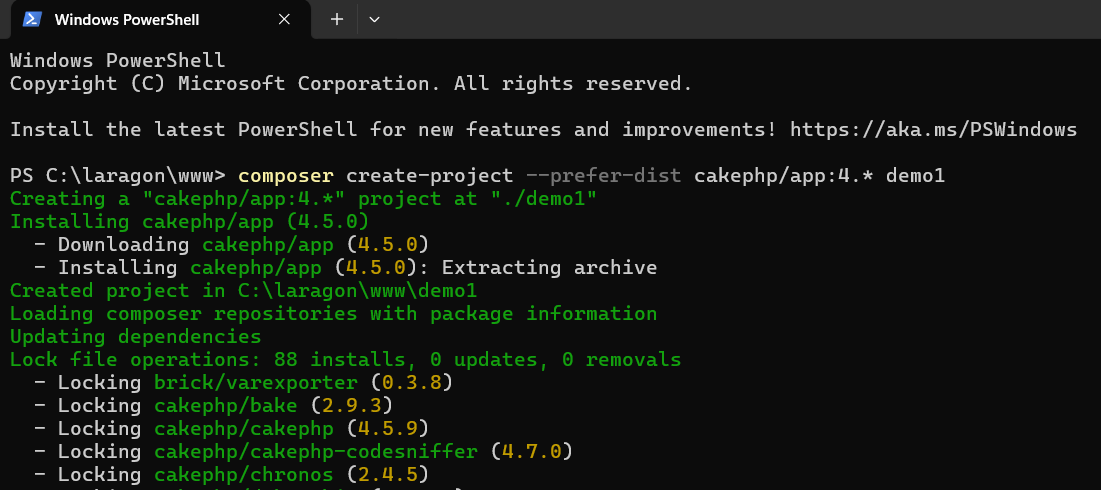
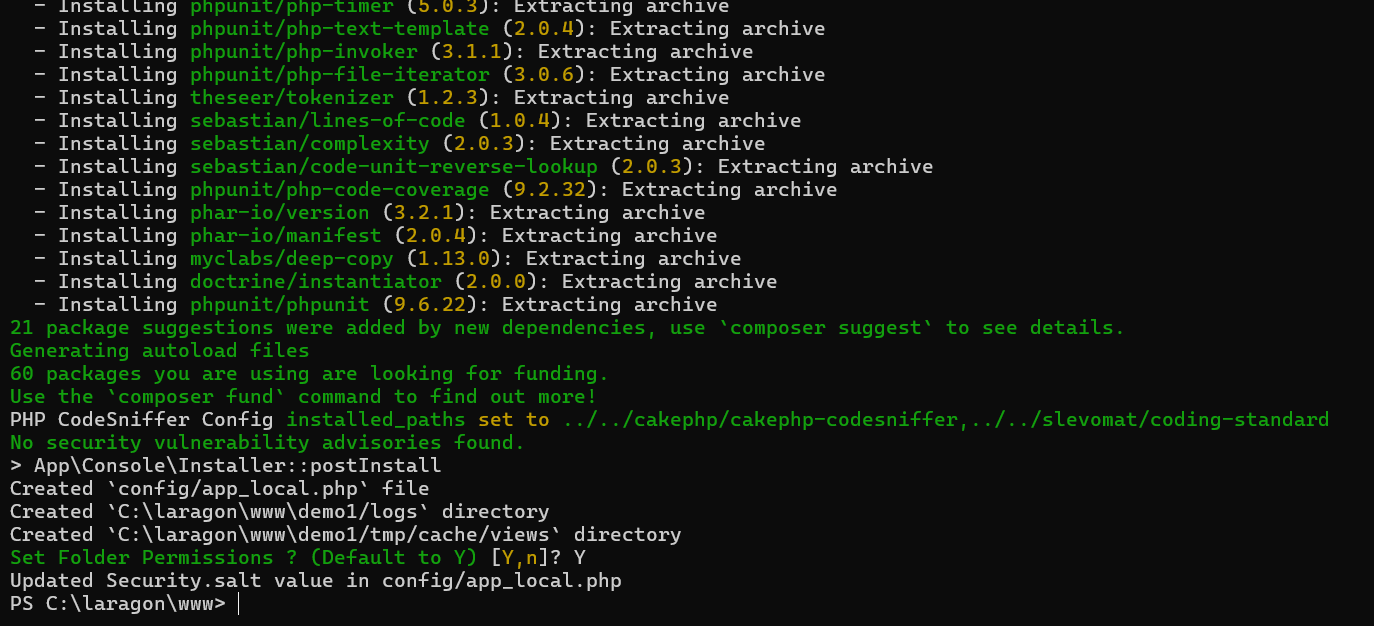
1. Bạn hãy chạy tất cả các lệnh hướng dẫn ở trên và chụp lại màn hình kết quả.

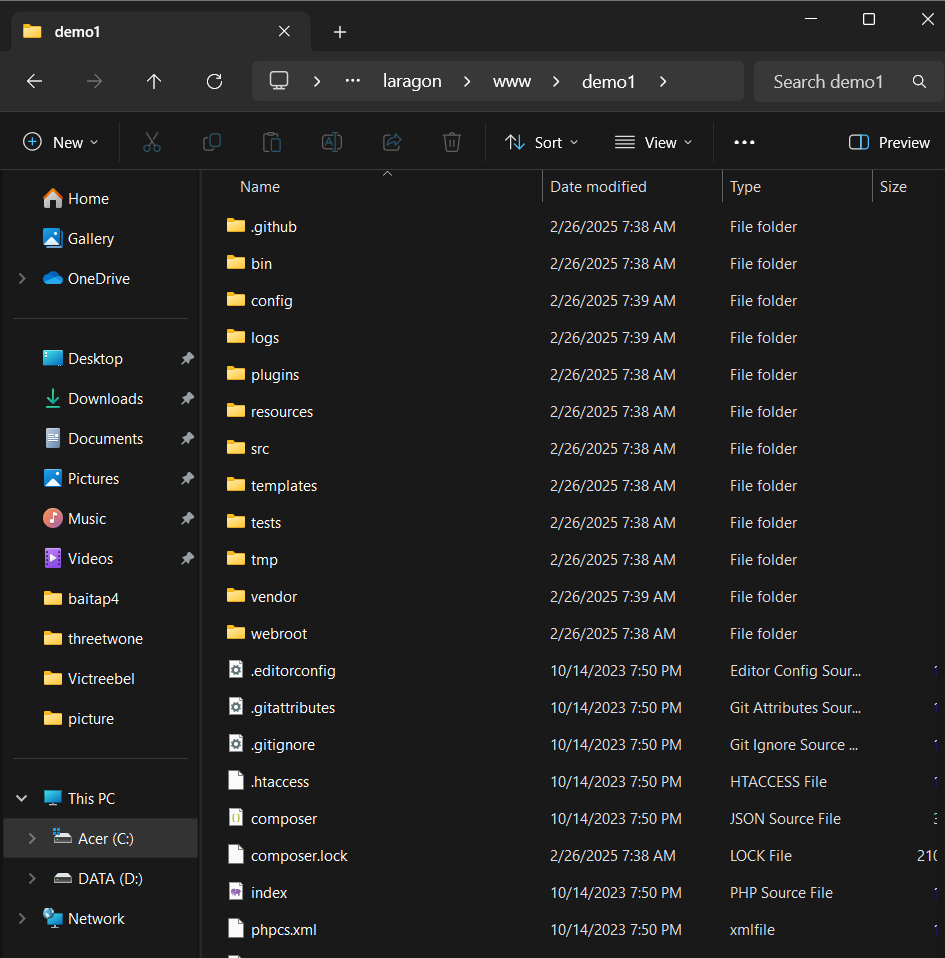
Tạo project với CakePHP:

composer create-project --prefer-dist cakephp/app:4.\* demo1





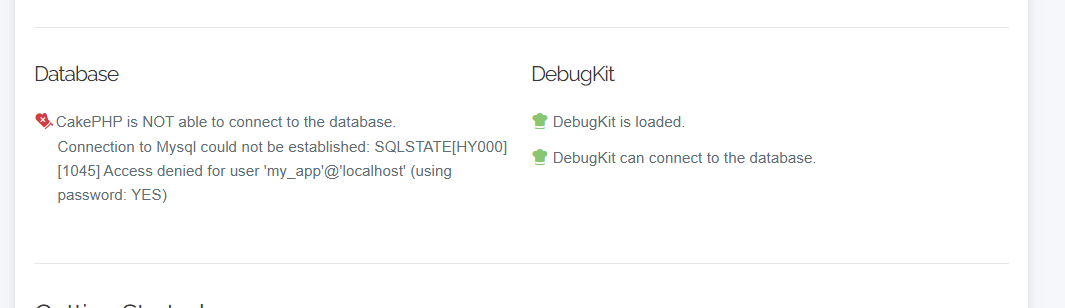
-Thư mục:



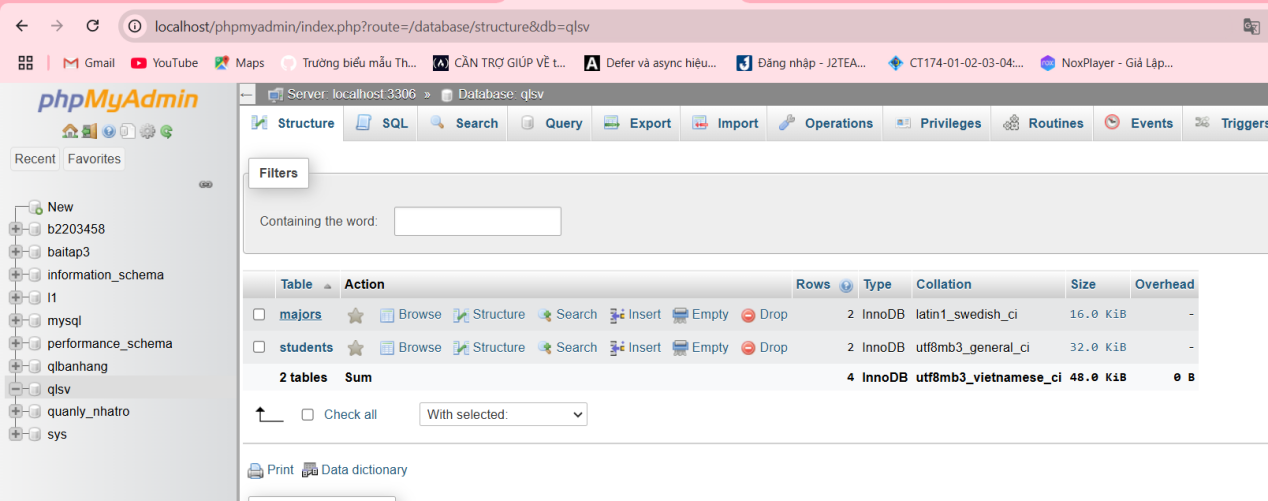
-Chạy trên localhost:



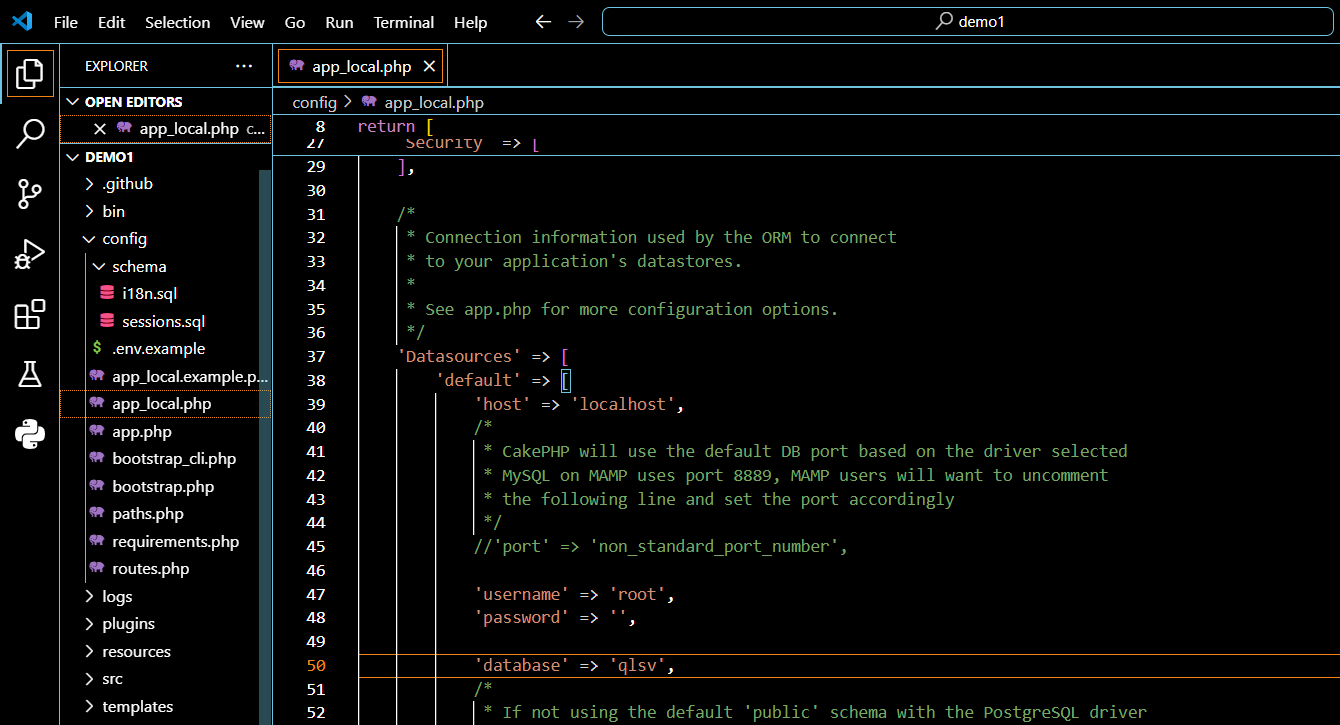
**-**Chưa thiết lập kết nối database nên có lỗi:



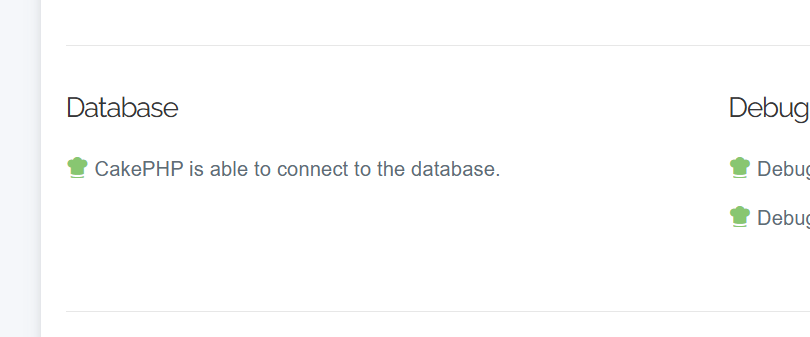
\_Tạo cơ sở dữ liệu để kết nối đến CakePHP vừa tạo:



\_Sửa file app\_local.php để kết nối cơ sở dữ liệu:

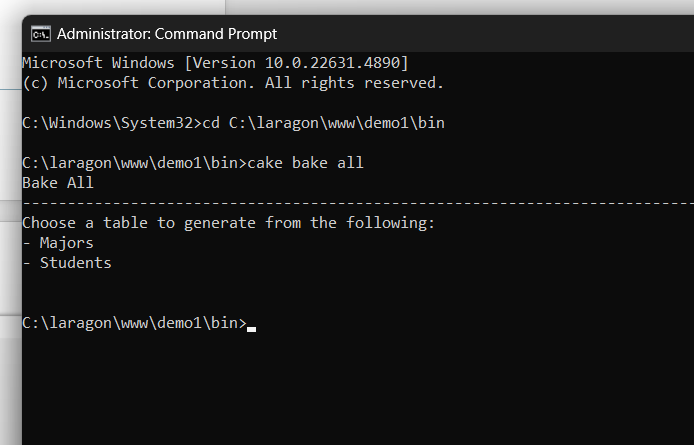


-Kết quả:



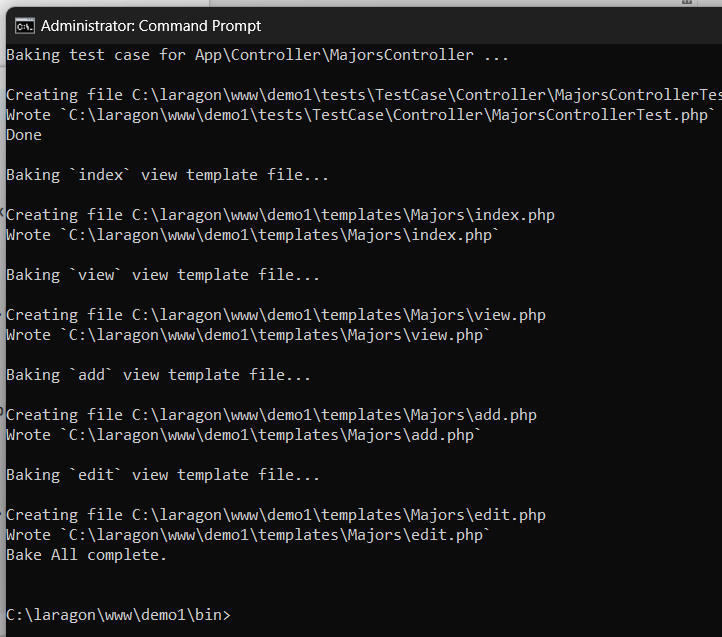
 Chúng ta trở lại giao diện dòng lệnh cmd ban nãy và gõ:

**cd C:\laragon\www\demo1\bin**

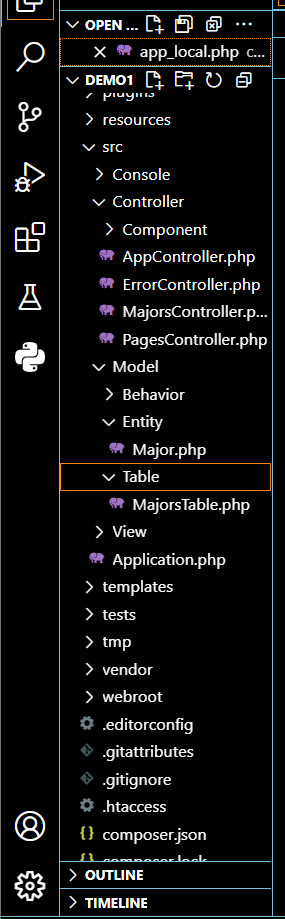


Tạo CRUD cho bảng majors (là bảng cha), ta thực hiện lệnh

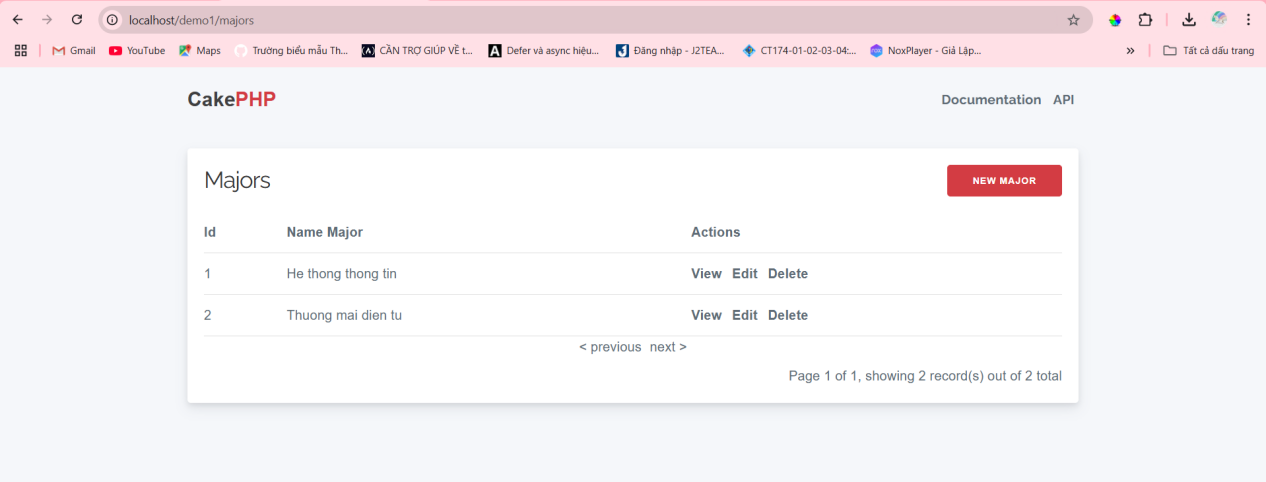
**cake bake all majors**



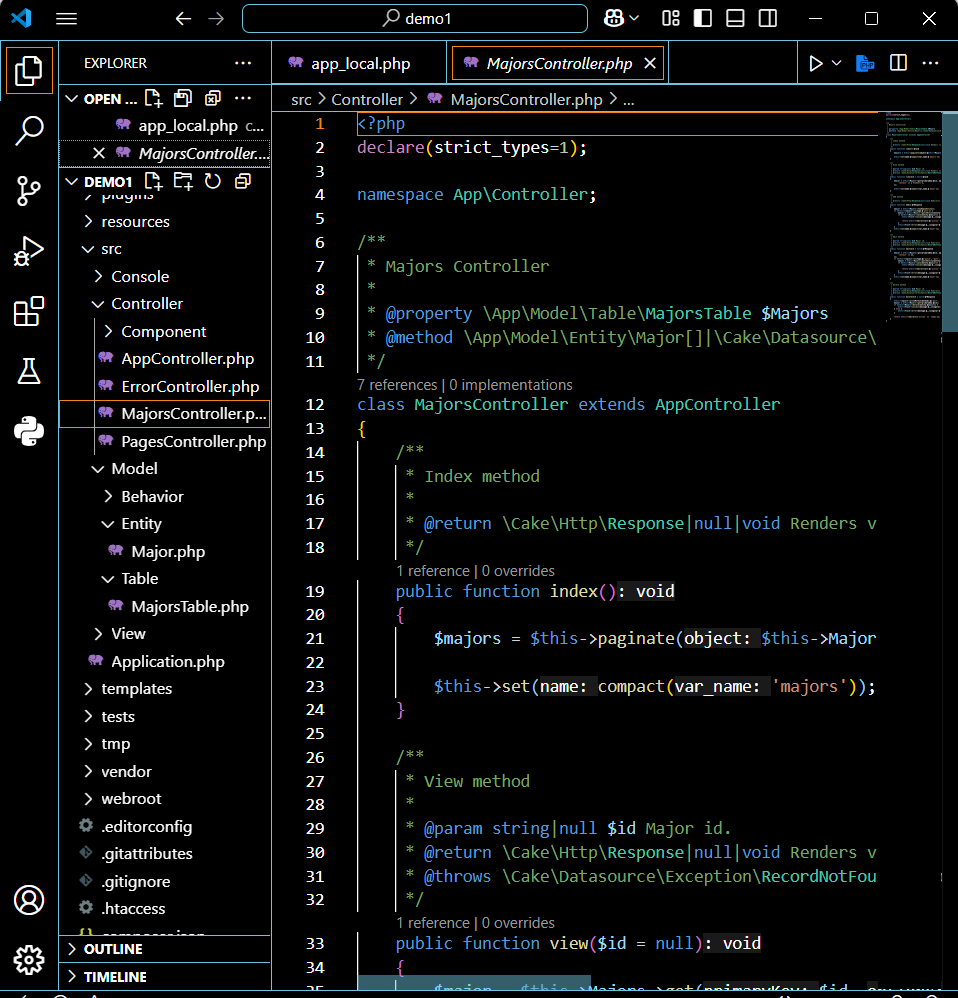
Các file được sinh ra



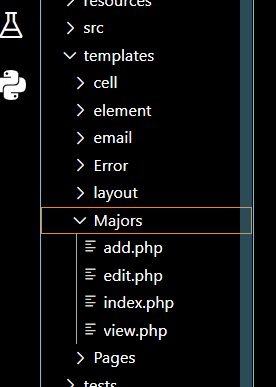
*[http://localhost/demo1/majors](http://localhost/demo1/majors )*



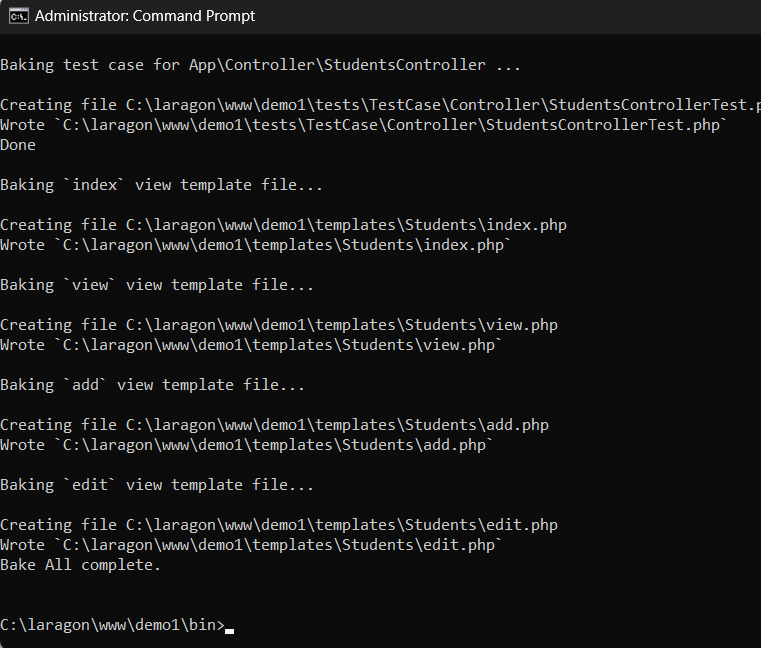
MajorsController.php



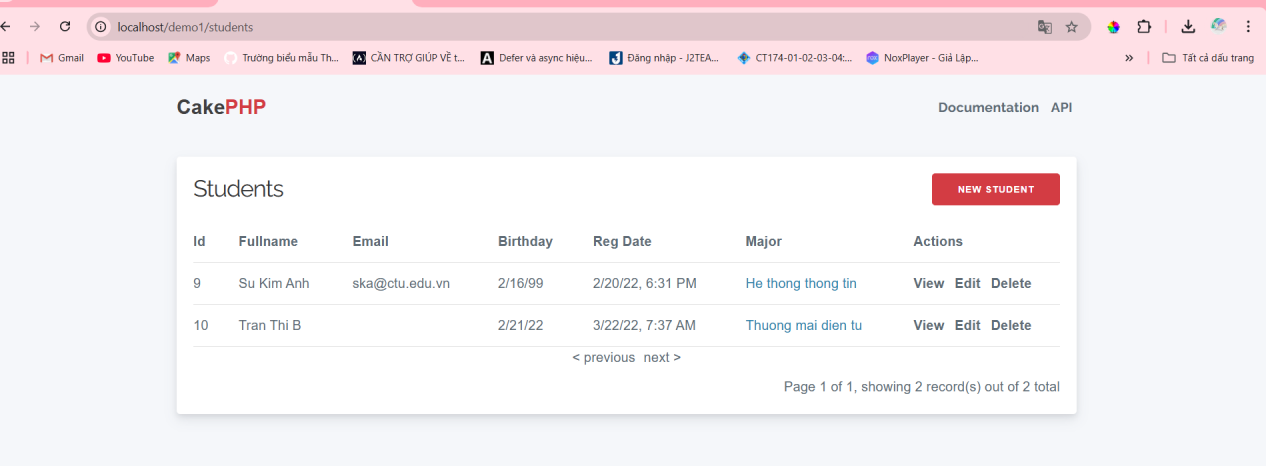
 Template/Majors/



Lặp lại tương tự đối với bảng students: **cake bake all students**

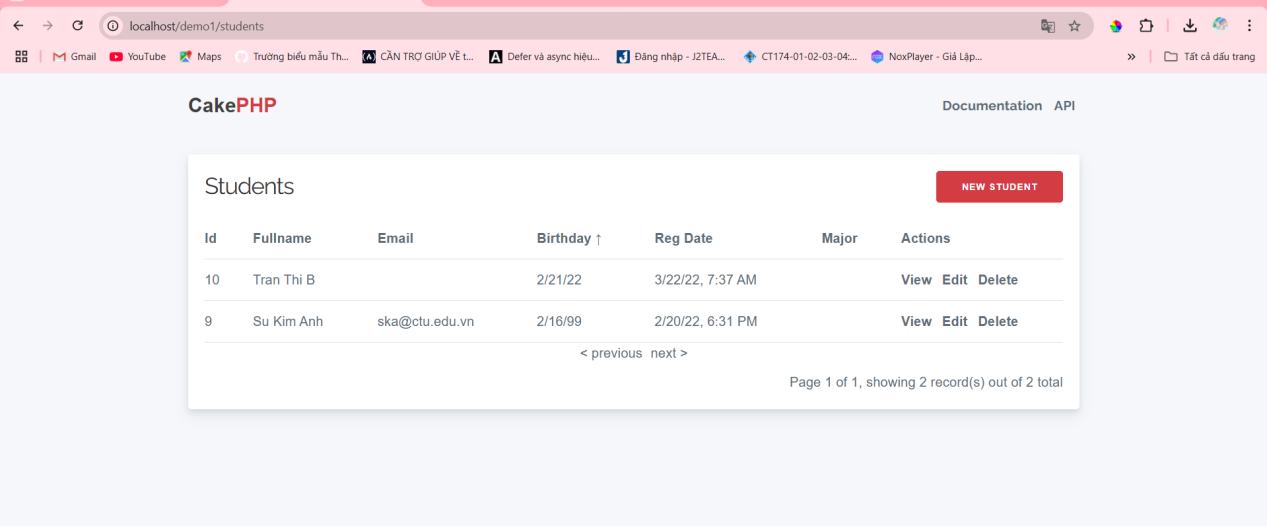


<http://localhost/demo1/students>



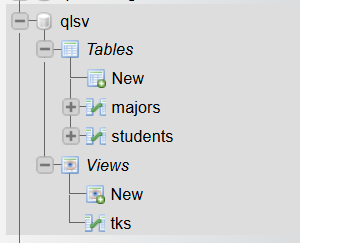
Ta muốn sắp xếp dữ liệu với ngày sinh lớn hơn sẽ hiển thị trước chúng ta sẽ vào Controller của Students .Chỉnh sửa nội dung phương thức index() tùy chỉnh sắp xếp giảm dần theo ngày sinh



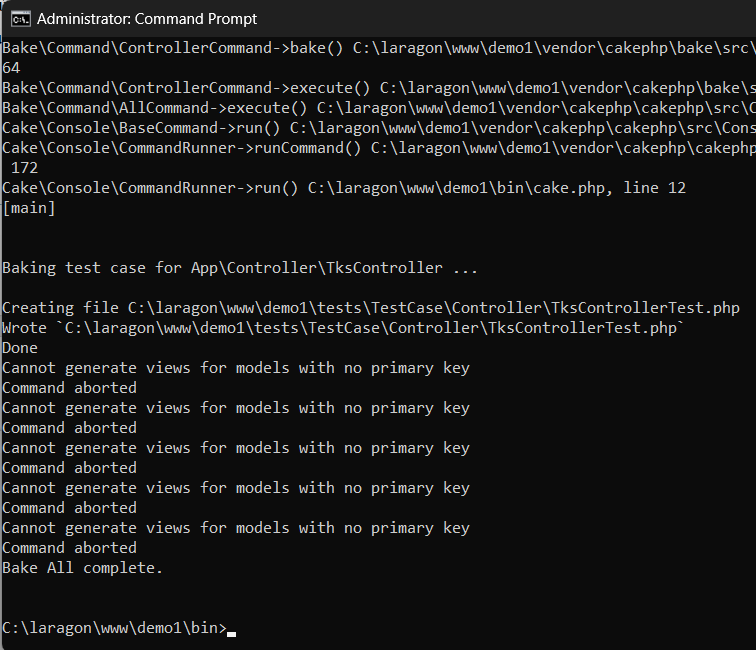


Hướng dẫn tạo CRUD từ view CSDL với CakePHP:

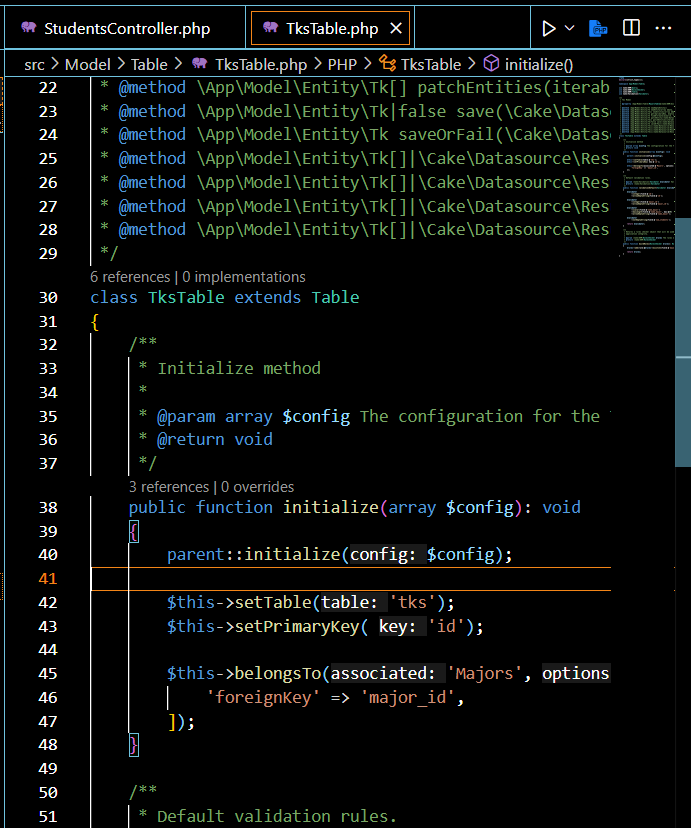
tạo view

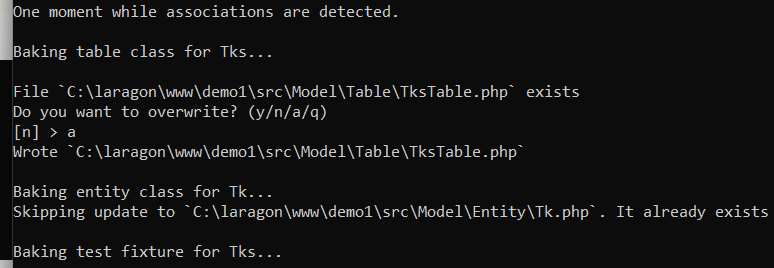


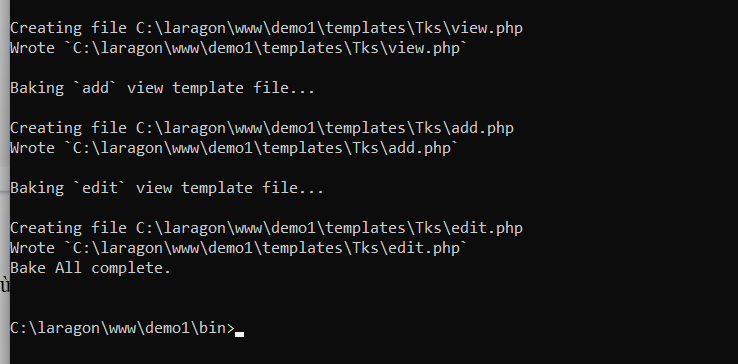
Cake bake all tks

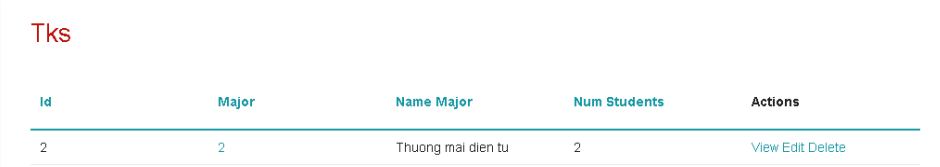


Trong phương thức initialize Sau lệnh *$this->setTable(“tks”);* Các bạn thêm lệnh sau: *$this->setPrimaryKey(‘id’);* 15 Lệnh này để Cakephp nhận biết id là khóa trong tks để có thể tiến hành tự động CRUD khi ta thực hiện lệnh *cake bake*



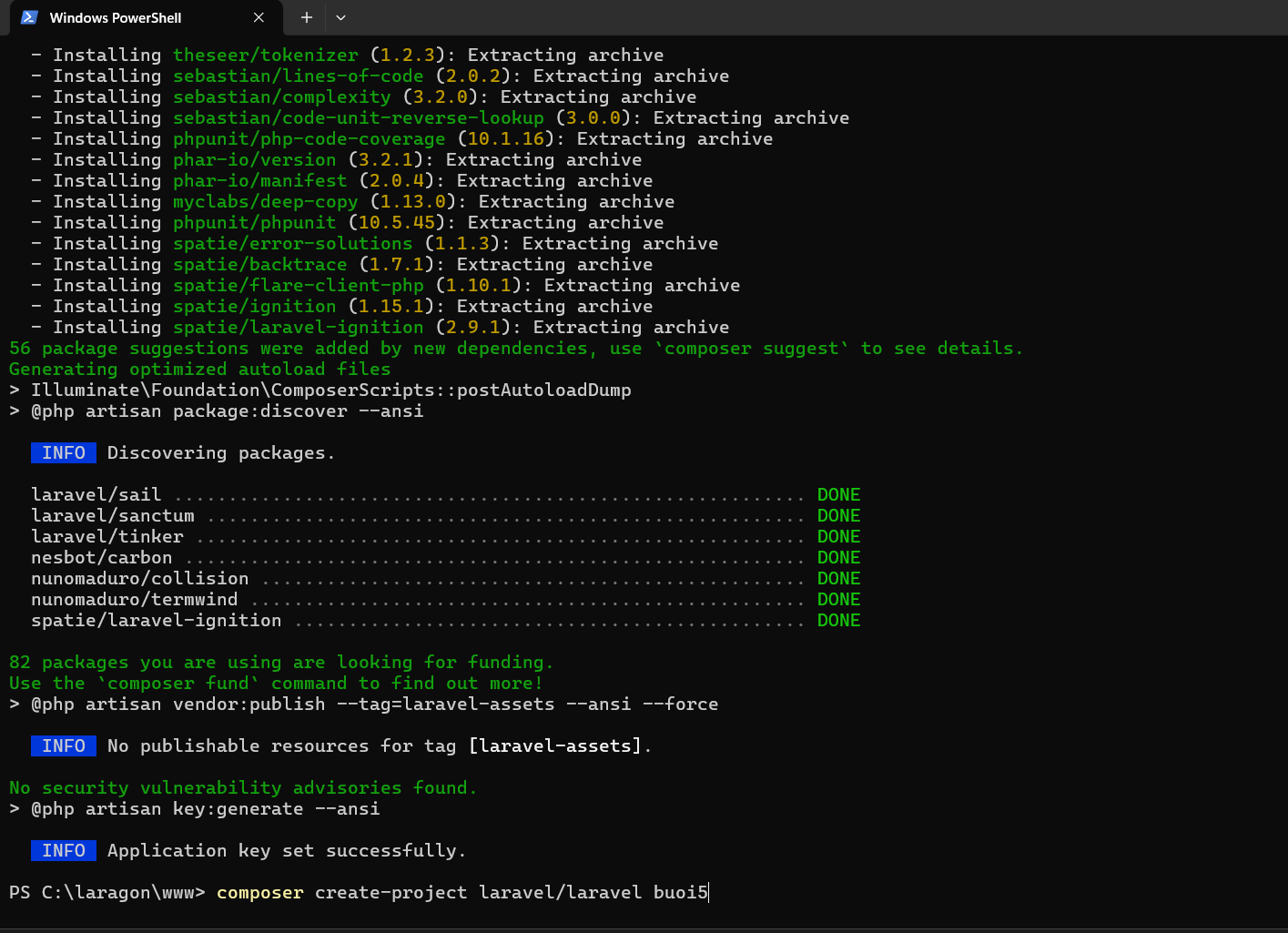




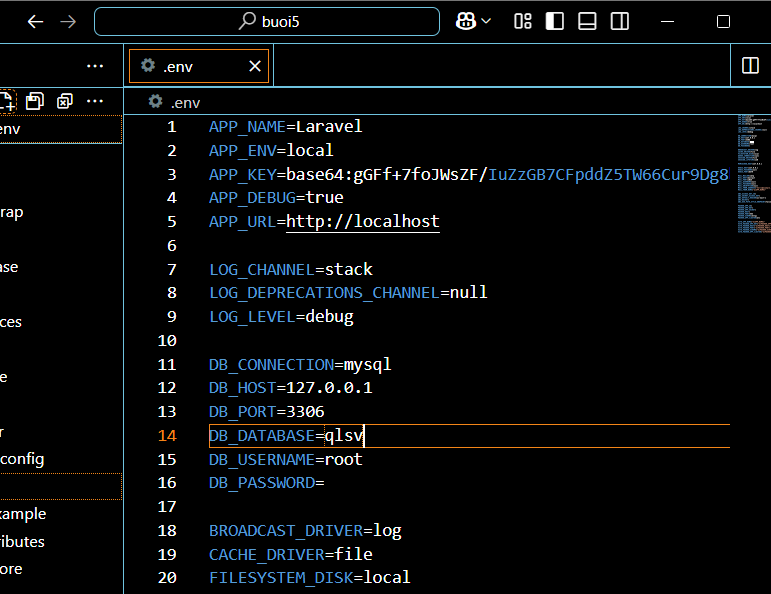


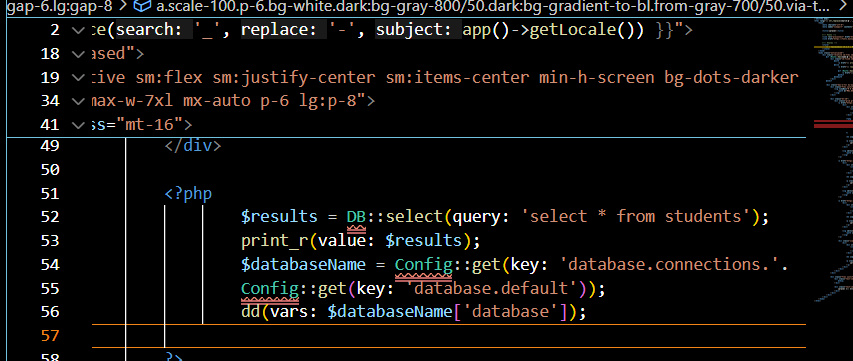
Tạo Project Laravel kết nối cơ sở dữ liệu :

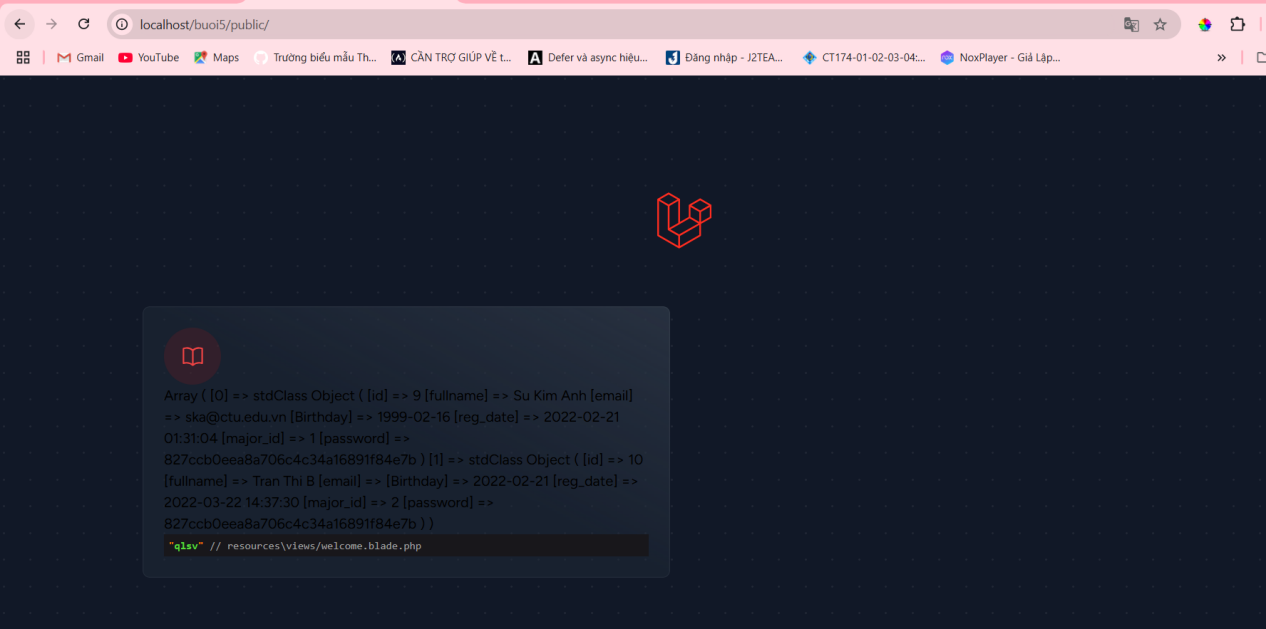
*composer create-project laravel/laravel buoi5*



Đặt tên csdl trong .env

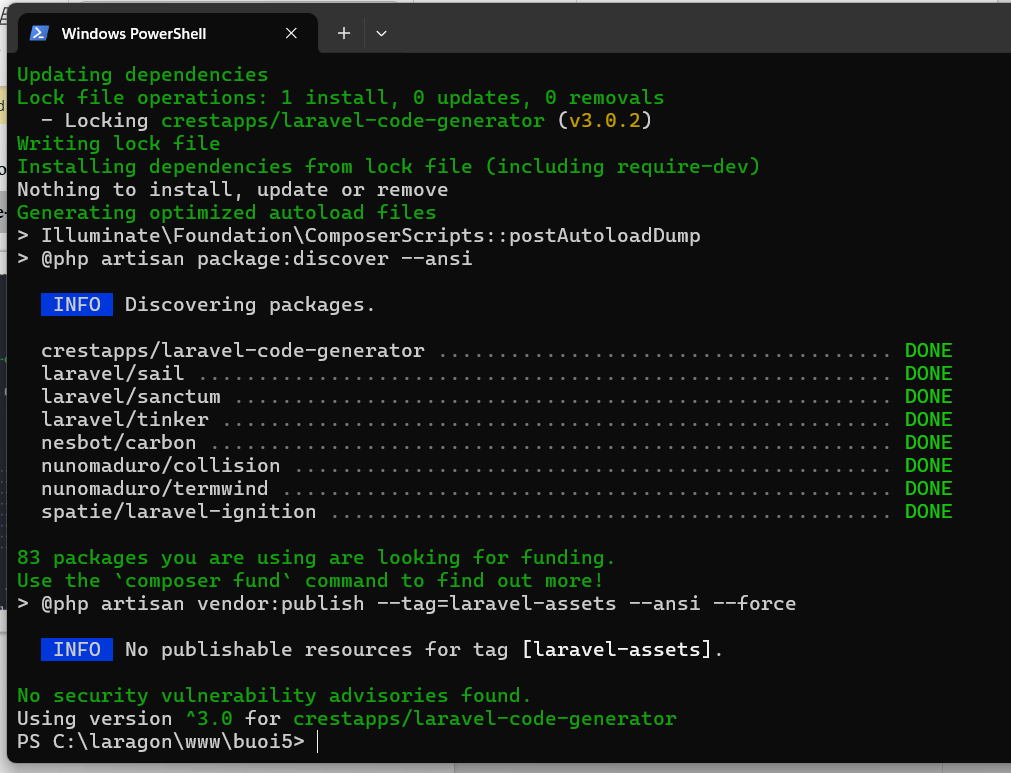






Cài đặt CrestApps/laravel-code-generator:

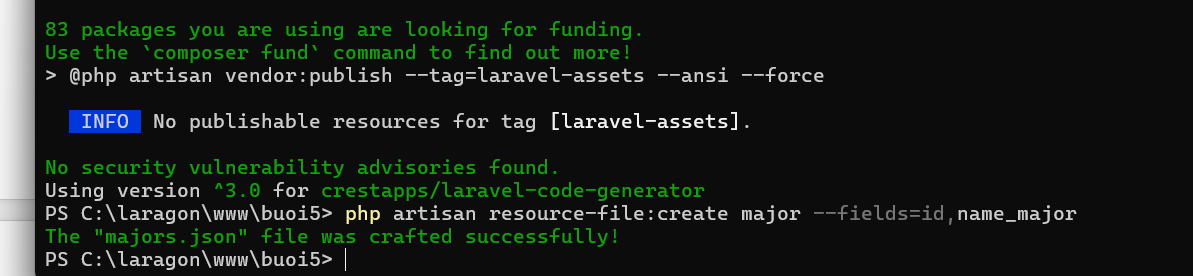
composer require crestapps/laravel-code-generator --dev



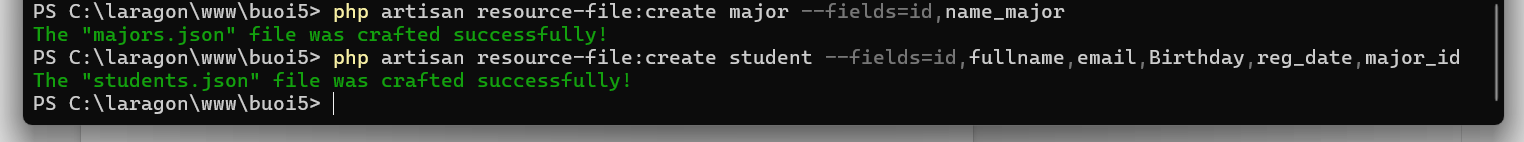
Tạo các script mô tả để sinh ra code:

Go lenh sau:

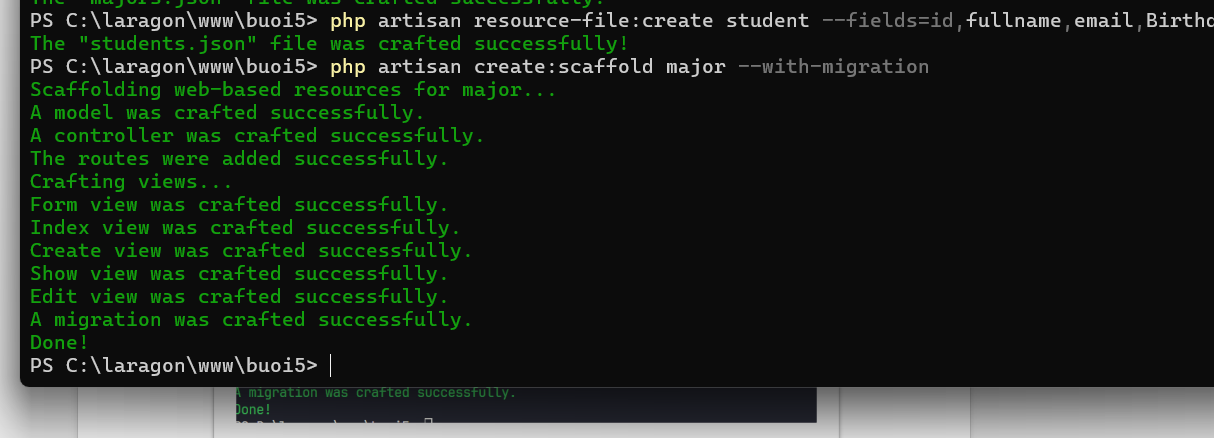
*php artisan resource-file:create major --fields=id,name\_major*



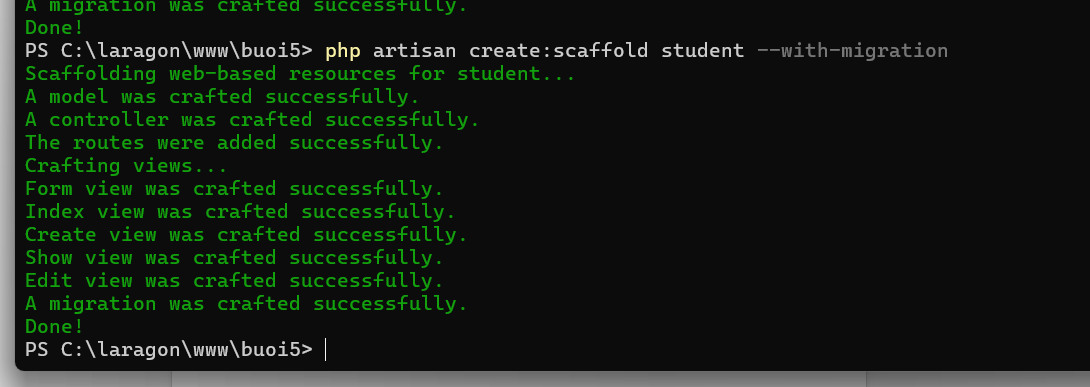
*php artisan resource-file:create student --fields=id,fullname,email,Birthday,reg\_date,major\_id*



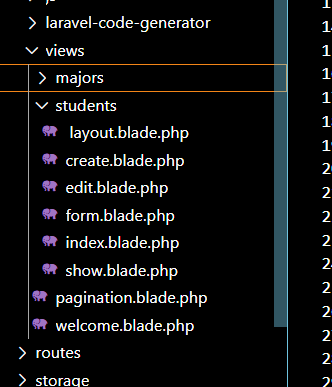
*php artisan create:scaffold major --with-migration*

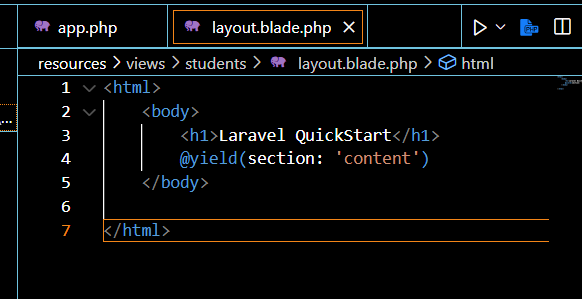


*php artisan create:scaffold student --with-migration*

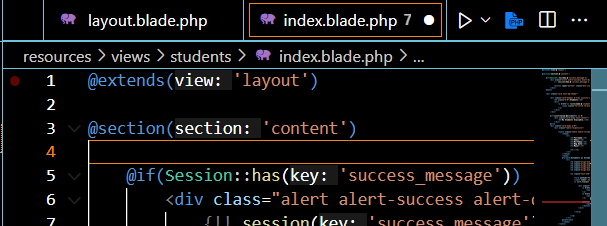


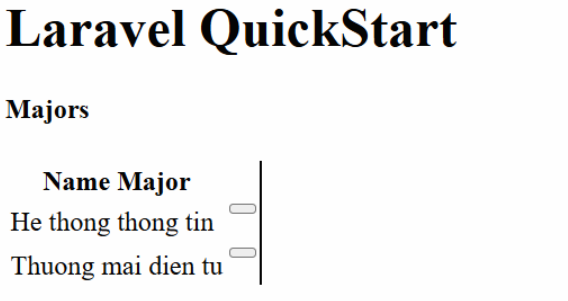
layout.blade.php





buoi5/resources/views/students/ mở tập tin index.blade.php Sửa dòng @extends('layouts.app') thành @extends('pages.layouts.app') hoặc @extends('layout') như hình:





1. Đọc và tìm hiểu về Cakephp tại đây: , bạn hãy cho biết Cakephp là gì? Hãy mô tả các quy tắc trong các Layer Model, View, Controller của Cakephp. Tìm hiểu thêm trên Internet, các bạn trình bày Cakephp ra đời năm nào? Tác giả, lịch sử phát triển các phiên bản.

\_CakePHP là một framework mã nguồn mở dành cho PHP, được thiết kế dựa trên mô hình **MVC (Model-View-Controller)**. Framework này giúp lập trình viên phát triển ứng dụng web nhanh chóng nhờ vào cấu trúc rõ ràng và các quy tắc có sẵn, giúp giảm thiểu việc viết lại mã. CakePHP hỗ trợ **CRUD (Create, Read, Update, Delete)** tự động, tuân thủ lập trình hướng đối tượng (OOP) và cung cấp các tính năng bảo mật mạnh mẽ cũng như quản lý cơ sở dữ liệu hiệu quả.

### \_Cấu trúc MVC trong CakePHP

#### ****\*Model****

* Quản lý dữ liệu và kết nối với cơ sở dữ liệu.
* Đại diện cho các bảng trong database, thường được đặt tên theo dạng số nhiều.
* Hỗ trợ các quy tắc như **liên kết bảng (associations), xác thực dữ liệu (validation), và các sự kiện callback** trong quá trình lưu trữ dữ liệu.

#### ****\*View****

* Chịu trách nhiệm hiển thị dữ liệu ra giao diện.
* CakePHP sử dụng **templating engine**, giúp tổ chức HTML/PHP theo từng hành động của controller.
* View không làm việc trực tiếp với database mà nhận dữ liệu từ controller, sử dụng các **helper** như FormHelper, HtmlHelper để hỗ trợ tạo giao diện.

#### ****\*Controller****

* Xử lý yêu cầu từ người dùng, tương tác với model để lấy dữ liệu và gửi đến view.
* Thường có tên theo dạng số ít, đại diện cho một model cụ thể.
* Chứa các phương thức **index, add, edit, delete**, tương ứng với các hành động CRUD, và mỗi phương thức này liên kết với một file view tương ứng.

### \_Lịch sử phát triển CakePHP

#### ****\*Sự ra đời****

CakePHP xuất hiện vào tháng 4/2005, được sáng tạo bởi **Michal Tatarynowicz**, một lập trình viên người Ba Lan. Sau khi giới thiệu phiên bản đầu tiên, framework này tiếp tục được phát triển bởi **Larry E. Masters** cùng cộng đồng lập trình viên, dẫn đến sự thành lập của **Cake Software Foundation**.

#### ****\*Các phiên bản quan trọng****

* **CakePHP 1.0 (2006):** Phiên bản đầu tiên chính thức ra mắt.
* **CakePHP 2.x (2011):** Cải thiện hiệu suất, bảo mật và tương thích tốt hơn với các phiên bản PHP mới.
* **CakePHP 3.x (2015):** Thay đổi lớn về cấu trúc code, hiệu suất và bổ sung ORM mạnh mẽ hơn.
* **CakePHP 4.x (2019):** Cập nhật hiện đại hóa, nâng cao hiệu suất và khả năng mở rộng.

1. Dựa vào hãy cho biết các quy tắc đặt tên các bảng trong CSDL để Cakephp có thể nhận dạng được hỗ trợ cho việc sinh CRUD tự động.

CakePHP có các quy tắc đặt tên nhất quán giúp framework tự động nhận diện bảng dữ liệu và liên kết với các model tương ứng. Tuân thủ các quy tắc này giúp việc tạo CRUD tự động trở nên dễ dàng hơn.

#### ****1. Đặt tên bảng ở dạng số nhiều****

Tên bảng trong cơ sở dữ liệu nên sử dụng dạng số nhiều để CakePHP có thể tự động ánh xạ đến model tương ứng. Ví dụ, bảng users sẽ được framework liên kết với model User.

#### ****2. Sử dụng chữ thường và dấu gạch dưới****

Tên bảng nên được viết bằng chữ thường và các từ được ngăn cách bằng dấu gạch dưới \_. Ví dụ, bảng order\_items sẽ tương ứng với model OrderItem.

#### ****3. Primary key mặc định là**** id

Mỗi bảng cần có một cột id đóng vai trò là khóa chính. CakePHP mặc định nhận diện id là primary key của bảng dữ liệu.

#### ****4. Định danh bảng liên kết (Join Table) trong quan hệ nhiều-nhiều****

Với các bảng trung gian trong quan hệ **many-to-many**, CakePHP yêu cầu đặt tên theo dạng số nhiều của hai bảng liên kết, sắp xếp theo thứ tự bảng chữ cái và dùng dấu gạch dưới để nối. Ví dụ, bảng trung gian giữa products và categories sẽ có tên là categories\_products.

#### ****5. Quy tắc đặt tên khóa ngoại (Foreign Key)****

Các cột khóa ngoại trong bảng con được đặt theo quy tắc: **tên bảng cha ở dạng số ít +** \_id. Ví dụ, bảng comments có cột post\_id để tham chiếu đến bảng posts.

1. Hãy tìm và trình bày các số liệu mới nhất về người sử dụng Cakephp và Laravel mà bạn tìm thấy được trên Internet.

#### ****CakePHP****

* Hiện có khoảng **44,059** trang web đang chạy trên CakePHP.
* Tổng số trang web từng sử dụng CakePHP trước đây là **132,432**.
* Tại Hoa Kỳ, **15,654** trang web đang sử dụng framework này.

#### ****Laravel****

* Hiện tại, **1,154,113** trang web đang hoạt động trên nền tảng Laravel.
* Tổng số trang web từng triển khai Laravel nhưng không còn sử dụng là **1,294,871**.
* Riêng tại Hoa Kỳ, có **286,524** trang web đang sử dụng Laravel.

1. Hãy phân tích điểm mạnh và điểm yếu của Cakephp và Laravel, tiến hành so sánh 2 framework này.

### ****1. CakePHP****

#### ****Ưu điểm:****

* **Quy ước thay vì cấu hình:** CakePHP áp dụng triết lý “Convention over Configuration,” giúp lập trình viên tiết kiệm thời gian bằng cách sử dụng các quy tắc có sẵn thay vì phải thiết lập thủ công.
* **Bảo mật tích hợp:** Framework này có sẵn các tính năng bảo vệ như chống SQL injection, XSS, CSRF…, giúp nâng cao tính an toàn cho ứng dụng.
* **Hỗ trợ CRUD tự động:** CakePHP cung cấp các phương thức để dễ dàng thao tác với dữ liệu (tạo, đọc, cập nhật, xóa) mà không cần viết nhiều mã.
* **Duy trì và bảo trì lâu dài:** Framework này có tính ổn định cao, phù hợp cho các dự án có vòng đời dài và cần bảo trì định kỳ.
* **Hỗ trợ tốt đa ngôn ngữ:** CakePHP giúp phát triển ứng dụng với khả năng nội địa hóa (localization) dễ dàng, hỗ trợ nhiều ngôn ngữ.

#### ****Nhược điểm:****

* **Ít phổ biến hơn so với Laravel:** Vì cộng đồng nhỏ hơn, tài liệu và hỗ trợ từ bên thứ ba ít hơn, đôi khi gây khó khăn khi gặp vấn đề phức tạp.
* **Hạn chế trong việc tùy chỉnh:** CakePHP tuân thủ chặt chẽ các quy ước, điều này có thể gây khó khăn nếu dự án cần tùy biến linh hoạt hơn.
* **Tính linh hoạt thấp hơn:** Do tập trung vào quy ước, framework này không có nhiều lựa chọn mở rộng hoặc thay đổi cấu trúc như các framework khác.

### ****2. Laravel****

#### ****Ưu điểm:****

* **Cộng đồng lớn và tài liệu phong phú:** Laravel có một hệ sinh thái phát triển rộng rãi, tài liệu hướng dẫn chi tiết và dễ tiếp cận.
* **Hệ sinh thái mở rộng đa dạng:** Laravel hỗ trợ nhiều công cụ hữu ích như Laravel Echo (real-time), Laravel Passport (OAuth2), Laravel Sanctum (API Token), Laravel Horizon (quản lý job queue).
* **Cú pháp hiện đại, dễ sử dụng:** Laravel sử dụng cú pháp rõ ràng, trực quan, kết hợp với Eloquent ORM và Blade Template Engine giúp lập trình viên làm việc hiệu quả hơn.
* **Công cụ dòng lệnh mạnh mẽ:** Artisan CLI giúp dễ dàng tạo các thành phần như controller, model, migration… mà không cần thao tác thủ công.
* **Hỗ trợ tốt cho API và microservices:** Laravel có kiến trúc module hóa, phù hợp với các ứng dụng dạng API hoặc microservices.
* **Tích hợp với công nghệ front-end hiện đại:** Laravel có thể kết hợp tốt với Vue.js, React, Tailwind CSS, giúp phát triển ứng dụng full-stack dễ dàng hơn.

#### ****Nhược điểm:****

* **Hiệu suất có thể thấp hơn:** Vì Laravel cung cấp nhiều tính năng, đôi khi có thể gây ảnh hưởng đến hiệu suất nếu không được tối ưu tốt. Tuy nhiên, có thể sử dụng cache (Redis, Memcached) để cải thiện tốc độ.
* **Cấu trúc phức tạp hơn:** So với CakePHP, Laravel yêu cầu lập trình viên hiểu rõ về các thành phần của framework để sử dụng hiệu quả.
* **Cần cấu hình nhiều hơn:** Khi bắt đầu một dự án, Laravel yêu cầu thiết lập nhiều thông số hơn so với CakePHP, đặc biệt nếu sử dụng nhiều thành phần trong hệ sinh thái của nó.

1. Bạn hãy tạo 1 CSDL đơn giản để lưu trữ dữ liệu với các mô tả bên dưới:

Môn học (mã môn, tên môn, số tín chỉ, có tính điểm tích lũy)

Sinh viên (mã sinh viên, họ tên sinh viên, ngày sinh, email, giới tính, số điện thoại, mật khẩu)

Lớp môn học (Mã lớp môn học, mã môn, học kỳ niên khóa)

Điểm (mã lớp môn học, mã sinh viên, điểm)

Tên bảng, tên cột, kiểu dữ liệu các bạn tự đặt sao cho thỏa các quy tắc để các framework nhận dạng được, vd dùng quy tắc với Cakephp:[[1]](#footnote-0), có thể thêm các cột khóa giả (ID) nếu thấy thực sự cần thiết cho việc lập trình và tự động sinh mã. Tạo 1 số dữ liệu mẫu với dữ liệu là thông tin điểm số các môn học mà bạn đã học.

-- Bảng môn học (courses)

CREATE TABLE courses (

id VARCHAR(10) PRIMARY KEY,

name VARCHAR(100) NOT NULL,

credit INT NOT NULL,

is\_cumulative BOOLEAN NOT NULL DEFAULT TRUE

);

-- Bảng sinh viên (students)

CREATE TABLE students (

id VARCHAR(10) PRIMARY KEY,

full\_name VARCHAR(100) NOT NULL,

birth\_date DATE NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

gender ENUM('Nam', 'Nữ', 'Khác') NOT NULL,

phone VARCHAR(15) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL

);

-- Bảng lớp môn học (course\_classes)

CREATE TABLE course\_classes (

id VARCHAR(10) PRIMARY KEY,

course\_id VARCHAR(10) NOT NULL,

semester VARCHAR(50) NOT NULL,

FOREIGN KEY (course\_id) REFERENCES courses(id) ON DELETE CASCADE

);

-- Bảng điểm (grades)

CREATE TABLE grades (

course\_class\_id VARCHAR(10) NOT NULL,

student\_id VARCHAR(10) NOT NULL,

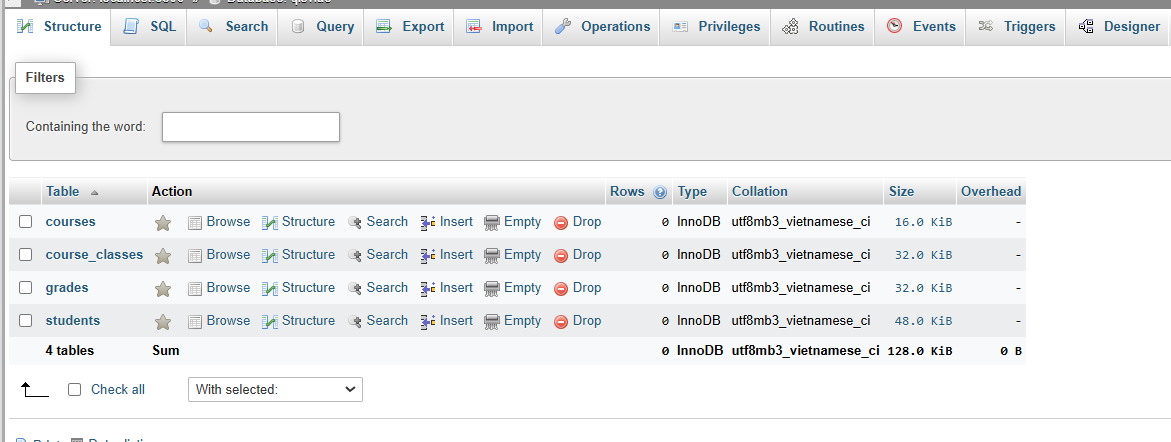
grade FLOAT CHECK (grade BETWEEN 0 AND 10),

PRIMARY KEY (course\_class\_id, student\_id),

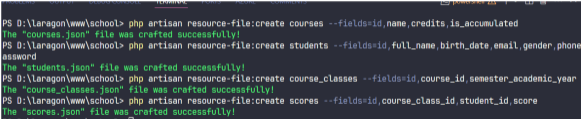
FOREIGN KEY (course\_class\_id) REFERENCES course\_classes(id) ON DELETE CASCADE,

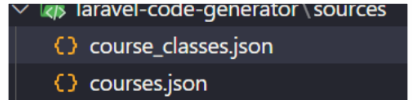
FOREIGN KEY (student\_id) REFERENCES students(id) ON DELETE CASCADE

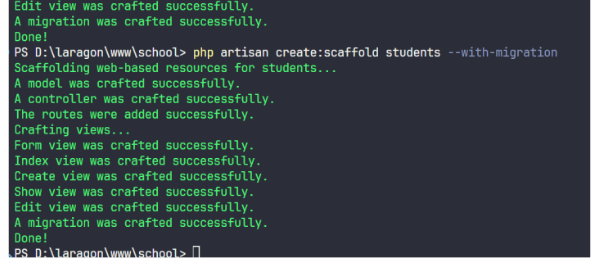
);

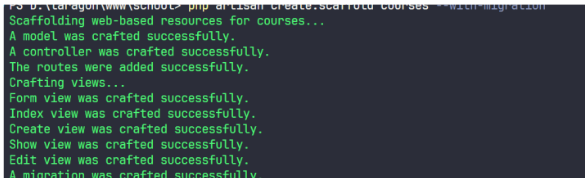


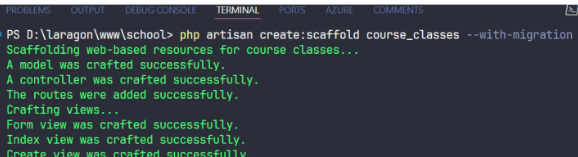
7. Bạn lựa chọn hoặc dùng Cakephp hoặc dùng Laravel để tạo project và tự động sinh CRUD cho CSDL đã tạo ở câu 6.

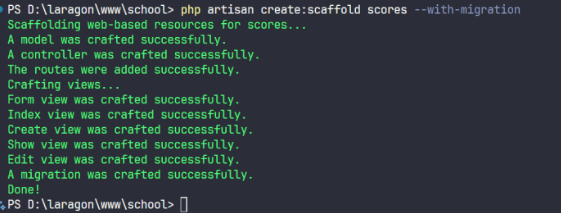






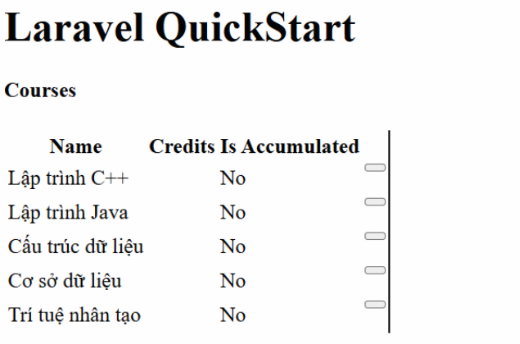




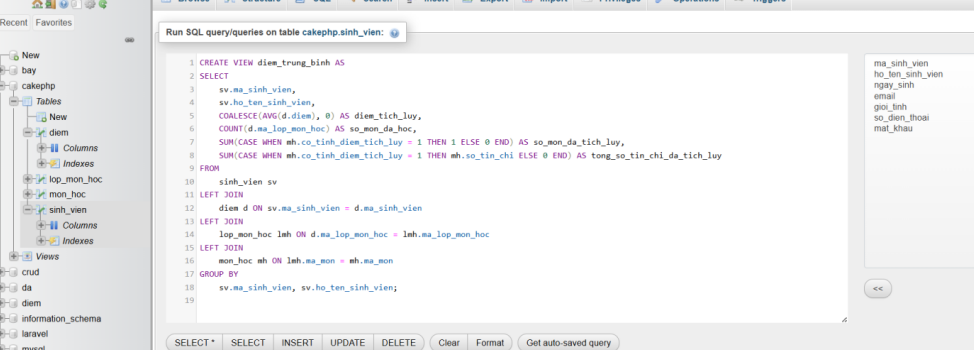


8. Từ các giao diện đã tạo bạn sẽ thêm các bạn sinh viên cùng nhóm làm bài tập của bạn,và nhập thêm 1 số dòng dữ liệu nữa để đảm bảo có ít nhất 10 sinh viên có điểm trong hệ thống.

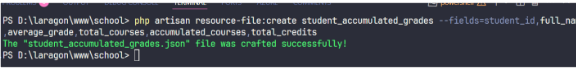


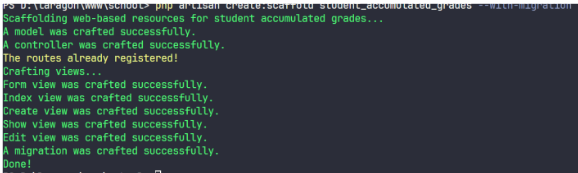


1. Tạo 1 khung nhìn (view CSDL) bằng câu lệnh SQL để tính điểm trung bình tích lũy của mỗi sinh viên, thông tin bao gồm: mã số sinh viên, họ tên, điểm tích lũy, số môn đã học, số môn đã tích lũy, tổng số tín chỉ đã tích lũy. Tên view CSDL nên đặt tên sao cho thỏa điều kiện để có thể tạo CRUD tự động.



10. Hãy tạo CRUD cho view CSDL đã tạo ở câu 9, quan sát và chụp hình kết quả. Vào controller chỉnh sửa sao cho trang index của CRUD đã được tạo hiển thị điểm tích lũy của các sinh viên sắp xếp giảm dần





1. [↑](#footnote-ref-0)