Laravel Project Cheat Sheet

1. Create a New Laravel Project

```
laravel new myproject
composer create-project laravel/laravel myproject
cd myproject php artisan serve
```

2. Environment Setup

```
cp .env.example .env
php artisan key:generate
Edit .env:
DB_CONNECTION=mysql
DB_DATABASE=myproject_db
DB_USERNAME=root
DB_PASSWORD=
```

3. Create Model, Migration, and Controller

```
php artisan make:model Product -mcr
# -m (migration), -c (controller), -r (resource)
```

4. Migration Example

```
Schema::create('products', function (Blueprint $table) {
        $table->id(); $table->string('name');
        $table->text('description')->nullable();
        $table->decimal('price', 10, 2);
        $table->integer('stock')->default(0);
        $table->timestamps();
});
```

```
php artisan migrate
```

5. Model Example

```
class Product extends Model {
        use HasFactory; protected $fillable =
        ['name','description','price','stock'];
}
```

6. Controller Example

```
namespace App\Http\Controllers;

use App\Models\Product;
use Illuminate\Http\Request;

class ProductController extends Controller
{
    public function index()
    {
        return Product::all();
    }

    public function store(Request $request)
    {
        $data = $request->validate([
            'name' => 'required|string',
            'description' => 'nullable|string',
            'price' => 'required|numeric',
            'stock' => 'integer'
        ]);

        return Product::create($data);
    }

    public function show(Product $product)
    {
        return $product;
    }

    public function update(Request $request, Product $product)
    {
        $product->update($request->all());
        return $product;
    }

    public function destroy(Product $product)
    {
        $product->delete();
        return response()->noContent();
    }
}
```

7. Seeder

a. Create seeder

```
php artisan make:seeder ProductSeeder
```

b. edit database/seeders/ProductSeeder

```
namespace Database\Seeders;

use Illuminate\Database\Seeder;
use App\Models\Product;
```

```
class ProductSeeder extends Seeder
{
    public function run(): void
    {
        Product::factory()->count(10)->create();
    }
}
```

c. Register seeder in DatabaseSeeder.php

```
public function run(): void
{
    $this->call(ProductSeeder::class);
}
```

d. Run seeder

```
php artisan db:seed
```

8. Factory

a. Create factory

```
php artisan make:factory ProductFactory --model=Product
```

b. Edit database/factories/ProductFactory.php

```
namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;

class ProductFactory extends Factory
{
    public function definition(): array
    {
        return [
            'name' => fake()->word(),
            'description' => fake()->sentence(),
            'price' => fake()->randomFloat(2, 10, 500),
            'stock' => fake()->numberBetween(0, 100)
        ];
    }
}
```

c. Test generating fake data

```
php artisan tinker
>>> App\Models\Product::factory()->count(5)->create();
```

9. Routing

```
Route::apiResource('products', ProductController::class);
```

10. Refresh & Reseed

```
php artisan migrate:fresh --seed
```

Quick Reference Commands

Task Command

- Create project: composer create-project laravel/laravel appname
- Make model: php artisan make:model ModelName
- Make controller: php artisan make:controller NameController
- Make migration: php artisan make:migration create_table_name
- Run migration: php artisan migrate
- Rollback migration: php artisan migrate:rollback
- Make seeder: php artisan make:seeder SeederName
- Make factory: php artisan make:factory FactoryName --model=ModelName
- Seed database: php artisan db:seed
- Rerun all php artisan migrate:fresh --seed