**Laravel Cheat Sheet**

**COMMANDS**

* php artisan serve –> *run Laravel site*
* php artisan make:model Listing –> *make model*
* php artisan make:migration create\_listings\_table –> *create a migration file*
* php artisan migrate –> *run to create all table in migrations*
* php artisan migrate:refresh –> *refresh table (removes all existing data)*
* php artisan migrate:refresh --seed –> *refresh table and seed dummy data*
* php artisan db:seed -> *creating dummy data from seeder/Database Seeder.php*
* php artisan make:factory ListingFactory –> *create factory (used for dummy data creator)*
* php artisan make:controller ListingController –> *make controller*
* php artisan storage:link -> *shares storage/app/public’s resources publicly*
* php artisan tinker -> *open tinker terminal (run model functions and relationship), use “exit” to stop process*
  + \App\Models\Listing::find(3)->user
  + \App\Models\User::find(1)->listing

**ROUTES**

// Common Resource Routes:

// index - Show all listings

// show - Show single listing

// create - Show form to create new listing

// store - Store new listing

// edit - Show form to edit listing

// update - Update listing

// destroy - Delete listing

**CREATE TABLE IN MIGRATION**

Schema::create('listings', function (Blueprint $table) {

            $table->id();

            $table->string('title');

            $table->string('tags');

            $table->string('company');

            $table->string('location');

            $table->string('email');

            $table->string('website');

            $table->string('description');

            $table->timestamps();

        });

**CREATE FAKE DATA IN FACTORY**

return [

            'title' => $this->faker->sentence(),

            'tags' => 'laravel, api, backend',

            'company' => $this->faker->company(),

            'email' => $this->faker->companyEmail(),

            'website' => $this->faker->url(),

            'location' => $this->faker->city(),

            'description' => $this->faker->paragraph(3),

        ];

**ADDING FILTER TO MODEL**

class Listing extends Model

{

    use HasFactory;

    protected $fillable = ['title', 'company', 'location', 'website', 'email', 'tags', 'description'];

    public function scopefilter($query, array $filters)

    {

        if ($filters['tag'] ?? false) {

            $query->where('tags', 'like', '%' . request('tag') . '%');

        }

        if ($filters['search'] ?? false) {

            $query->where('title', 'like', '%' . request('search') . '%')

                ->orWhere('description', 'like', '%' . request('search') . '%')

                ->orWhere('tags', 'like', '%' . request('search') . '%');

        }

    }

}