**Belajar LARAVEL WPU**

**Apa itu Laravel ?**

Laravel is a web application framework with expressive, elegant syntax.Laravel dibuat oleh **Taylor Otwell** pada Juni 2011 sebagai alternatif framework untuk Codeigniter

**Apa itu Web Application Framework ?**

* sebuah ‘kerangka’ yang didesain untuk mendukung Pembangunan aplikasi web
* Menyediakan standar / cara untuk membangun aplikasi web
* Bertujuan untuk mengotomasi hal-hal umum yang biasanya dilakukan saat membangun aplikasi web
* Contohnya : database library, templating engine, session management, authentication, security, dll.
* Perintah run laravel : php artisan serve

**Fitur utama pada Laravel :**

* MVC (Model, View, Controller)
* Templating Engine
* Artisan Console

Tempat melakukan modifikasi pada framework

* Eloquent ORM

Mempermudah berinterkasi dengan datababase yang relasional

* Authentication & Authorization
* Testing
* Packaging System
* Multiple File System
* Task Scheduling
* Webscocket Programming

**Apa itu Ekosistem Laravel ?**

Produk laravel diluar framework Laravel

**Syarat belajar PHP :**

* PHP Dasar
* Object Oriented PHP
* Konsep MVC

Valet : sebuah teknologi yang memudahkan kita mengakses project menggunakan nama, tidak menggunakan port. Setiap memanggil nama harus diakhiri dengan .test (ex: projectuas.test)

**Struktur Folder**

Jika di sesuaikan dengan konsep **MVC**

* **Model** terletak pada : app -> Models
* **View** terletak pada : resources -> Views
* **Controller** terletak pada : app -> Http -> controllers

Selain itu folder yang harus diperhatikan adalah sebagai berikut :

* Routes : folder untuk mengatur jalur nya website kita
* Public : tempat menyimpan asset statis
* File **.env** : untuk menentukan environment di laravel

Lebih baik menggunakan format templating blade pada framework laravel karena banya keunggulannya, salah satunya :

<?php echo $nama ?> menjadi {{$nama}} memiliki keunggulan yaitu dengan menggunakan blade templating otomatis ditambahkan HTML Sepcial Chars

Collection adalah pembungkus untuk sebuah array

1. **Database, Migration & Eloquent**

**Migrate**

* Perintah migrate : php artisan migrate
* Perintah menghapus migrate : php artisan migrate:rollback
* Perintah menjalankan keduanya (bertujuan jika kita menambahkan/mengurangi tabel kita yang sudah ada) :php artisan migrate:fresh

Dalam file migrations ada namanya method up() dan method down()

**up()** = sebuah method yang digunakan ketika kita ingin membikin sebuah schema / struktur tabel

**down()** = sebuah method yang digunakan untuk menghapus tabel atau kembali ke seperti sebelumnya

**Eloquent**

Merupakan ORM(Object Relational Mapper) yang dapat membuat kita dengan mudah berinteraksi dengan database yang telah dibuat. Eloquent bukan hanya mengambil data dari databse, tapi juga memungkinkan kita melakukan CRUD (Create, Read, Update, Delete) lewat codingan.

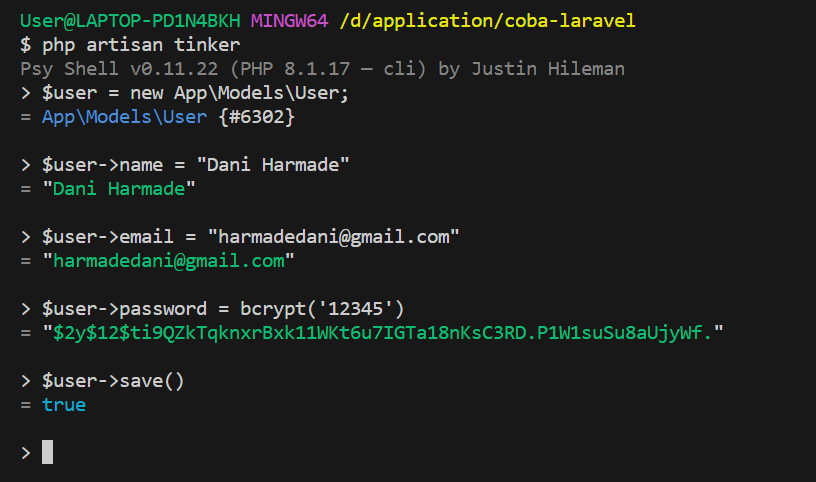
Cara masuk dan menggunakan tinker :

Perintah :

php artisan tinker

$user = new User // $user itu deklarasi, new User itu model nya

Cara mengisi data ke db tanpa memiliki interface nya menggunakan tinker :



Melihat data menggunakan tinker :



1. **Post Category & Relationship**

One to One nama relasinya = **belongsTo**

Many to One nama relasinya = **hasMany**

1. **Datababse Seeder**

Mempermudah mengisi data secara otomatis ketika melakukan migrasi (menambahkan tabel)

**Apa itu seeder?**

Sebuah fitur yang ada di laravel untuk meng generate atau mempopulasi isi dari tabel kita secara otomatis dari data testing/dumy pada saat development.

Perintah menjalankan seeder di cli : php artisan db:seed

1. **N+1 Problem**

Erat kaitannya dengan relasi / query yang kita lakukan pada project yang sedang dikerjakan

Apa itu **N+1 Problem** ?

Terjadi ketika aplikasi kita mengambil data dari database, dimana didalamnya kita melakukan looping di dalam data kita.

Contohnya ketika kita mau mengambil 20 data dari database, maka kita akan melooping query nya sebanyak 40 kali untuk mengambil 20 data tersebut, untuk mengatasi itu kita dapat menggunakan sebuah library yang sudah tersedia yaitu **clockwork**

**Link repo /instalasi** : https://github.com/itsgoingd/clockwork

Contoh:

**Tanpa menggunakna binding**

Tanpa menggunakan clockwork library

Syntax :

public function index()

{

return view('posts', [

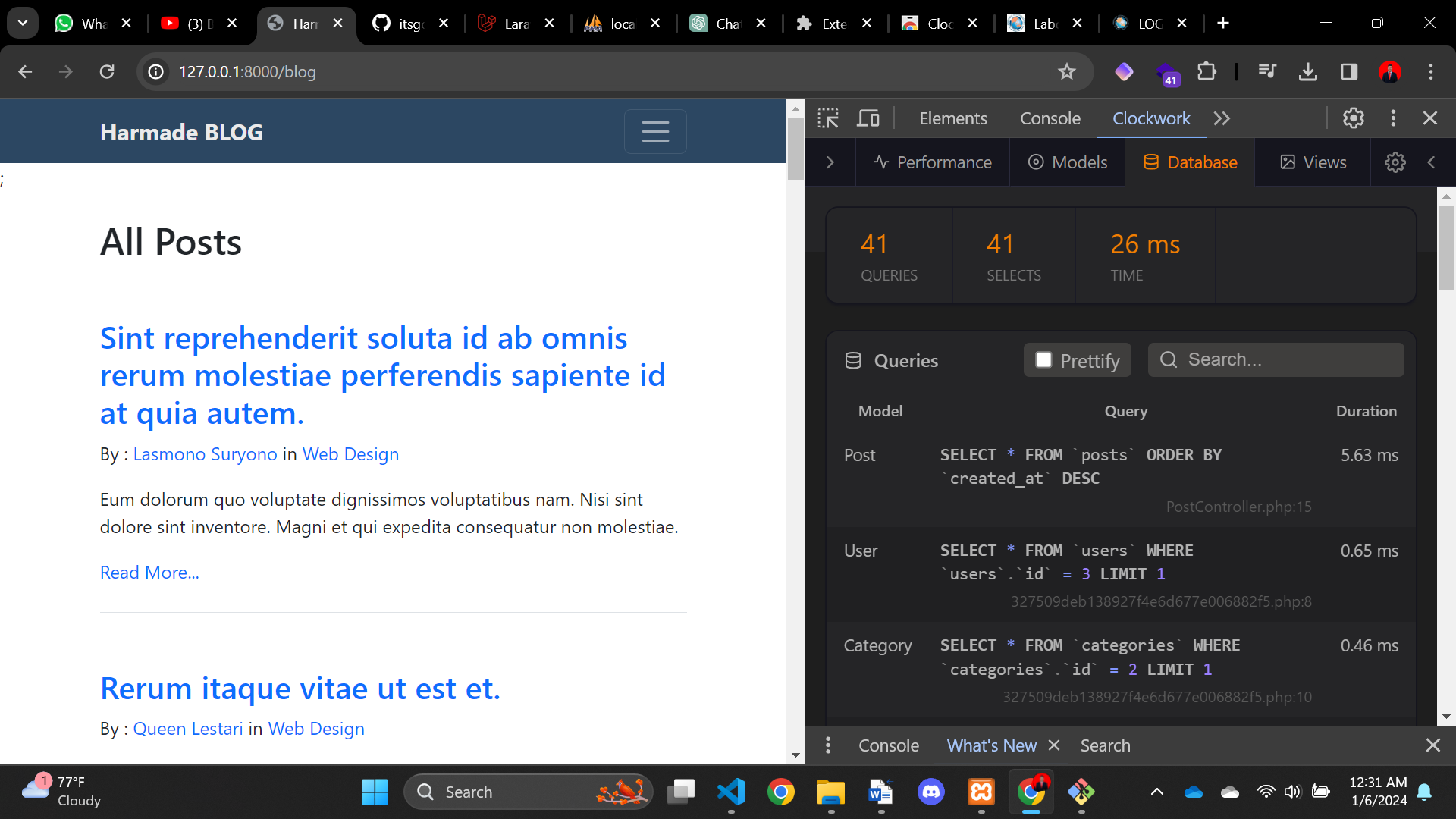
"title" => "All Posts",

// "posts" => Post::all()

"posts" => Post::latest()->get()

]);

}



Menggunakan clockwork library :

Syntax :

public function index()

{

return view('posts', [

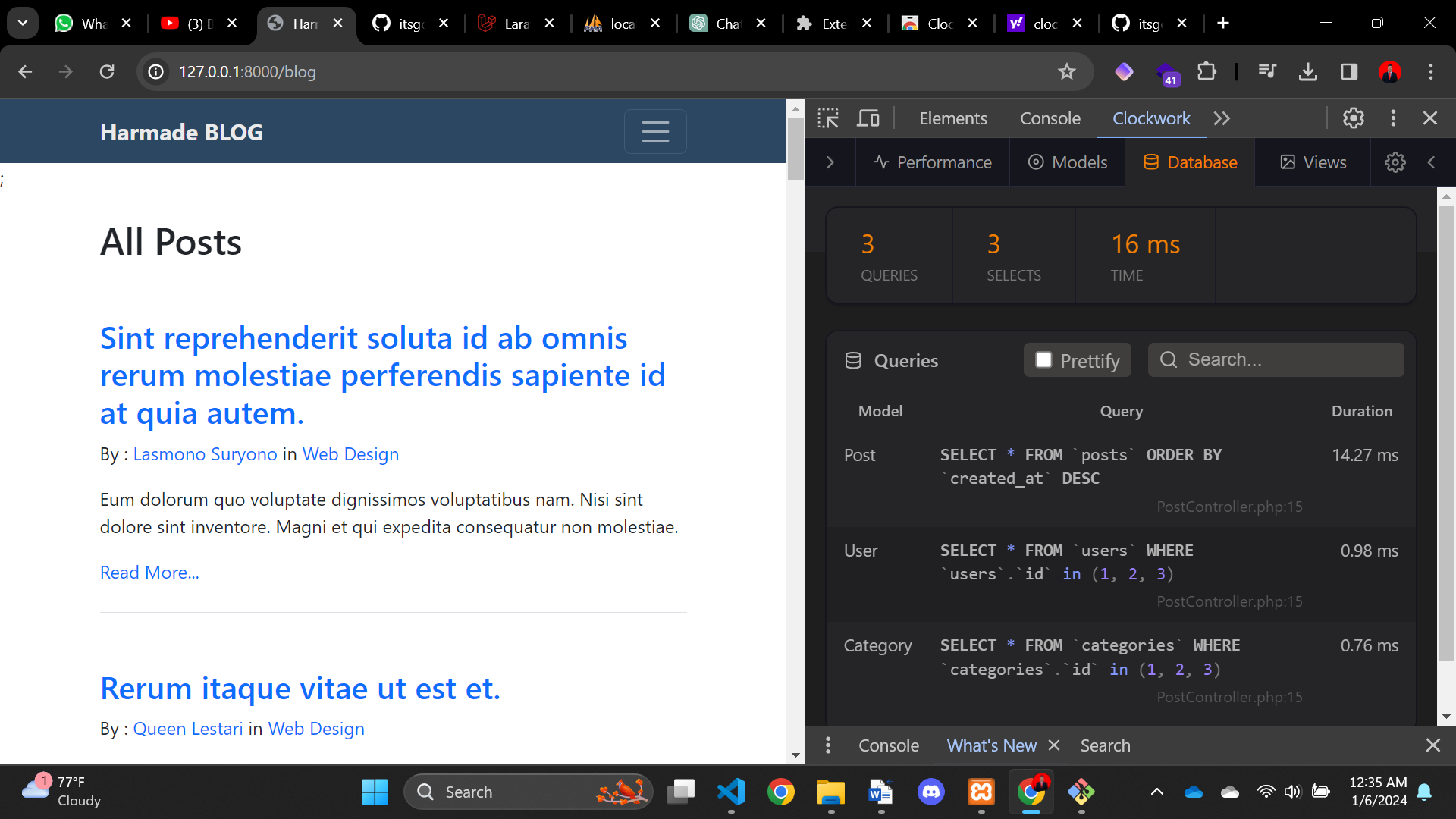
"title" => "All Posts",

// "posts" => Post::all()

"posts" => Post::with('author', 'category')->latest()->get()

]);

}



**Menggunakan binding**

Jika dia menggunakan binding tidak bisa menggunkana with(), jadi kita menggunakan load(). Contoh :

Tanpa library:

Route::get('/authors/{author:username}', function (User $author) {

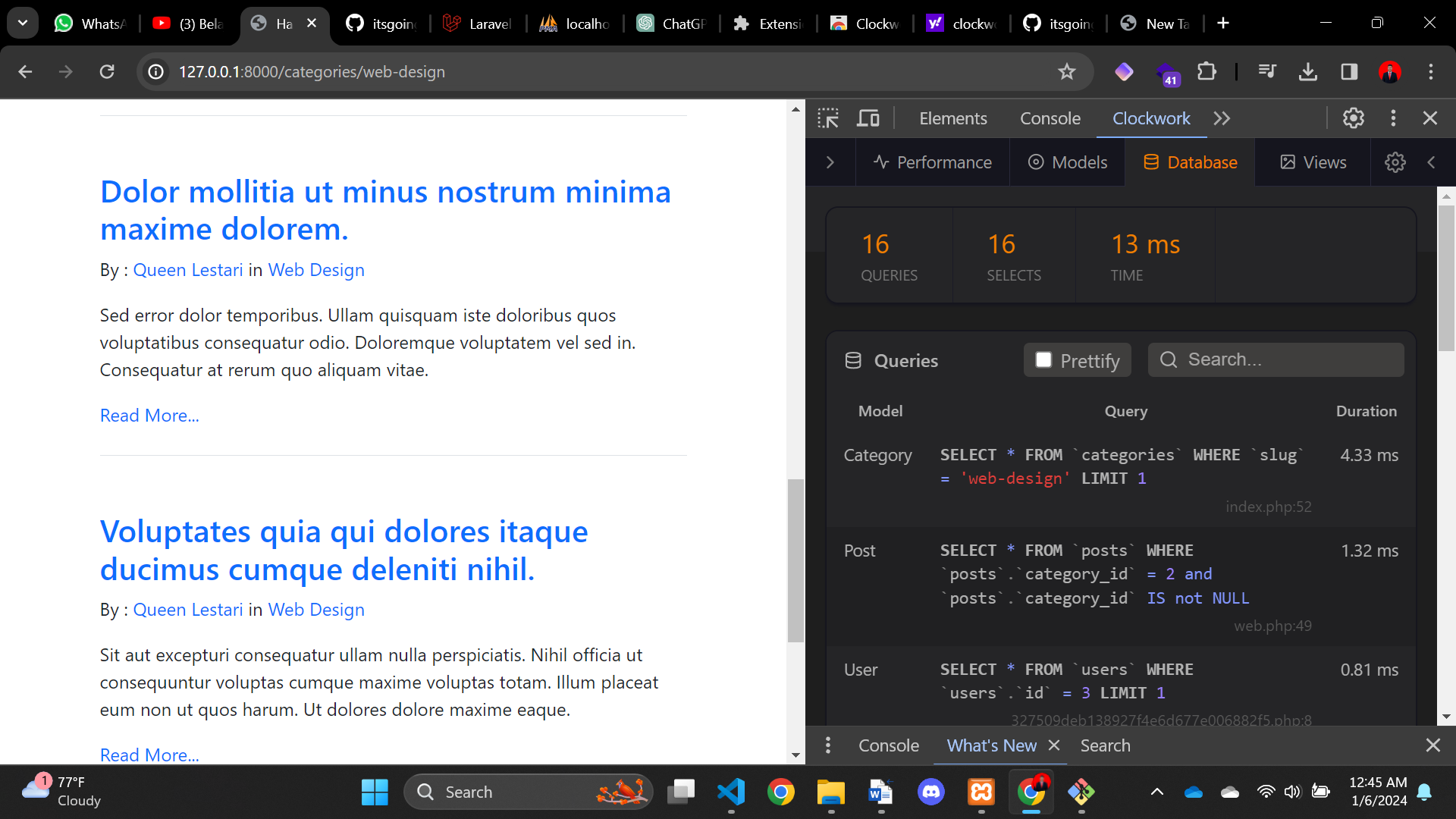
return view('posts', [

'title' => "Post by Author : $author->name",

'posts' => $author->posts->

]);

});



Menggunakan library :

Script :

Route::get('/authors/{author:username}', function (User $author) {

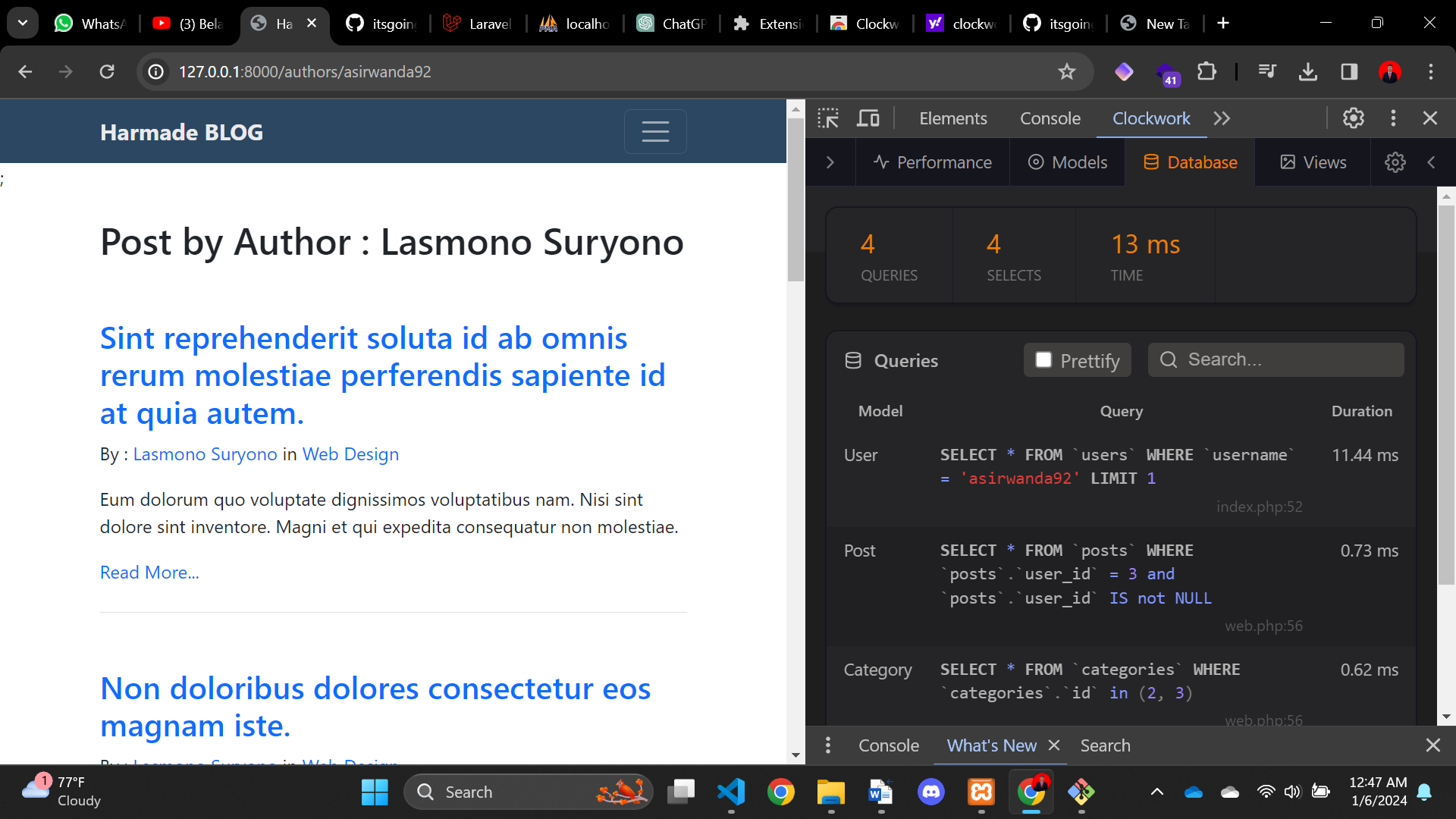
return view('posts', [

'title' => "Post by Author : $author->name",

'posts' => $author->posts->load('category', 'author')

]);

});

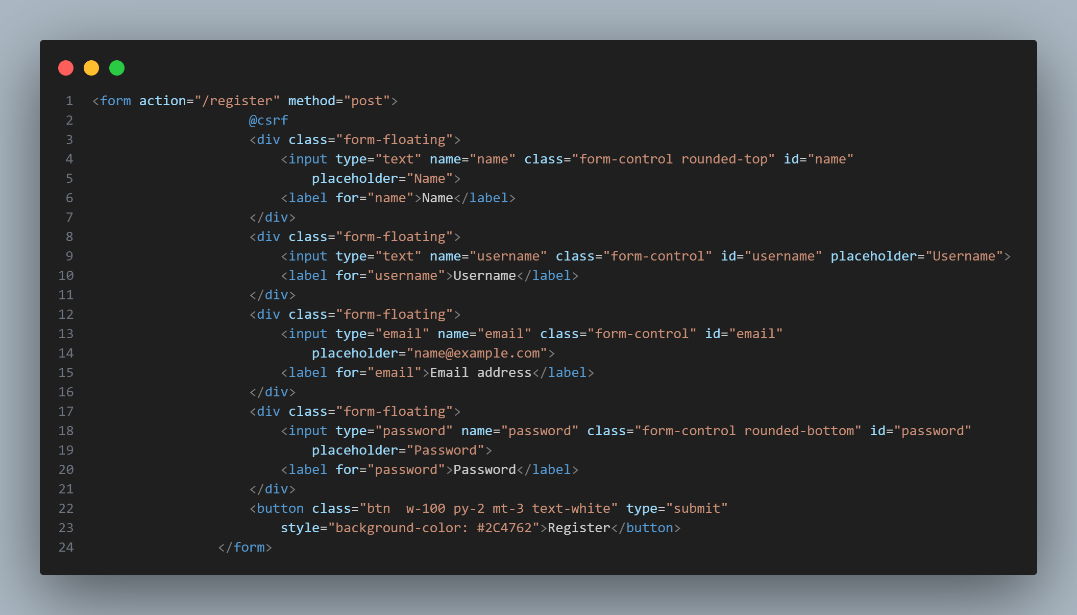


Library untuk menampilkan waktu agar mudah dibaca oleh manusia :

diffForHumans()

1. **Searching dan Pagination**
2. **Login dan Registrasi**

Untuk mengamankan sistem kita dari serangan CSRF (Cross-Site-Request-Forgery) laravel sudah menyidakan sebuah fitur yaitu dengan menambahkan @csrf pada tag form kita



1. **User Login & Middleware**

Laravel menyediakan yang namanya Starter Kits semacam plugin yang menangani masalah autentikasi

Middleware menyediakan sebuah mekanisme yang memudahkan kita untuk melakukan inspeksi dan filtering pada http request, jadi request ktia di filter dulu yh

1. **Dashboard UI**

**Resource Controller**

Laravel menyediakan sebuah route khusus untuk memudahkan kita mengelola controller untuk melakukan CRUD yang dinamakan Resource Controller dengan cara menuliskan pada cmd vscode kita :

php artisan make:controller –resource

1. **Create Post Form**

Route default dari laravel untuk menambahkan data : create

Membuat slug otomatis berdasarkan judul post :\

1. **Upload Image**

Untuk menghubungkan folder public di storage dgn folder public di aplikasi kita menggunakan “symbolic link” caranya tulis diteriminal :

php artisan storage:link

1. **Preview Gambar**

Cara membuat agar ketika kita mengedit gambar, gambar yang lama hilang dri database kita agar tidak membebani db :

Panggil library Storage :

use Illuminate\Support\Facades\Storage;

lakukan pengecekan dan penghapusan gambar lama serta mengupload gambar baru :

if ($request->file('image')) {

if ($request->oldImage) {

Storage::delete($request->oldImage);

}

$validatedData['image'] = $request->file('image')->store('post-images');

}

1. **Authorization**

Cara membuat hak akses admin dan user

* Buat Middleware kita sendiri

php artisan make:middleware

* buat nama middleware ex : is\_admin
* dalam file middleware is\_admin lakukan konfigurasi ex :

public function handle(Request $request, Closure $next): Response

{

if (auth()->guest() || auth()->user()->username !== 'daniharmade') {

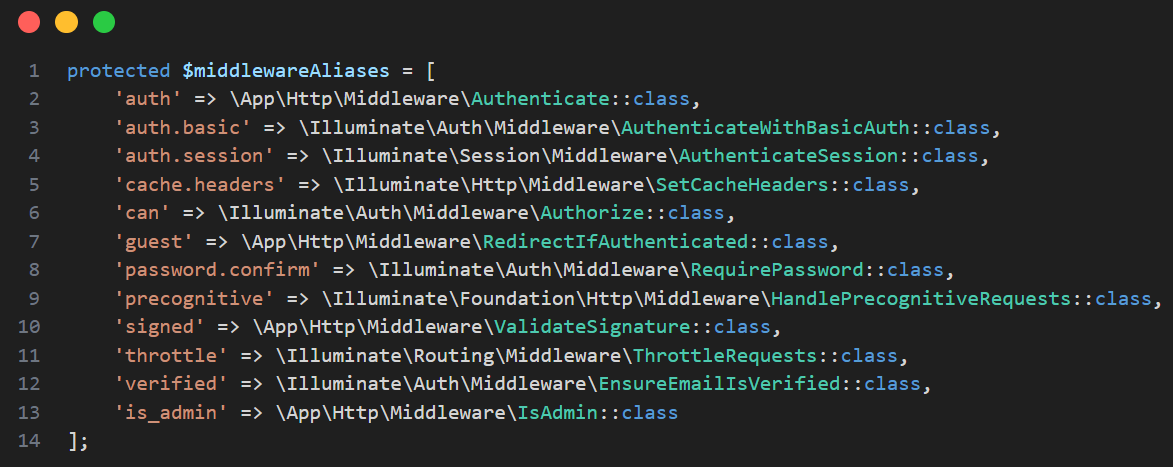
abort(403);

}

return $next($request);

}

* masuk ke karnel.php lalu buat middleware kita sendiri



Middleware yang saya buat adalah ‘is\_admin’

* Tambahkan middleware pada routes yang sudah kita buat

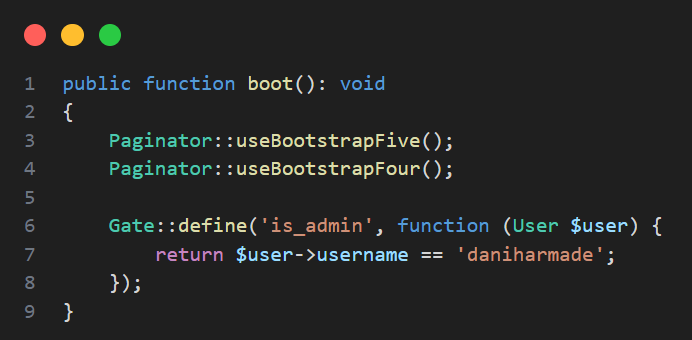
Route::resource('/dashboard/categories', AdminCategoryController::class)->except('show')->middleware('is\_admin');

Nah itu tadi untuk membatasi agar selain admin tidak bisa mengakses halaman tertentu, lalu gmn cara agar menu tersebut hilang dari interface selain admin? Simak cara berikut yhh adikks”

Dalam melakukan ini dinamakan autorisasi menggunakan sebuah library dari laravel yaitu **gate**

Buat gate nya di folder **provider** lalu file nya **AppService :**

dekat fn boot nya buat Gate :



Lalu di sidebarnya masukkan kedalam fungsi blade yaitu **can** untuk element yang akan muncul dengan gate yang telah dibuat tadi, contoh script :

**@can('is\_admin')**

<h6

class="sidebar-heading d-flex justify-content-between align-items-center px-3 mt-4 mb-1 text-muted fs-6">

<span>ADMINISTRATOR</span>

</h6>

<ul class="nav flex-column">

<li class="nav-item">

<a class="nav-link d-flex align-items-center gap-2 {{ Request::is('dashboard/categories\*') ? 'active' : '' }} "

aria-current="page" href="/dashboard/categories">

<svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24"

fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round"

stroke-linejoin="round" class="feather feather-list">

<line x1="8" y1="6" x2="21" y2="6"></line>

<line x1="8" y1="12" x2="21" y2="12"></line>

<line x1="8" y1="18" x2="21" y2="18"></line>

<line x1="3" y1="6" x2="3.01" y2="6"></line>

<line x1="3" y1="12" x2="3.01" y2="12"></line>

<line x1="3" y1="18" x2="3.01" y2="18"></line>

</svg>

Post Categories

</a>

</li>

</ul>

**@endcan**

**Alhamdulillah DONE Laravel di 22 Januari 2024, Next Time Lebih Semangat Lagi :)**