**KHOA CNTT – TRƯỜNG ĐẠI HỌC THỦY LỢI**

Học phần : **Công nghệ Web**

Khóa : **K62**

**HƯỚNG DẪN**

**BÀI TẬP THỰC HÀNH SỐ 05**

**Bước 1**: Tạo Project

- Nếu ko cài đặt Laravel Installer:

composer create-project --prefer-dist laravel/laravel music-project

- Nếu máy đã có Laravel Installer (không cần mạng):

laravel new music-project

**Bước 2**: Tạo CSDL. Giả sử đặt tên là music

**Bước 3**: Chỉnh sửa tệp .env trong dự án để khai báo kết nối CSDL

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

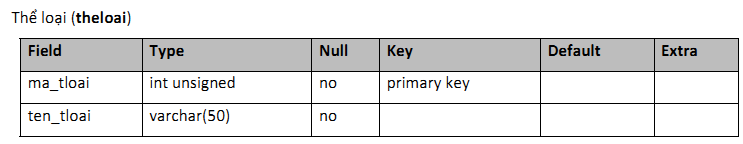
DB\_DATABASE=music

DB\_USERNAME=root

DB\_PASSWORD=

**Bước 4:** Tạo bảng MySQL bằng Laravel Migration

* **Tạo bảng thể loại và tác giả trước vì nó chỉ chứa khóa chính, không có khóa ngoại**
* **Tạo bảng Thể loại (theloai)**

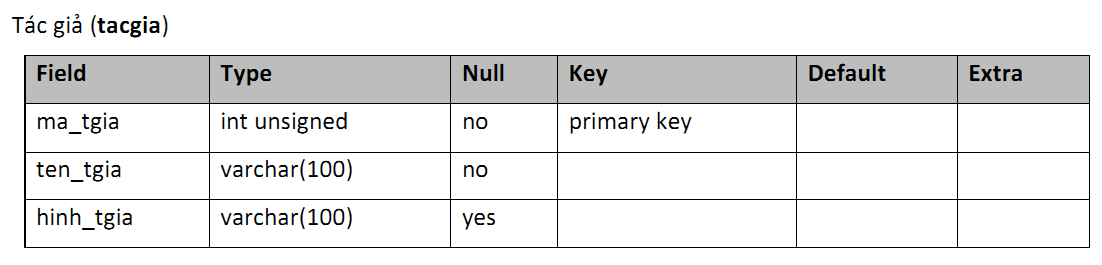


php artisan make:migration create\_theloai\_table

**Kết quả**: Một tệp tin mới có tên: 2023\_06\_06\_032602\_create\_theloai\_table.php được tạo ra trong thư mục database/migrations

|  |
| --- |
| <?php  use Illuminate\Database\Migrations\Migration;  use Illuminate\Database\Schema\Blueprint;  use Illuminate\Support\Facades\Schema;  return new class extends Migration  {  /\*\*  \* Run the migrations.  \*  \* @return void  \*/  public function up()  {  Schema::create('theloai', function (Blueprint $table) {  $table->increments('ma\_tloai');  $table->string('ten\_tloai',50);  // $table->timestamps();  });  }  /\*\*  \* Reverse the migrations.  \*  \* @return void  \*/  public function down()  {  Schema::dropIfExists('theloai');  }  }; |

* **Tạo bảng Tác giả (tacgia)**

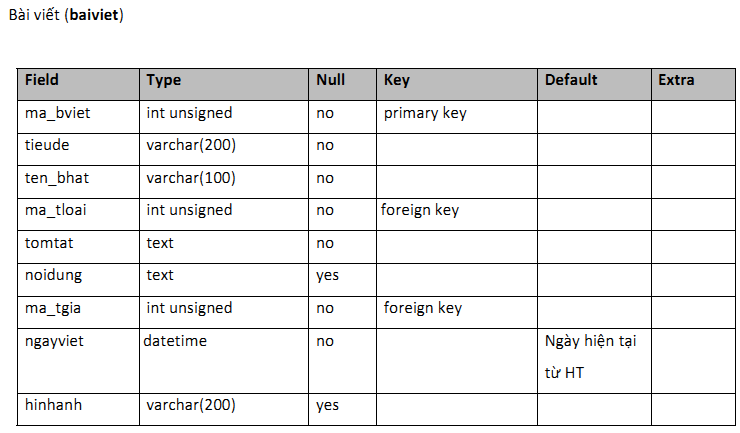


php artisan make:migration create\_tacgia\_table

**Kết quả**: Một tệp tin mới có tên: 2023\_06\_06\_032602\_create\_tacgia\_table.php được tạo ra trong thư mục database/migrations

|  |
| --- |
| <?php  use Illuminate\Database\Migrations\Migration;  use Illuminate\Database\Schema\Blueprint;  use Illuminate\Support\Facades\Schema;  return new class extends Migration  {  /\*\*  \* Run the migrations.  \*  \* @return void  \*/  public function up()  {  Schema::create('tacgia', function (Blueprint $table) {  $table->increments('ma\_tgia');  $table->string('ten\_tgia',100);  $table->string('hinh\_tgia',100);  // $table->timestamps();  });  }  /\*\*  \* Reverse the migrations.  \*  \* @return void  \*/  public function down()  {  Schema::dropIfExists('tacgia');  }  }; |

* **Tạo bảng Bài viết (baiviet)**



php artisan make:migration create\_baiviet\_table

**Kết quả**: Một tệp tin mới có tên: 2023\_06\_06\_032602\_create\_baiviet\_table.php được tạo ra trong thư mục database/migrations

|  |
| --- |
| <?php  use Illuminate\Database\Migrations\Migration;  use Illuminate\Database\Schema\Blueprint;  use Illuminate\Support\Facades\Schema;  return new class extends Migration  {  /\*\*  \* Run the migrations.  \*  \* @return void  \*/  public function up()  {  Schema::create('baiviet', function (Blueprint $table) {  $table->increments('ma\_bviet');  $table->string('tieude',200);  $table->string('ten\_bhat',100);  $table->unsignedInteger('ma\_tloai');  $table->text('tomtat');  $table->text('noidung')->nullable();  $table->unsignedInteger('ma\_tgia');  $table->dateTime('ngayviet')->default(DB::raw('CURRENT\_TIMESTAMP'));  $table->string('hinhanh',200)->nullable();  $table->foreign('ma\_tloai')->references('ma\_tloai')->on('theloai')->onDelete('cascade');  $table->foreign('ma\_tgia')->references('ma\_tgia')->on('tacgia')->onDelete('cascade');  // $table->timestamps();  });  }  /\*\*  \* Reverse the migrations.  \*  \* @return void  \*/  public function down()  {  Schema::dropIfExists('baiviet');  }  }; |

**Bước 6**: Thực hiện lệnh sau để thực hiện để chạy script tạo Bảng vào CSDL đã thiết lập:

php artisan migrate

**Bước 7**: Tạo model

* **Tạo model Category**

php artisan make:model **Category**

**app/Model/Category.php**

|  |
| --- |
| <?php  namespace App\Models;  use Illuminate\Database\Eloquent\Factories\HasFactory;  use Illuminate\Database\Eloquent\Model;  class Category extends Model  {  protected $table = 'theloai';  use HasFactory;  } |

* **Tạo model Author**

php artisan make:model **Author**

**app/Model/Author.php**

|  |
| --- |
| <?php  namespace App\Models;  use Illuminate\Database\Eloquent\Factories\HasFactory;  use Illuminate\Database\Eloquent\Model;  class Author extends Model  {  protected $table='tacgia';  use HasFactory;  } |

* **Tạo model Article**

php artisan make:model **Article**

**app/Model/Article.php**

|  |
| --- |
| <?php  namespace App\Models;  use Illuminate\Database\Eloquent\Factories\HasFactory;  use Illuminate\Database\Eloquent\Model;  class Article extends Model  {  protected $table='article';  use HasFactory;  } |

**Bước 8:** Gieo dữ liệu với Faker và Seeder

Faker là một thư viện PHP tạo dữ liệu giả cho bạn (tương tự như mockaroo đã được giới thiệu). Thật hữu ích khi bạn cần điền dữ liệu vào cơ sở dữ liệu cho mục đích thử nghiệm.Bạn có thể sử dụng nó kết hợp với các trình tạo cơ sở dữ liệu của Laravel, là các lớp chứa một phương thức để chèn dữ liệu vào cơ sở dữ liệu.

🡺 Thư viện Faker được cài sẵn trong Laravel 8 trở đi

- Gieo dữ liệu cho bảng theloai

php artisan make:seeder CategoryTableSeeder

- Gieo dữ liệu cho bảng tacgia

php artisan make:seeder AuthorTableSeeder

- Gieo dữ liệu cho bảng baiviet

php artisan make:seeder ArticleTableSeeder

