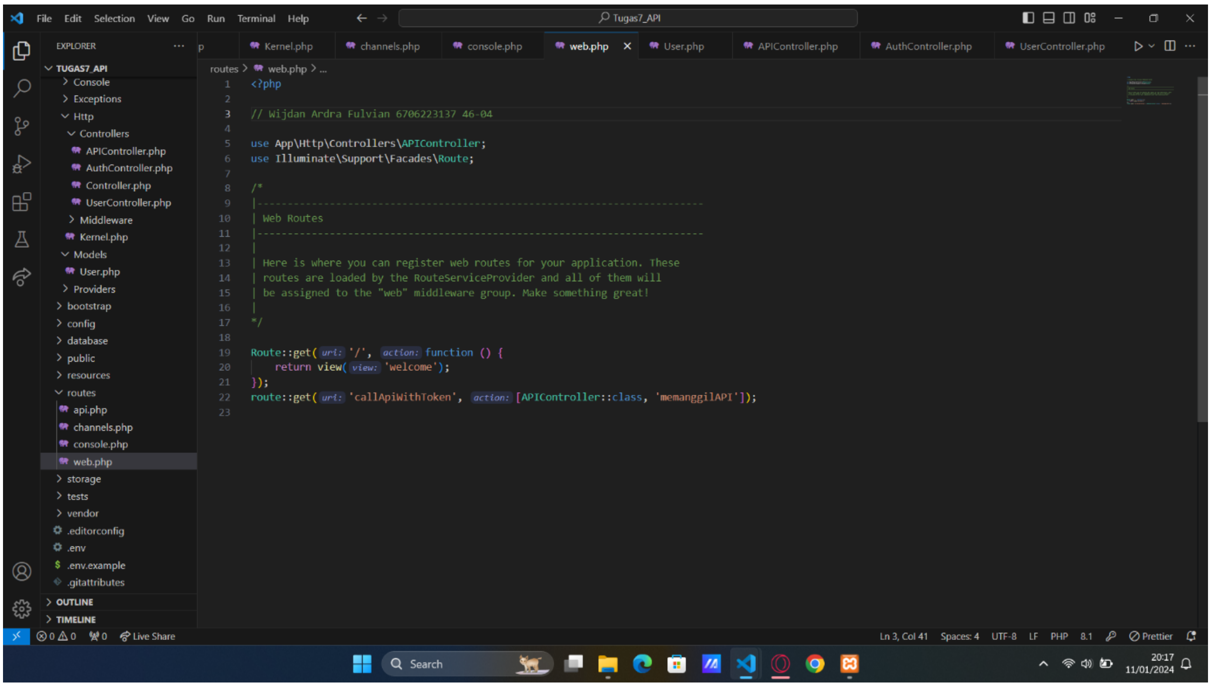
```php
<?php

// Wijdan Ardra Fulvian 6706223137 46-04

use Illuminate\Support\Facades\Broadcast;

/*
|--------------------------------------------------------------------------
| Broadcast Channels
|--------------------------------------------------------------------------
|
| Here you may register all of the event broadcasting channels that your
| application supports. The given channel authorization callbacks are
| used to check if an authenticated user can listen to the channel.
|
*/

Broadcast::channel('App.Models.User.{id}', function ($user, $id) {
    return (int) $user->id === (int) $id;
});
```

- console.php


File Edit Selection View Go Run Terminal Help
Tugas7_API
api.php Kernel.php channels.php console.php User.php APIController.php AuthController.php UserController.php
routes > console.php > ...
TUGAS7_API
OUTLINE
TIMELINE
Ln 3, Col 41 Spaces: 4 UTF-8 LF PHP 8.1 Prettier
Live Share
20:17
11/01/2024

```php
<?php

// Wijdan Ardra Fulvian 6706223137 46-04

use Illuminate\Foundation\Inspiring;
use Illuminate\Support\Facades\Artisan;

/*
|--------------------------------------------------------------------------
| Console Routes
|--------------------------------------------------------------------------
|
| This file is where you may define all of your Closure based console
| commands. Each Closure is bound to a command instance allowing a
| simple approach to interacting with each command's IO methods.
|
*/

Artisan::command('inspire', function () {
    $this->comment(Inspiring::quote());
})->purpose('Display an inspiring quote');
```

- web.php



```php
<?php

// Wijdan Ardra Fulvian 6706223137 46-04

use App\Http\Controllers\APIController;
use Illuminate\Support\Facades\Route;

/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider and all of them will
| be assigned to the "web" middleware group. Make something great!
|
*/

Route::get( uri: '/',  action: function () {
    return view( view: 'welcome');
});
route::get( uri: 'callApiWithToken',  action: [APIController::class, 'memanggilAPI']);
```