**Zoekfunctionaliteit**

Voor deze opdracht gaan we in een tutorial beschrijven hoe we de zoekfunctionaliteit gaan implementeren in ons Laravel project. We beschrijven duidelijk in welke bestanden we gewerkt hebben en hoe de code werkt.

Om te beginnen gaan we in de **posts.php** de zoekbar aanpassen. Vervang deze door deze code:

<!-- Search -->

            <div *class*="relative flex lg:inline-flex items-center bg-gray-100 rounded-xl px-3 py-2">

                <form *method*="GET" *action*="{{ route('search') }}" *class*="flex">

                    <input *type*="text" *name*="value" *placeholder*="Find something"

*class*="bg-transparent font-semibold text-sm focus:outline-none" *value*="@if (strlen(request()->get('value')) > 0) {{ request()->get('value') }} @endif"  />

                    <select *id*="category" *name*="category" *class*="rounded-xl text-sm focus:outline-none bg-transparent cursor-pointer">

                        <option *value*="everything" *@if* *(* *request()-*>get('category') == 'everything' || request()->get('category') == '' ) selected @endif>Everything</option>

                        <option *value*="title" *@if* *(* *request()-*>get('category') == 'title') selected @endif>Title</option>

                        <option *value*="author" *@if* *(* *request()-*>get('category') == 'author') selected @endif>Author</option>

                        <option *value*="body" *@if* *(* *request()-*>get('category') == 'body') selected @endif>Body</option>

                      </select>

                </form>

                {{-- <span *class*="text-red-700 text-xs">The value is required for searching</span> --}}

            </div>

Daarna gaan we in de **web.php** een route aanmaken. Deze route verwijst naar een controller en de functie hiervan.

*Route*::get('/search', [*PostController*::class, 'search'])->name('search');

Dan ga je naar **app/http/controllers**. Maak hier een bestand genaamt **PostController.php** aan.

Hier krijgen we de if-statement die de checken welke categorie er is aangeklikt en ervoor zorgen dat er in de juiste colommen gekeken wordt. Als dit gebeurt is wordt de listings view op het scherm teruggezet er wordt hier een array met alle gefilterde posts aan meegegeven. Daarna loopen we met foreach loops over de teruggegeven loops-array heen en geven deze weer in de listings view.

Zet hierin de volgende code:

<?php

namespace *App\Http\Controllers*;

use *App\Models\Listing*;

use *Illuminate\Http\Request*;

class PostController extends *Controller*

{

    public static function index()

    {

        $blogs = *Listing*::all();

        return view('listings', [

            'listings' => $blogs

        ]);

    }

    public static function search(*Request* $request)

    {

        // IN DE REQUEST VARIABLE ZITTEN ALLE FORM VALUES

        $inputValue = $request->value; //kaas

        $dropdownValue = $request->category; //title, author, body

        if ($inputValue == '' || $dropdownValue == '') {

            return redirect()->route('blogs');

        }

        if ($dropdownValue == "everything") {

            $names = ['introduction', 'title', 'author'];

        } else if ($dropdownValue == "title") {

            $names = ['title', 'body\_title'];

        } else if ($dropdownValue == "author") {

            $names = ['author', 'author\_status'];

        } else if ($dropdownValue == "body") {

            $names = ['introduction', 'body', 'conclusion'];

        }

        $query = *Listing*::where($names[0], 'LIKE', '%' . $inputValue . '%');

        for ($i = 1; $i < (count($names) - 1); $i++) {

            $query->orWhere($names[$i], 'LIKE', '%' . $inputValue . '%');

        }

        $blogsNew = $query->get();

        return view('listings', [

            'listings' => $blogsNew

        ]);

    }

    public function viewblog(*Listing* $listing)

    {

        return view('listing', [

            'listing' => $listing

        ]);

    }

}