

เอกสารประกอบการเรียนปฏิบัติการครั้งที่ 8

เรื่อง การพัฒนาฐานข้อมูล โดยการสร้าง และ อ่าน ข้อมูล
ผ่านโปรแกรมประยุกต์ Laravel Framework



วัตถุประสงค์ : เมื่อจบบทปฏิบัติการนี้แล้วนักศึกษาสามารถ

- เข้าใจการทำงานการสร้างโปรเจ็คโดยใช้ Laravel Framework
- เข้าใจและสามารถใช้คำสั่งในการสร้างฐานข้อมูล
- เข้าใจและสามารถใช้คำสั่งในการสร้างตารางและวิว
- เข้าใจและสามารถใช้คำสั่งในการสร้างฟังก์ชันการเพิ่มข้อมูล
- เข้าใจและสามารถใช้คำสั่งในการสร้างฟังก์ชันการแสดงข้อมูล

ขั้นตอนการปฏิบัติ (120 นาที)

ส่วนที่ 1

การสร้างโปรเจ็คและการตั้งค่าโปรเจ็ค Laravel

ส่วนที่ 2

การสร้างฐานข้อมูล ตาราง วิว และการเพิ่มข้อมูล

ส่วนที่ 3

การสร้างหน้าแสดงข้อมูล และการตกลแต่งหน้าต่าง

ส่วนที่ 4

การสร้างฟังก์ชันการเพิ่มข้อมูล และการตกลแต่งหน้าต่าง

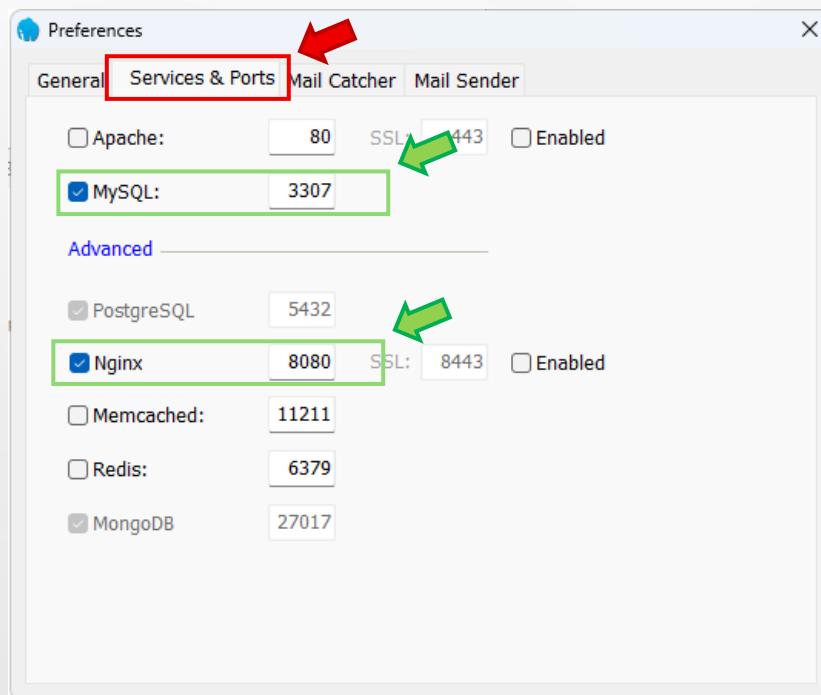
DGT01 0310 พื้นฐานการออกแบบและพัฒนาฐานข้อมูล
สำนักวิชาศาสตร์และศิลป์ดิจิทัล มหาวิทยาลัยเทคโนโลยีสุรนารี
ผู้ช่วยศาสตราจารย์ ดร.สติตย์โชค โพธิ์สะอาด
นายปิยะ รองในเมือง

ส่วนที่ 1 การสร้างโปรเจ็คและการตั้งค่าโปรเจ็ค Laravel

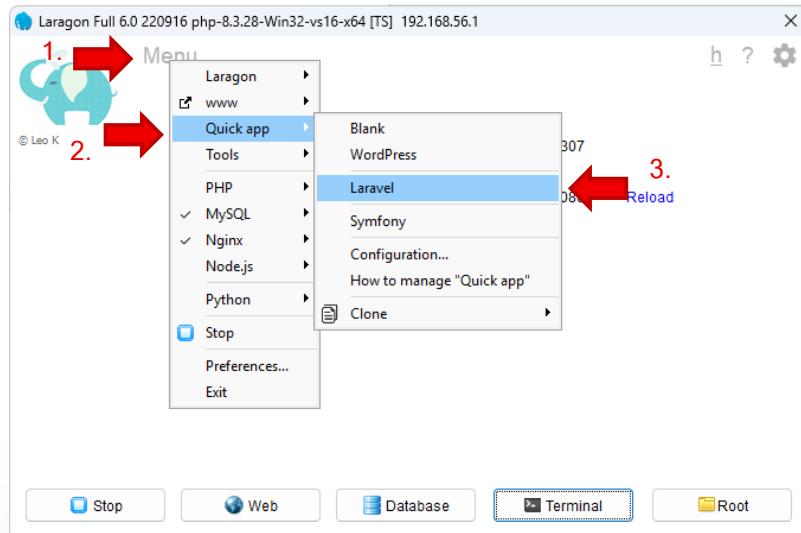
1. เปิดโปรแกรม Laragon แล้วให้ตั้งค่าตามหน้าจ่อไปนี้ เพื่อให้โปรเจ็คสามารถทำไปในทิศทางเดียวกัน กดที่รูปเพื่องาน



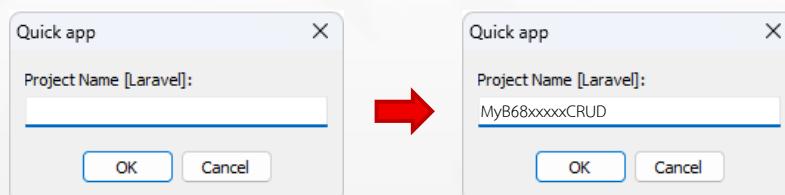
2. เลือกแท็บ Service & Ports ตามกรอบสีแดง และให้กำหนด Service และกำหนดเลขพอร์ตให้ตรงกับที่ล้อมกรอบสีเขียว เมื่อตั้งค่าเสร็จแล้วให้กดหน้าต่างปิด จากนั้นกด Start All



3. ให้เลือกเมนูตามขั้นตอนที่ปรากฏในรูปต่อไปนี้



4. จะได้หน้าต่างขึ้นมา ให้ตั้งชื่อโปรเจ็คว่า “MyB68xxxxCRUD” ***B68xxxx ให้ใส่เป็นรหัสนักศึกษาของตัวเอง แล้ว กด OK เพื่อสร้างโปรเจ็ค Laravel



5. เมื่อกด OK แล้วจะมีหน้าต่าง Terminal ขึ้นมา ให้รอรันจนกระบวนการรันเสร็จสิ้น

```

Creating project: [MyB68xxxxCRUD]. Please wait...
***** Database:
Created database: [MyB68xxxxCRUD]

***** Hint: In Terminal, you can type:
cd C:\laragon\www
composer create-project laravel/laravel MyB68xxxxCRUD --prefer-dist

Running.....
Creating a "laravel/laravel" project at "./MyB68xxxxCRUD"
Installing laravel/laravel (v12.10.1)
- Installing laravel/laravel (v12.10.1): Extracting archive
Created project in C:\laragon\www\MyB68xxxxCRUD
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
|
```



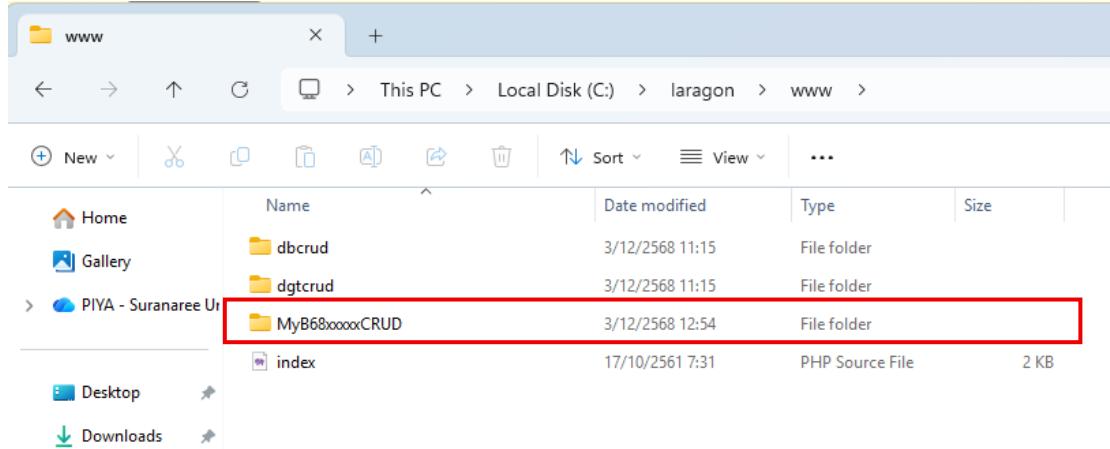
```

Use the 'composer fund' command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force
INFO No publishable resources for tag [laravel-assets].
No security vulnerability advisories found
> @php artisan key:generate --ansi
INFO Application key set successfully.

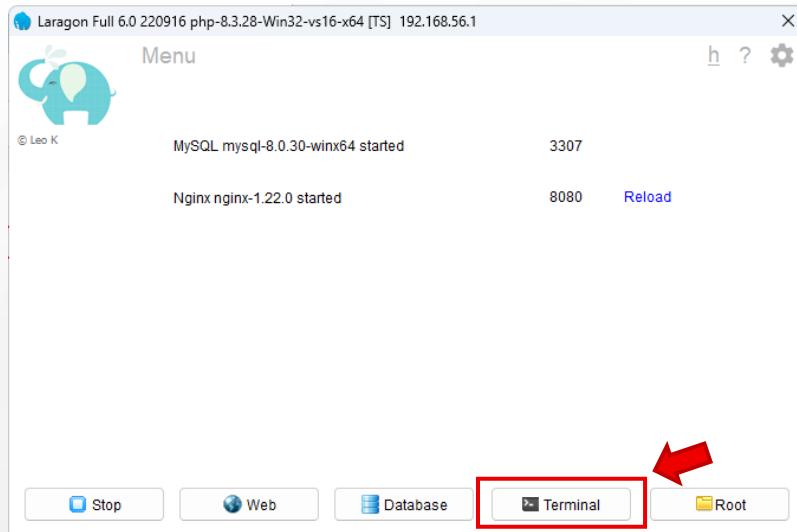
> @php -r "file_exists('database/database.sqlite') || touch('database/database.sqlite');"
> @php artisan migrate --graceful --ansi
WARN could not find driver (Connection: sqlite, SQL: select exists (
select 1 from "main".sqlite_master where name = 'migrations' and type =
'table') as "exists").

***** NOTE: Now, you can use pretty url for your awesome project : ****
*
(Laragon) Project path: C:/Laragon/www/MyB68xxxxCRUD
(Laragon) Pretty url: http://MyB68xxxxCRUD.test
```

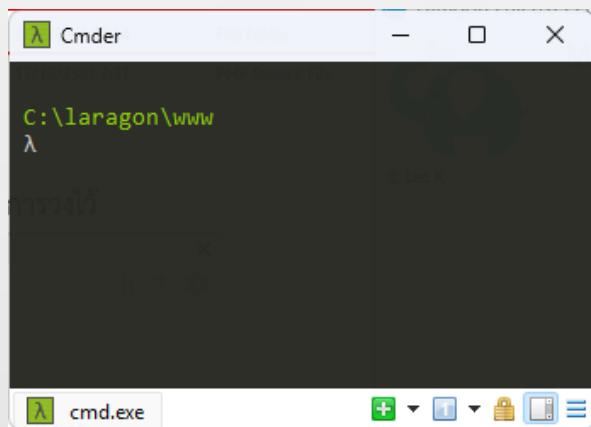
6. หากสร้างโปรเจ็คได้สำเร็จ จะมีโฟลเดอร์ “MyB68xxxxCRUD” ขึ้นมา



7. จากนั้นให้เปิด Terminal ผ่าน Laragon กดตามที่กรอบได้ทำการรันไว้



จะได้หน้าต่างนี้ขึ้นมา



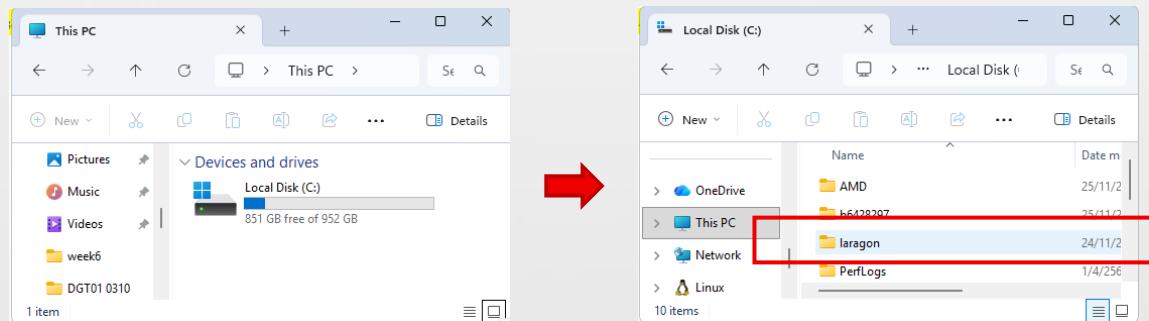
8. ให้ลองใช้คำสั่ง ls (คำสั่งเหมือน linux) แล้วจะได้รายการไฟล์ขึ้นมา จะพบโฟลเดอร์ MyB68xxxxCRUD/ หากสร้างโปรเจคได้ถูกต้องจะพบโฟลเดอร์ดังกล่าว

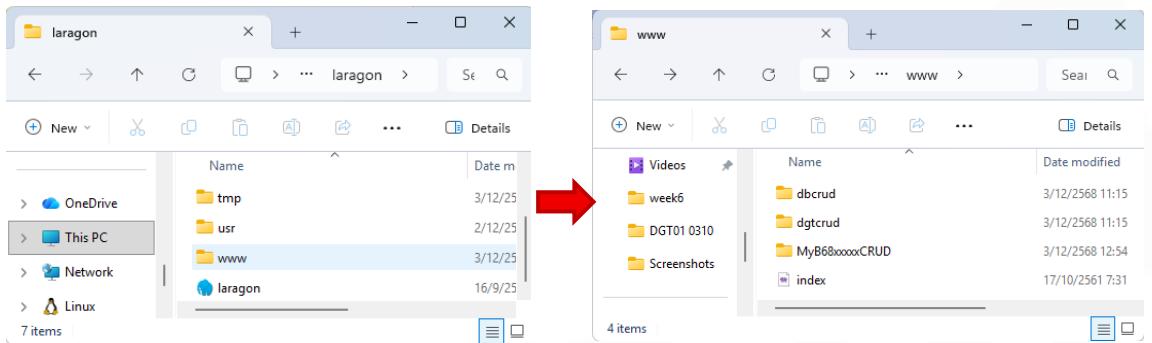
```
C:\laragon\www
λ ls
dbcrud/ index.php
dgtcrud/ MyB68xxxxCRUD/
C:\laragon\www
λ
```

9. หากพับโปรเจคแล้วแสดงโฟลเดอร์ ให้ทำการเข้าไปที่โฟลเดอร์นั้น โดยใช้คำสั่ง cd โดยพิมพ์ cd MyB68xxxxCRUD หากเข้าไปที่โฟลเดอร์ได้ Path จะเปลี่ยนเป็น C:\laragon\www\MyB68xxxxCRUD

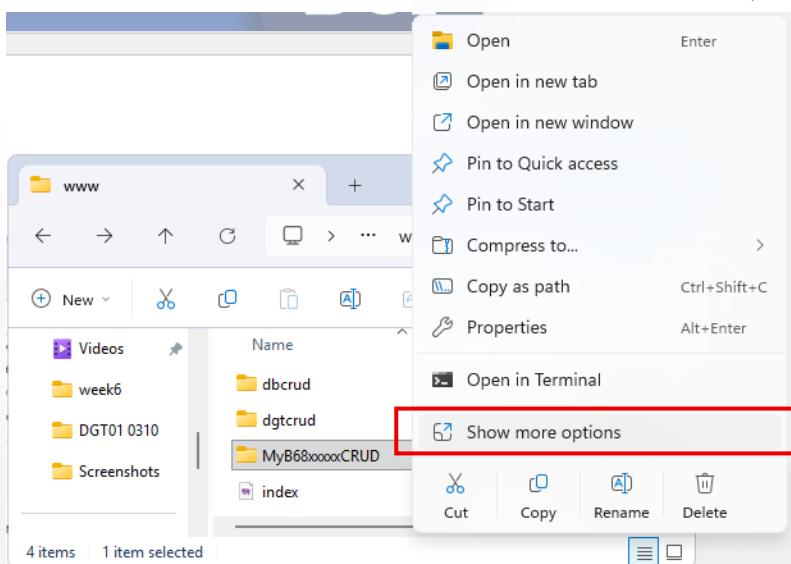
```
C:\laragon\www
λ ls
dbcrud/ index.php
dgtcrud/ MyB68xxxxCRUD/
C:\laragon\www
λ cd MyB68xxxxCRUD\
C:\laragon\www\MyB68xxxxCRUD
λ
```

10. จากนั้นให้เข้าไปที่โฟลเดอร์ที่เก็บข้อมูลเว็บในโปรแกรม Laragon ผ่าน File explore แล้วเข้าไปที่ โฟลเดอร์ในเครื่องของนักศึกษา ให้ Path ที่อยู่ไฟล์ใกล้เคียงตามนี้ C:\laragon\www\MyB68xxxxCRUD

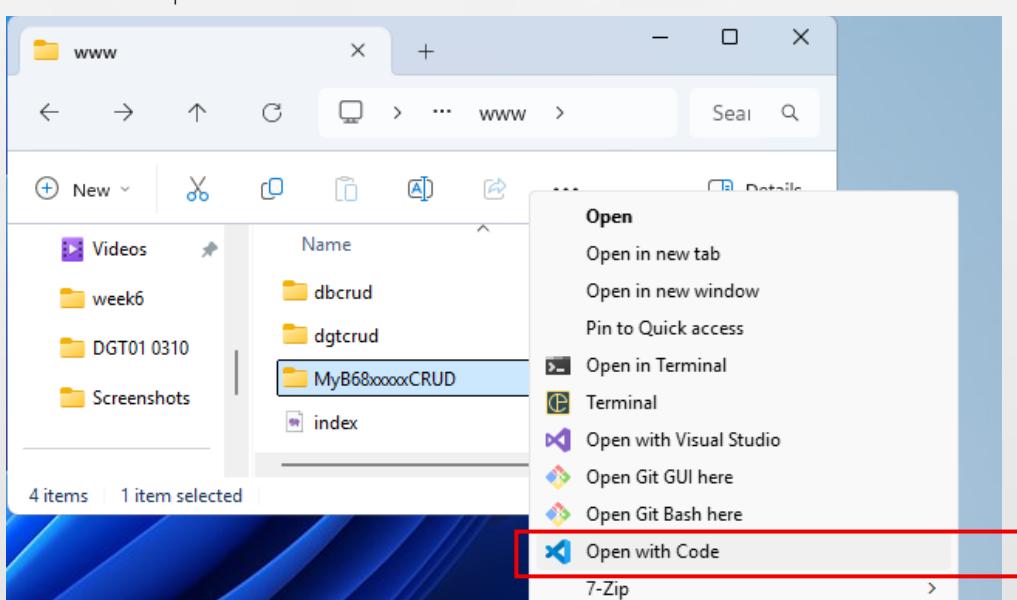




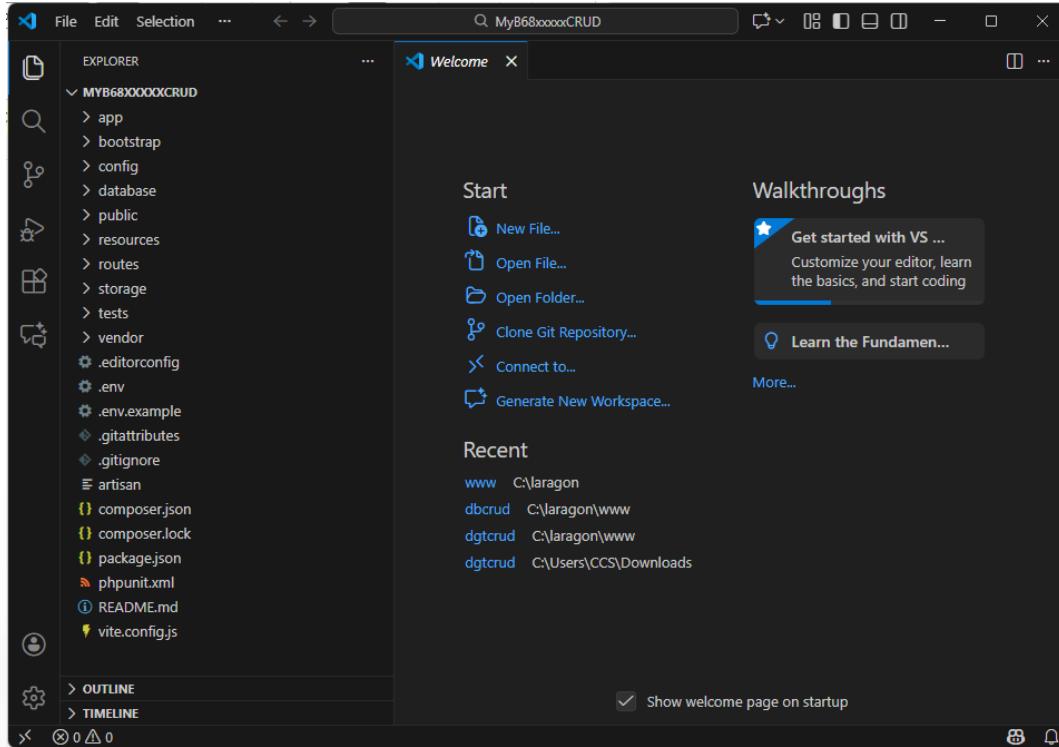
11. จากนั้นคลิกขวาที่ไฟล์เดอร์ MyB68xxxxCRUD > จากนั้น กด “show more options”



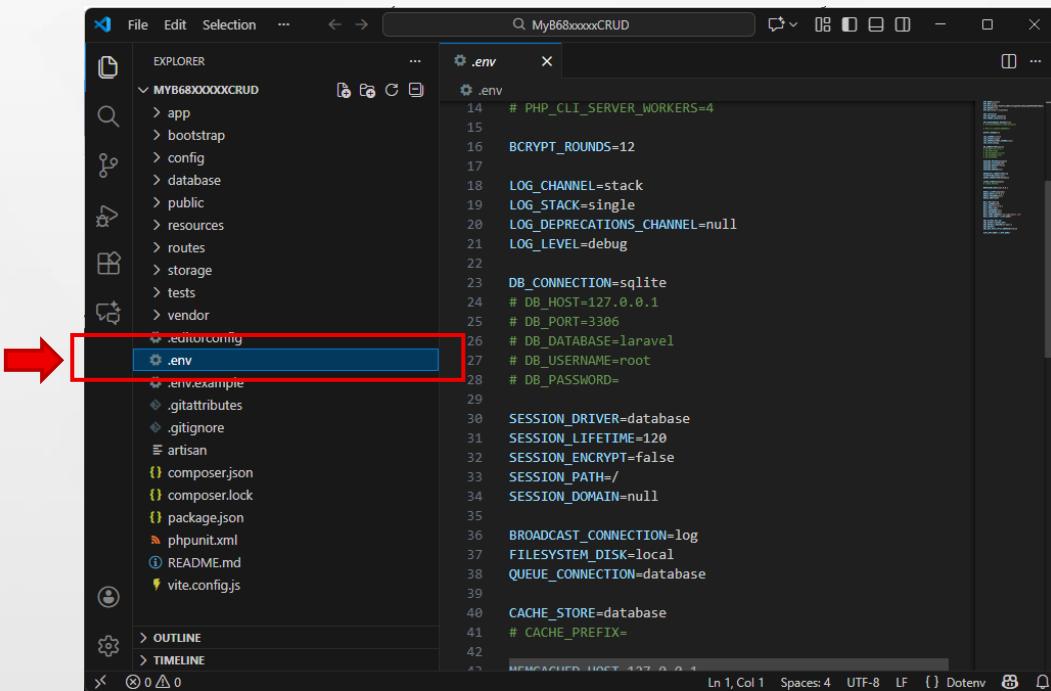
จากนั้นเลือก “Open with Code”



12. จะมีหน้าต่าง vscode ขึ้นมาพร้อมโปรเจคไฟล์โคด์ โฟลเดอร์ และไฟล์ตั้งค่าต่างๆ ตามนี้



13. จากนั้นเปิดไฟล์ .env เพื่อแก้ไขโคด์ในไฟล์ เมื่อเปิดไฟล์แล้วจะมีโคด์ขึ้นมา



ให้แก้ไฟล์ .env และให้หาโคด์ที่เหมือนกับกรอบและลูกศรที่ว่าง จากนั้นแก้ตามรูปต่อไปจากนี้

```

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3307
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=

```

14. จากนั้นกลับไปที่ Terminal ของ Laragon (Cmder)  และพิมพ์คำสั่งต่อไปนี้

`php artisan migrate` จากนั้นกด Enter

```

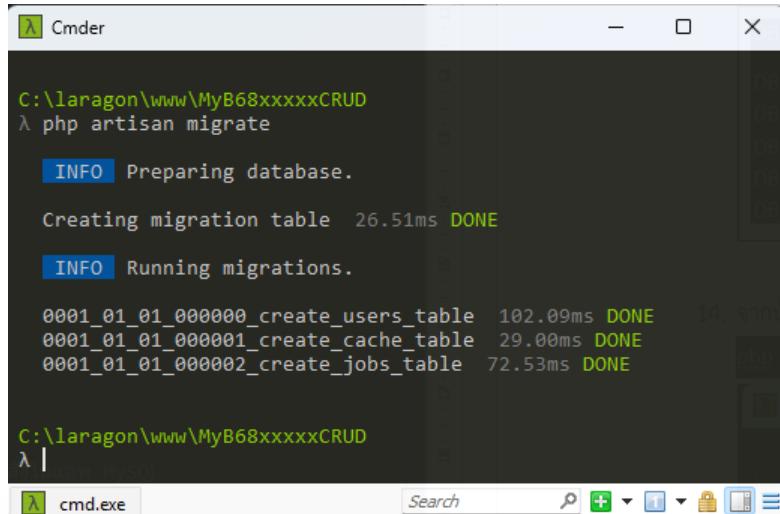
C:\laragon\www
λ ls
dbcrud/ index.php
dgcrud/ MyB68xxxxCRUD/

C:\laragon\www
λ cd MyB68xxxxCRUD\

C:\laragon\www\MyB68xxxxCRUD
λ php artisan migrate

```

15. เมื่อทำการใช้คำสั่งก่อนหน้านี้นั้น หากสามารถรันทำงานได้สำเร็จ จะได้ข้อความเหล่านี้



```
C:\laragon\www\MyB68xxxxCRUD
λ php artisan migrate

INFO Preparing database.

Creating migration table 26.51ms DONE
INFO Running migrations.

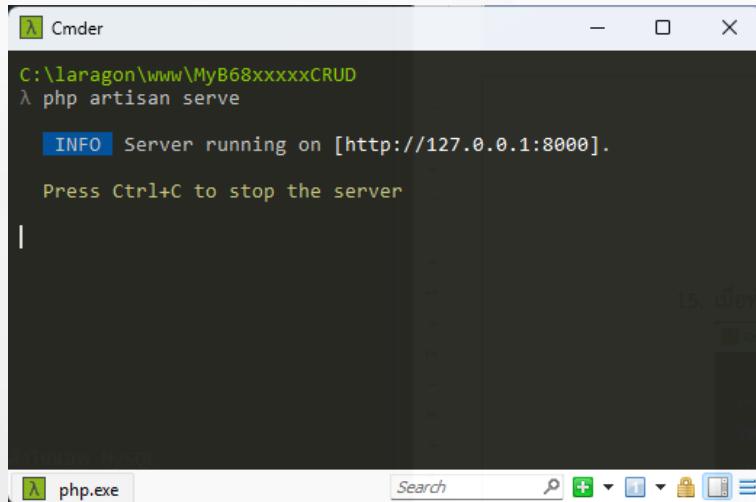
0001_01_01_000000_create_users_table 102.09ms DONE
0001_01_01_000001_create_cache_table 29.00ms DONE
0001_01_01_000002_create_jobs_table 72.53ms DONE
```

16. จากนั้นพิมพ์คำสั่งลงไฟใน Terminal ของ Laragon (Cmder) ด้วยคำสั่งต่อไปนี้ จากนั้นกด Enter เพื่อรันเว็บ

```
php artisan serve
```

จากนั้นจะได้ผลลัพท์ดังรูป จากนั้นให้กดปุ่ม Ctrl ค้างไว้กับตำแหน่ง Maus Cursor ของมาส์เพซีแล้วกดที่ลิ้งค์

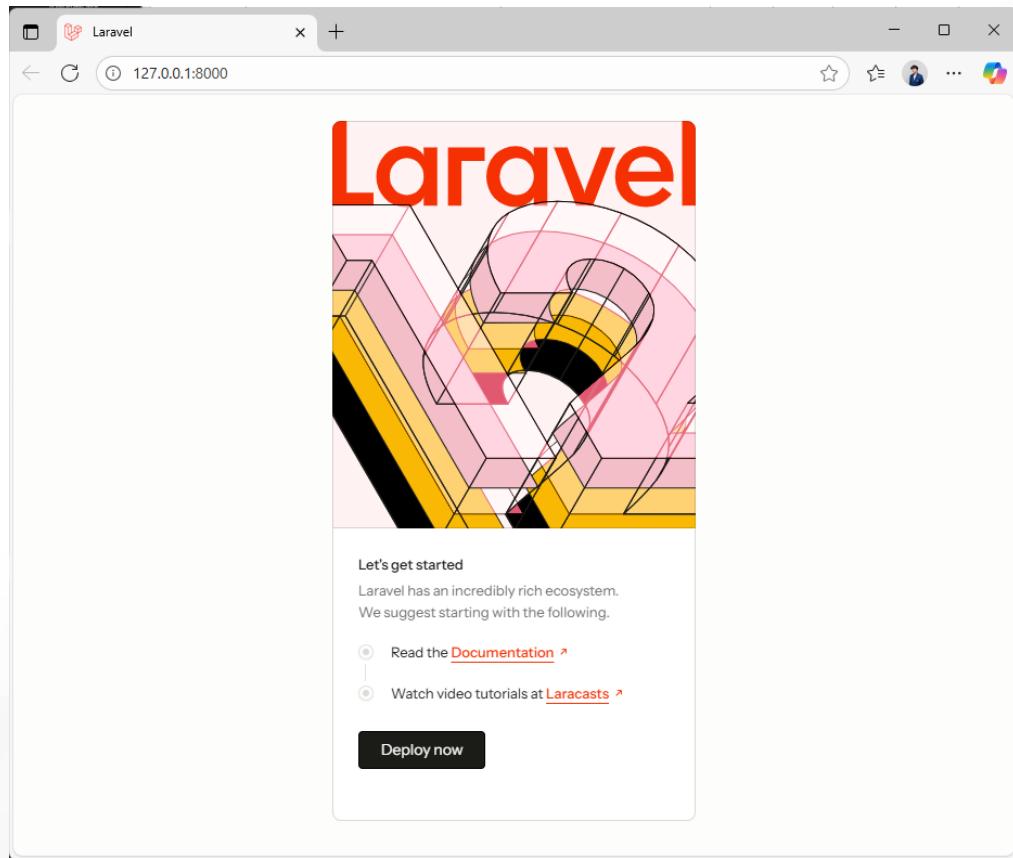
[http://127.0.0.1:8000]



```
C:\laragon\www\MyB68xxxxCRUD
λ php artisan serve

INFO Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server
```

จากนั้นโปรแกรมเริบบรรยายเซอร์ปราภูชั้นมา



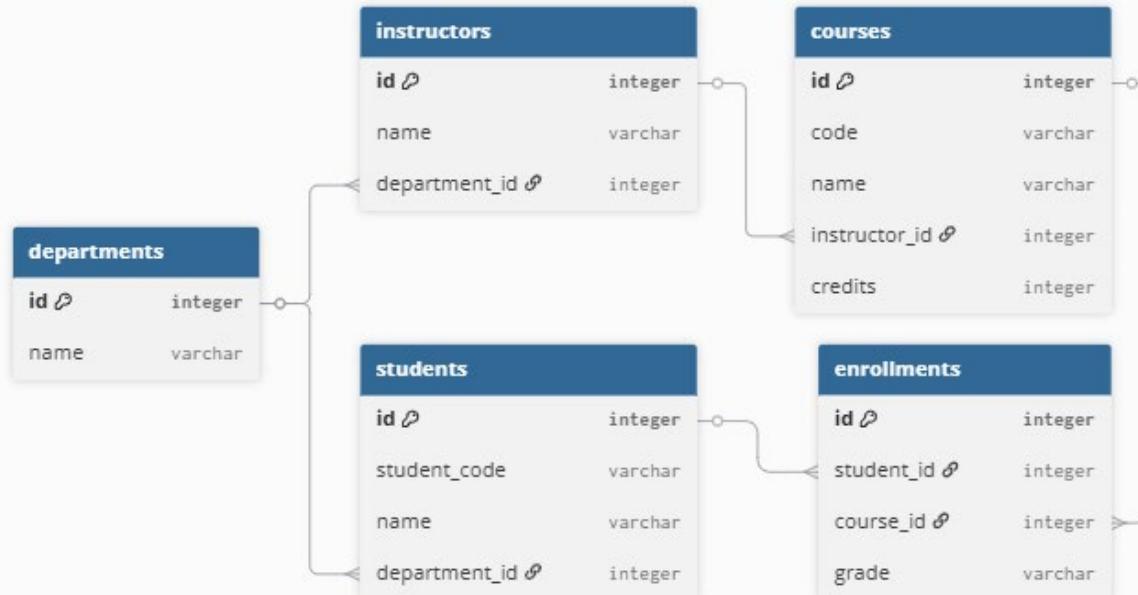
ส่วนที่ 2: การสร้างฐานข้อมูล ตาราง วิว และการเพิ่มข้อมูล

ໂປຣເຈັດຕ່ອໄປນີ້ເປັນການສ້າງໂປຣເຈັດສໍາຫຼັບການຮຽນວິຊາ Database development fundamental ເພຣະຈະໄດ້ເຫັນກາທີ່ກ່າວກຳກອບແບບ (Design), ການຈັດການຂໍ້ມູນຮະດັບ SQL (Views/Joins), ແລກການນຳມາໃໝ່ຈິງ ໃນ Application (Laravel) ໂດຍໂປຣເຈັດນີ້ເກີ່ວກັບ “ຮະບບລົງທະບຽນຮຽນມາວິທາລີຍ (University Course Registration)” ໂດຍເຮີມກະບວນກາຮັດຕ່າງໆ

1. ການອອກແບບฐานขໍ້ມູນ (Database Schema) ເຮັດວຽກກຳນົດກຳນົດ 4 ຕາຮາງຫລັກ (+1 ຕາຮາງສໍາຫຼັບເຂື່ອມຄວາມສັນພັນ) ເພື່ອໃຫ້ສາມາຮັດ Join ກັນໄດ້ຄຽບທຸກຕາຮາງ ໂດຍມີຕາຮາງ ດັ່ງນີ້

Entity Relationship (ER):

- 1.1. Departments (ຄະນະ/ການວິຊາ): ເກີບຂໍ້ມູນສັງກັດ
- 1.2. Students (ນັກສຶກສາ): ສັງກັດການວິຊາ (FK)
- 1.3. Instructors (ອາຈາຍຢີ): ສັງກັດການວິຊາ (FK)
- 1.4. Courses (ວິຊາ): ສອນໂດຍອາຈາຍຢີທ່ານໄໝ (FK)
- 1.5. Enrollments (ກາຮລົງທະບຽນ): ຕາຮາງ Transaction ເຂື່ອມຮ່ວງ Student ແລະ Course



2. ขั้นตอนการทำใน Laravel (Migration & Seeding)

2.1. สร้างตาราง (Migrations)

ให้ใช้คำสั่งต่อไปนี้ในการสร้างตาราง ใน Terminal Laragon (Cmder)

```
php artisan make:migration create_university_tables
```

จากนั้นกด Enter แล้วรอนก่อนกว่าจะขึ้นข้อความ “Created successfully”

The screenshot shows a Cmder terminal window with the following output:

```
λ ls
app/      config/      public/    tests/
artisan*  database/    README.md  vendor/
bootstrap/ myb68xxxxxcrud.sql  resources/ vite.config.js
composer.json  package.json  routes/
composer.lock  phpunit.xml  storage/

C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ php artisan make:migration create_university_tables

[INFO] Migration [C:\laragon\www\MyB68xxxxxCRUD\database\migrations\2025_12_04_030443_create_university_tables.php] created
successfully.

C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ |
```

A red box highlights the terminal output where the migration file was created successfully. A red arrow points from the bottom right towards this highlighted area.

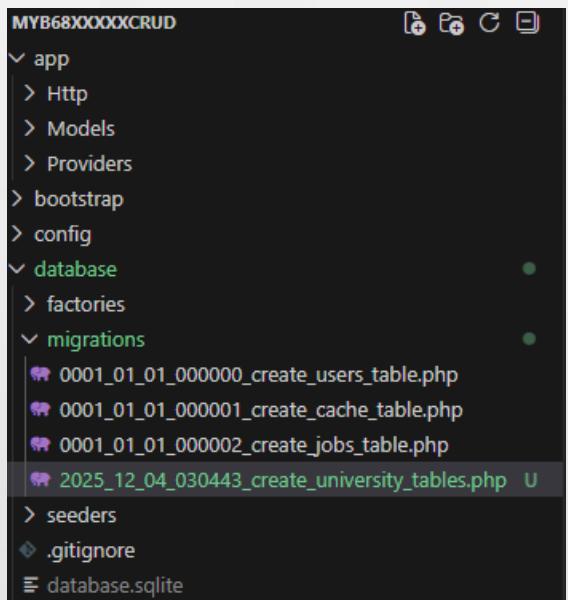
2.2. หลังจากสร้างตารางแล้ว จะได้ไฟล์ขึ้นมา ให้ใส่โค้ดที่ไฟล์ใหม่ต่อไปนี้ในไฟล์ที่มีชื่อว่า

2025_11_18_033331_create_university_tables.php

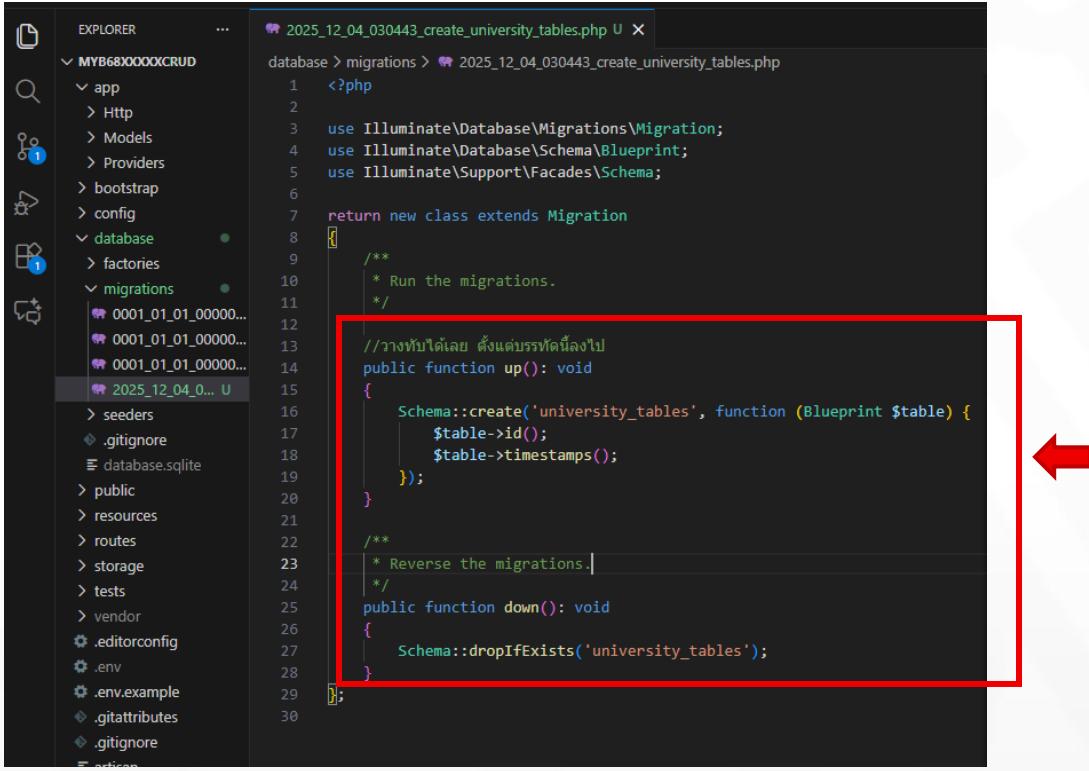
ในโฟลเดอร์ database/migrations

***ชื่อไฟล์อาจไม่เหมือนกันกับตัวอย่าง ให้ดูช่วงข้อความชื่อไฟล์ด้านหลังคือ

create_university_tables.php



จะเห็นว่าให้โค้ด function up() และ down() ขึ้นมา



EXPLORER

MYB68XXXXCRUD

app

Http

Models

Providers

bootstrap

config

database

factories

migrations

0001_01_01_00000...

0001_01_01_00000...

0001_01_01_00000...

2025_12_04_0... U

seeders

.gitignore

database.sqlite

public

resources

routes

storage

tests

vendor

.editorconfig

.env

.env.example

.gitattributes

.gitignore

.gitignore

artisan

2025_12_04_030443_create_university_tables.php

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */

    // วางทันใจเลย ตั้งแต่บรรทัดล่องไว
    public function up(): void
    {
        Schema::create('university_tables', function (Blueprint $table) {
            $table->id();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('university_tables');
    }
};
```

2.3. ให้นำโค้ดต่อไปนี้ไปวางทับที่ function up() และ down()

```
public function up(): void
{
    // 1. ตาราง Departments (คณาจารย์)
    Schema::create('departments', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->timestamps();
    });

    // 2. ตาราง Instructors (อาจารย์)
    Schema::create('instructors', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->foreignId('department_id')->constrained('departments');
        $table->timestamps();
    });
}
```

```

// 3. ตาราง Students (นักศึกษา)
Schema::create('students', function (Blueprint $table) {
    $table->id();
    $table->string('student_code');
    $table->string('name');
    $table->foreignId('department_id')->constrained('departments');
    $table->timestamps();
});

// 4. ตาราง Courses (วิชา)
Schema::create('courses', function (Blueprint $table) {
    $table->id();
    $table->string('code');
    $table->string('name');
    $table->foreignId('instructor_id')->constrained('instructors');
    $table->integer('credits');
    $table->timestamps();
});

// 5. ตาราง Enrollments (การลงทะเบียน)
Schema::create('enrollments', function (Blueprint $table) {
    $table->id();
    $table->foreignId('student_id')->constrained('students')-
>onDelete('cascade');
    $table->foreignId('course_id')->constrained('courses')-
>onDelete('cascade');
    $table->string('grade')->nullable();
    $table->timestamps();
});
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::dropIfExists('enrollments');
    Schema::dropIfExists('courses');
    Schema::dropIfExists('students');
    Schema::dropIfExists('instructors');
    Schema::dropIfExists('departments');
}
// รื้อสิ่งที่ไม่ต้องการ

```

โค้ดที่วางจะได้แบบนี้

```
database > migrations > 2025_12_04_030443_create_university_tables.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10      * Run the migrations.
11      */
12
13      // งานที่ต้องทำเมื่อต้องการสร้างฐานข้อมูล
14      public function up(): void
15      {
16          // 1. ตาราง Departments (គម្រោង)
17          Schema::create('departments', function (Blueprint $table) {
18              $table->id();
19              $table->string('name');
20              $table->timestamps();
21          });
22
23          // 2. ตาราง Instructors (ភាគចាយ)
24          Schema::create('instructors', function (Blueprint $table) {
25              $table->id();
26              $table->string('name');
27              $table->foreignId('department_id')->constrained('departments');
28              $table->timestamps();
29          });
23
30          // 3. ตาราง Students (ឯកសារឈ្មោះ)
31          Schema::create('students', function (Blueprint $table) {
32              $table->id();
33              $table->string('student_code');
34              $table->string('name');
35              $table->foreignId('department_id')->constrained('departments');
36              $table->timestamps();
37          });
38
39          // 4. ตาราง Courses (វិទ្យា)
40          Schema::create('courses', function (Blueprint $table) {
41              $table->id();
42              $table->string('code');
43              $table->string('name');
44              $table->foreignId('instructor_id')->constrained('instructors');
45              $table->integer('credits');
46              $table->timestamps();
47          });
48
49          // 5. ตาราง Enrollments (ការសិក្សាបើណ្ហាយ)
50          Schema::create('enrollments', function (Blueprint $table) {
51              $table->id();
52              $table->foreignId('student_id')->constrained('students')->onDelete('cascade');
53              $table->foreignId('course_id')->constrained('courses')->onDelete('cascade');
54              $table->string('grade')->nullable();
55              $table->timestamps();
56          });
57      }
58
59      /**
60      * Reverse the migrations.
61      */
62
63      public function down(): void
64      {
65          Schema::dropIfExists('enrollments');
66          Schema::dropIfExists('courses');
67          Schema::dropIfExists('students');
68          Schema::dropIfExists('instructors');
69          Schema::dropIfExists('departments');
70      }
71      // สิ้นสุดการวางที่บีบ
72
73  };
```

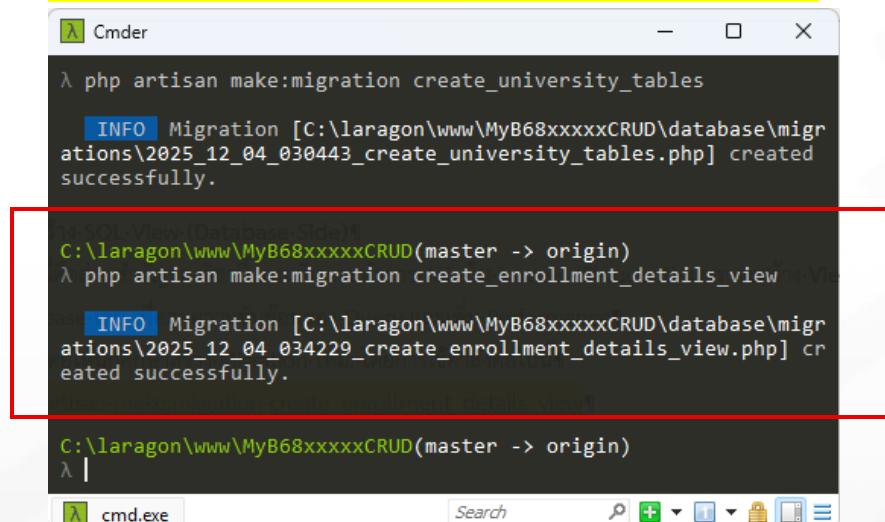


3. การสร้าง SQL View (Database Side)

นี่คือส่วนสำคัญของวิชานี้ แทนที่เราจะทำการ Join ใน Laravel (Eloquent) แต่เราจะสร้าง View ไว้ที่ Database เลย เพื่อลดความซับซ้อนของ Query และเพิ่ม Performance

3.1. โดยให้นักศึกษาสร้าง Migration ใหม่: โดยการใช้คำสั่งต่อไปนี้

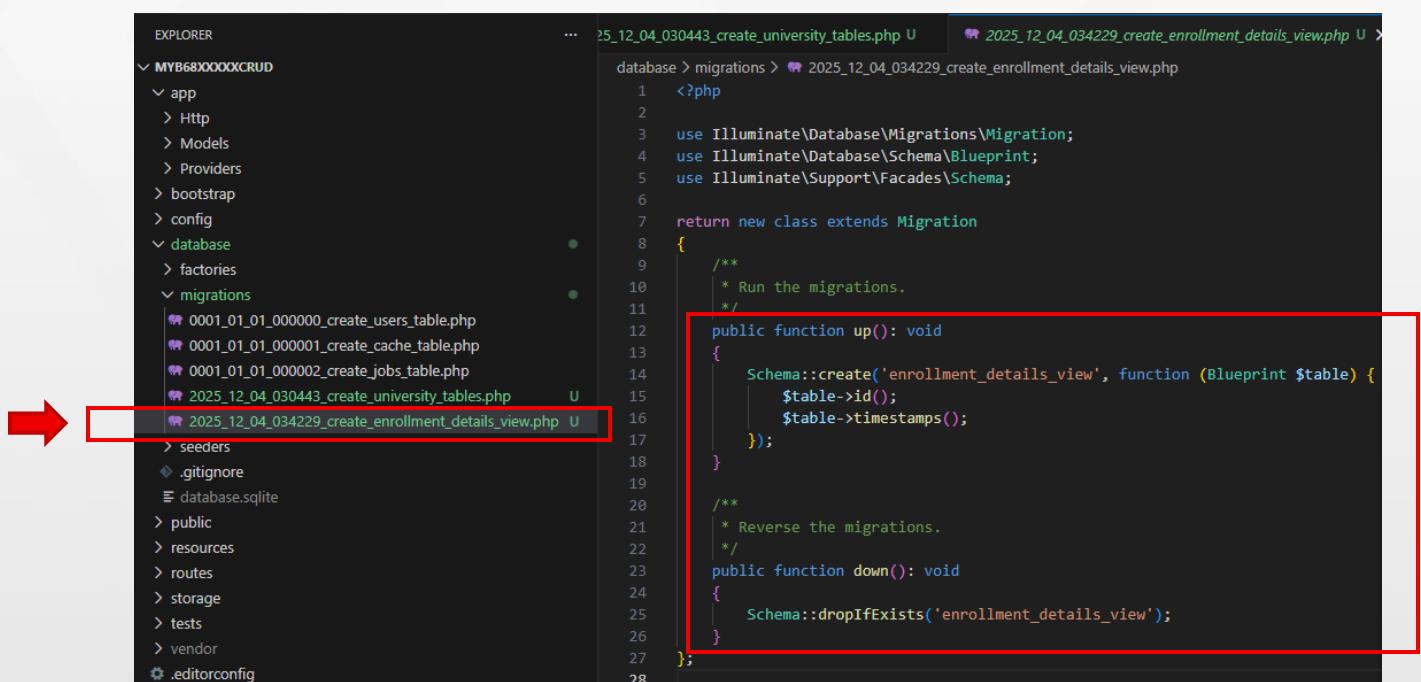
```
php artisan make:migration create_enrollment_details_view
```



```
C:\laragon\www\MyB68xxxxCRUD(master -> origin)
λ php artisan make:migration create_university_tables
INFO Migration [C:\laragon\www\MyB68xxxxCRUD\database\migrations\2025_12_04_030443_create_university_tables.php] created successfully.

C:\laragon\www\MyB68xxxxCRUD(master -> origin)
λ php artisan make:migration create_enrollment_details_view
INFO Migration [C:\laragon\www\MyB68xxxxCRUD\database\migrations\2025_12_04_034229_create_enrollment_details_view.php] created successfully.
```

จะได้ไฟล์ขึ้นมา และโค้ดดังกล่าว



EXPLORER

- MYB68XXXXCRUD
 - app
 - Http
 - Models
 - Providers
 - bootstrap
 - config
 - database
 - factories
 - migrations
 - 0001_01_01_00000_create_users_table.php
 - 0001_01_01_00001_create_cache_table.php
 - 0001_01_01_00002_create_jobs_table.php
 - 2025_12_04_030443_create_university_tables.php
 - 2025_12_04_034229_create_enrollment_details_view.php
 - seeders
 - .gitignore
 - database.sqlite
 - public
 - resources
 - routes
 - storage
 - tests
 - vendor
 - .editorconfig

```
25_12_04_030443_create_university_tables.php U 2025_12_04_034229_create_enrollment_details_view.php U
database > migrations > 2025_12_04_034229_create_enrollment_details_view.php
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9
10 /**
11 * Run the migrations.
12 */
13
14 public function up(): void
15 {
16     Schema::create('enrollment_details_view', function (Blueprint $table) {
17         $table->id();
18         $table->timestamps();
19     });
20 }
21
22 /**
23 * Reverse the migrations.
24 */
25 public function down(): void
26 {
27     Schema::dropIfExists('enrollment_details_view');
28 }
```

***ชื่อไฟล์อาจไม่เหมือนกันกับตัวอย่าง ให้ดูช่วงข้อความชื่อไฟล์ด้านหลัง

3.2. นำโค้ดต่อไปนี้ไปวางทับที่ function up() และ down()

```
// งานทั่วไปเพื่อเตรียมตั้งแต่บรรทัดนี้ลงไป

public function up(): void
{
    DB::statement("
        CREATE OR REPLACE VIEW enrollment_details AS
        SELECT
            e.id as enrollment_id,
            s.student_code,
            s.name as student_name,
            d.name as department_name,
            c.code as course_code,
            c.name as course_name,
            i.name as instructor_name,
            e.grade,
            e.student_id,
            e.course_id,
            e.created_at
        FROM enrollments e
        JOIN students s ON e.student_id = s.id
        JOIN departments d ON s.department_id = d.id
        JOIN courses c ON e.course_id = c.id
        JOIN instructors i ON c.instructor_id = i.id
    ");
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    DB::statement("DROP VIEW IF EXISTS enrollment_details");
}
// ลื้นสุดการวางแผน
```

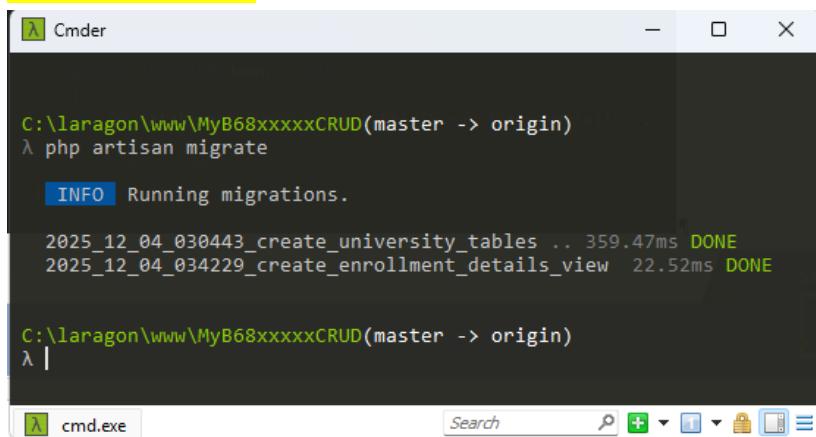
ตัวอย่างการวางแผนทับโค้ดของไฟล์ 2025_12_04_034229_create_enrollment_details_view.php

***ระวังวางแผนทับโค้ด ระวังปีกกา

```
database > migrations > 2025_12_04_034229_create_enrollment_details_view.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10      * Run the migrations.
11      */
12     // วางแผนทับได้เลย ตั้งแต่บรรทัดนี้ลงไป
13     public function up(): void
14     {
15         DB::statement([
16             'CREATE OR REPLACE VIEW enrollment_details AS
17             SELECT
18                 e.id as enrollment_id,
19                 s.student_code,
20                 s.name as student_name,
21                 d.name as department_name,
22                 c.code as course_code,
23                 c.name as course_name,
24                 i.name as instructor_name,
25                 e.grade,
26                 e.student_id,
27                 e.course_id,
28                 e.created_at
29             FROM enrollments e
30             JOIN students s ON e.student_id = s.id
31             JOIN departments d ON s.department_id = d.id
32             JOIN courses c ON e.course_id = c.id
33             JOIN instructors i ON c.instructor_id = i.id
34             "]);
35     }
36
37     /**
38     * Reverse the migrations.
39     */
40     public function down(): void
41     {
42         DB::statement("DROP VIEW IF EXISTS enrollment_details");
43     }
44     // สิ้นสุดการวางแผน
45 };
46
```

3.3. สั่งรัน Migration พิมพ์คำสั่งต่อไปนี้

```
php artisan migrate
```



```
C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ php artisan migrate

INFO  Running migrations.

2025_12_04_030443_create_university_tables .. 359.47ms DONE
2025_12_04_034229_create_enrollment_details_view 22.52ms DONE

C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ |
```

(ตอนนี้ใน Database จะมีตารางครบ และมี View ชื่อ enrollment_details โผล่มาแล้ว)

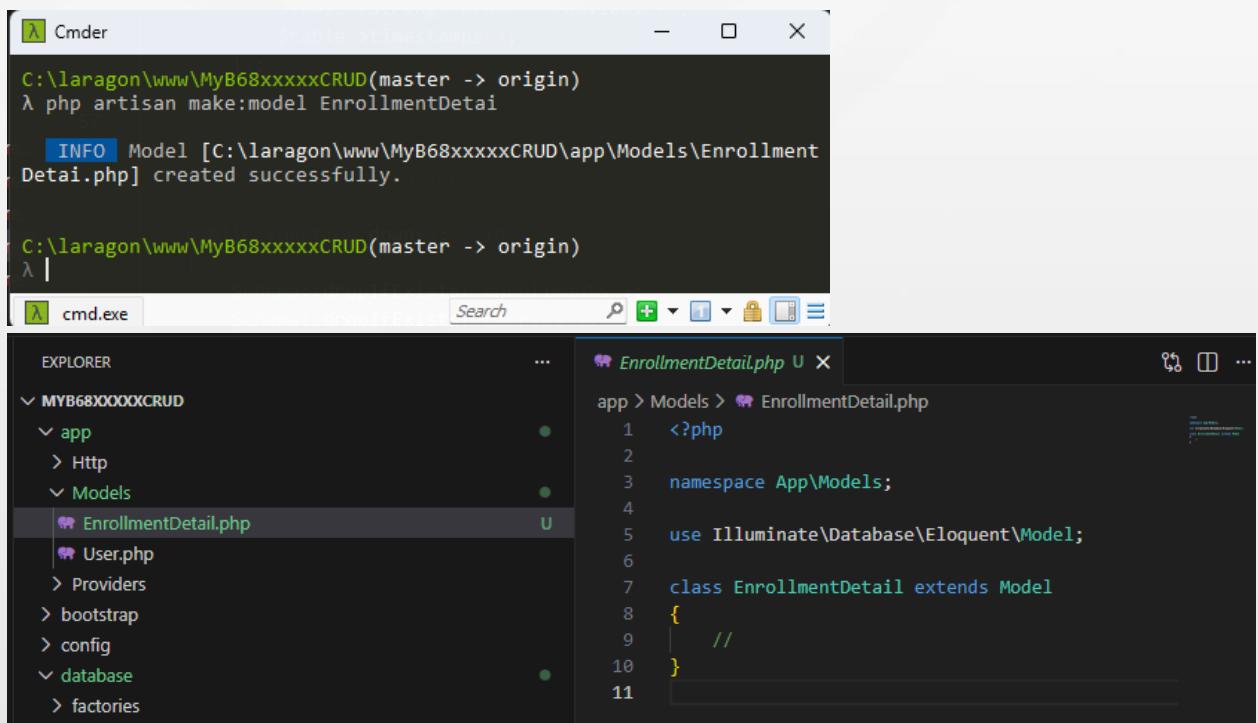
3.4. การเชื่อมต่อและทำ CRUD ใน Laravel

Model Setup

สร้าง Model สำหรับ View ด้วย (ทำเหมือนตารางปกติ แต่ปิด timestamp) ใช้คำสั่ง

```
php artisan make:model EnrollmentDetail
```

เมื่อรันคำสั่งเสร็จแล้วจะได้ไฟล์ EnrollmentDetail.php



```
C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ php artisan make:model EnrollmentDetail

INFO Model [C:\laragon\www\MyB68xxxxxCRUD\app\Models\EnrollmentDetail.php] created successfully.

C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ |
```

EXPLORER

- MYB68XXXXXCRUD
 - app
 - Http
 - Models
 - EnrollmentDetail.php
 - User.php
 - Providers
 - bootstrap
 - config
 - database
 - factories

EnrollmentDetail.php

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class EnrollmentDetail extends Model
{}
```

3.5. เปิดไฟล์ EnrollmentDetail.php ในโฟล์เดอร์ app/Models และเพิ่มโค้ดตามกรอบ

```

EXPLORER
...
app > Models > EnrollmentDetail.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class EnrollmentDetail extends Model
8  {
9      //
10     public $table = 'enrollment_details'; // ใช้ไปที่ View
11     public $timestamps = false; // View ไม่มี timestamp ของตัวเอง
12     // ใช้สำหรับ Read (R) เท่านั้น
13 }
14

```

โค้ด

```

public $table = 'enrollment_details'; // ใช้ไปที่ View
public $timestamps = false; // View ไม่มี timestamp ของตัวเอง
// ใช้สำหรับ Read (R) เท่านั้น

```

3.6. สร้าง Model ที่เหลือ ด้วยคำสั่ง ต่อไปนี้

```

php artisan make:model Department
php artisan make:model Instructor
php artisan make:model Student
php artisan make:model Course
php artisan make:model Enrollment

```

***ผลการรันอยู่ทางด้านขวา呢

จะได้ไฟล์ตามนี้

```

C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ php artisan make:model Department
[INFO] Model [C:\laragon\www\MyB68xxxxxCRUD\app\Models\Department.php] created successfully.

C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ php artisan make:model Instructor
[INFO] Model [C:\laragon\www\MyB68xxxxxCRUD\app\Models\Instructor.php] created successfully.

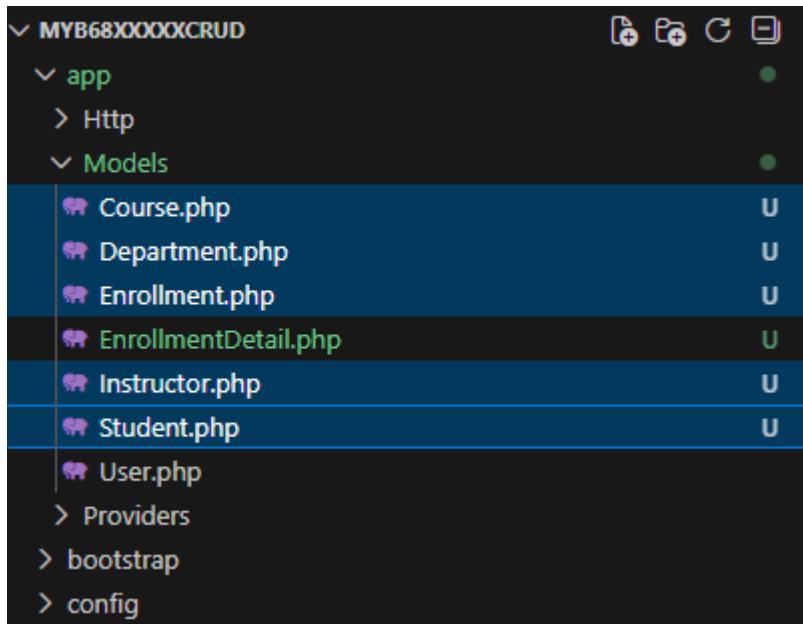
C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ php artisan make:model Student
[INFO] Model [C:\laragon\www\MyB68xxxxxCRUD\app\Models\Student.php] created successfully.

C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ php artisan make:model Course
[INFO] Model [C:\laragon\www\MyB68xxxxxCRUD\app\Models\Course.php] created successfully.

C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ php artisan make:model Enrollment
[INFO] Model [C:\laragon\www\MyB68xxxxxCRUD\app\Models\Enrollment.php] created successfully.

C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ |

```



3.7. จากนั้นเปิดไฟล์ที่มีชื่อว่า “Enrollment.php” และเพิ่มโค้ดต่อไปนี้

```
protected $fillable = ['student_id', 'course_id', 'grade'];
```

```

MYB68XXXXCRUD          app > Models > Enrollment.php
app
> Http
Models
Course.php
Department.php
Enrollment.php
EnrollmentDetail.php
Instructor.php
Student.php
User.php
Providers
bootstrap
config

```

```

1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Enrollment extends Model
8  {
9      //
10     protected $fillable = ['student_id', 'course_id', 'grade'];
11 }
12

```

3.8. จากนั้นให้เปิดไฟล์ Student.php ที่อยู่ในโฟลเดอร์ app/Models/ และให้เพิ่ม โค้ดและตัวแปรตามรูป ตัวอย่างและโค้ดที่ล้อมกรอบแดงไว้ ดังนี้

EXPLORER

MYB68XXXXCRUD

app

- Http\Controllers
- Controller.php
- EnrollmentController.php
- Models
- Course.php
- Department.php
- Enrollment.php
- EnrollmentDetail.php
- Instructor.php
- Student.php
- User.php

Providers

- AppServiceProvider.php

bootstrap

config

database

factories

... EnrollmentDetail.php U Enrollment.php U Student.php X

app > Models > Student.php

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Student extends Model
9 {
10    //
11    use HasFactory;
12
13    // เพิ่มบรรทัดนี้ลงไปครับ เพื่ออนุญาตให้บันทึกข้อมูล 3 ช่องนี้ได้
14    protected $fillable = [
15        'student_code',
16        'name',
17        'department_id'
18    ];
19 }
20
```

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Student extends Model
{
    //
    use HasFactory;

    // เพิ่มบรรทัดนี้ลงไปครับ เพื่ออนุญาตให้บันทึกข้อมูล 3 ช่องนี้ได้
    protected $fillable = [
        'student_code',
        'name',
        'department_id'
    ];
}
```

3.9. สร้างข้อมูลจำลอง (Seeding)

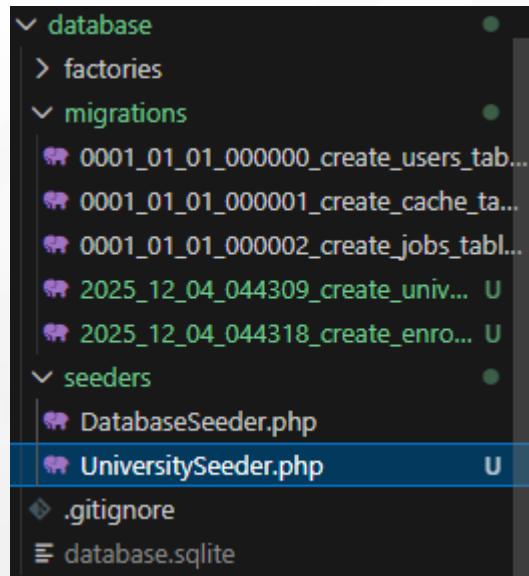
ไฟล์นี้ทำให้นักศึกษามีข้อมูลเล่นทันที โดยไม่ต้องนั่งกรอกเอง สร้าง Seeder โดยพิมพ์คำสั่ง:

```
php artisan make:seeder UniversitySeeder
```

```
C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ php artisan make:seeder UniversitySeeder

[INFO] Seeder [C:\laragon\www\MyB68xxxxxCRUD\database\seeders\UniversitySeeder.php] created successfully.
```

จะได้ไฟล์ UniversitySeeder.php ที่อยู่ภายใต้โฟลเดอร์ database/migrations/seeders



จากนั้นให้เพิ่มโค้ดตามรูปลงไป

***รูปและโค้ดอยู่ในหน้าถัดไป

```

database > seeders > UniversitySeeder.php
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7
8  use App\Models\Department;
9  use App\Models\Instructor;
10 use App\Models\Student;
11 use App\Models\Course;
12
13 class UniversitySeeder extends Seeder
{
14
15     /**
16      * Run the database seeds.
17     */
18     public function run(): void
19     {
20         /*
21          // 1. สร้างคณะ
22          $cs = Department::create(['name' => 'วิทยาการคอมพิวเตอร์']);
23          $it = Department::create(['name' => 'เทคโนโลยีสารสนเทศ']);
24
25          // 2. สร้างอาจารย์
26          $aj_a = Instructor::create(['name' => 'ดร.สมชาย ใจดี', 'department_id' => $cs->id]);
27          $aj_b = Instructor::create(['name' => 'อ.วิภา วงศ์เรือง', 'department_id' => $it->id]);
28
29          // 3. สร้างนักศึกษา
30          Student::create(['student_code' => '66001', 'name' => 'นาย ก.', 'department_id' => $cs->id]);
31          Student::create(['student_code' => '66002', 'name' => 'นางสาว ข.', 'department_id' => $it->id]);
32
33          // 4. สร้างวิชา
34          Course::create(['code' => 'CS101', 'name' => 'เขียนโปรแกรมเบื้องต้น', 'instructor_id' => $aj_a->id, 'credits' => 3]);
35          Course::create(['code' => 'IT202', 'name' => 'ฐานข้อมูล', 'instructor_id' => $aj_b->id, 'credits' => 3]);
36
37     }
38 }

```

***โค้ดอยู่ที่หน้าต่อไป

```

<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

use App\Models\Department;
use App\Models\Instructor;
use App\Models\Student;
use App\Models\Course;

class UniversitySeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        //
        // 1. สร้างคณบดี
        $cs = Department::create(['name' => 'วิทยาการคอมพิวเตอร์']);
        $it = Department::create(['name' => 'เทคโนโลยีสารสนเทศ']);

        // 2. สร้างอาจารย์
        $aj_a = Instructor::create(['name' => 'ดร.สมชาย ใจดี', 'department_id' => $cs->id]);
        $aj_b = Instructor::create(['name' => 'อ.วิภาดา รักเรียน', 'department_id' => $it->id]);

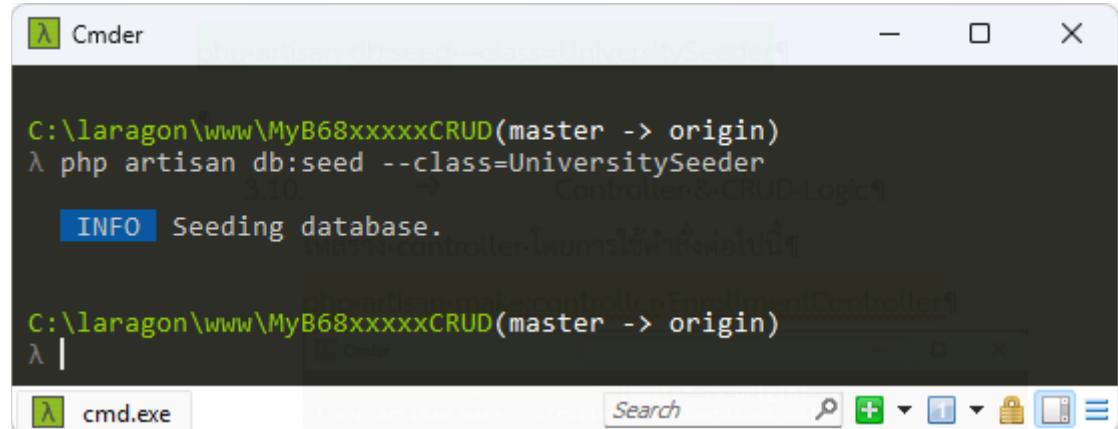
        // 3. สร้างนักศึกษา
        Student::create(['student_code' => '66001', 'name' => 'นาย ก.', 'department_id' => $cs->id]);
        Student::create(['student_code' => '66002', 'name' => 'นางสาว ข.', 'department_id' => $it->id]);

        // 4. สร้างวิชา
        Course::create(['code' => 'CS101', 'name' => 'ตีyan โปรแกรมเมืองด้น', 'instructor_id' => $aj_a->id, 'credits' => 3]);
        Course::create(['code' => 'IT202', 'name' => 'ฐานข้อมูล', 'instructor_id' => $aj_b->id, 'credits' => 3]);
    }
}

```

จากนั้นให้รันคำสั่ง seed ตามคำสั่งด้านล่าง

```
php artisan db:seed --class=UniversitySeeder
```



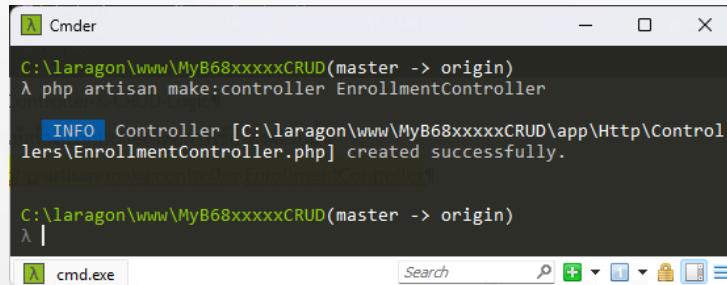
```
C:\laragon\www\MyB68xxxxCRUD(master -> origin)
λ php artisan db:seed --class=UniversitySeeder
      3.10.          → Controller & CRUD Logic
[INFO] Seeding database.

C:\laragon\www\MyB68xxxxCRUD(master -> origin)
λ |
```

3.10. Controller & CRUD Logic

ให้สร้าง controller โดยการใช้คำสั่งต่อไปนี้

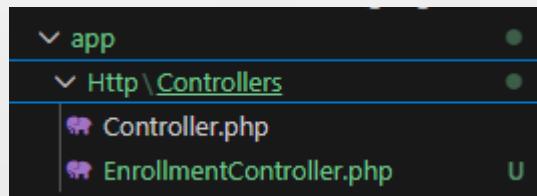
```
php artisan make:controller EnrollmentController
```



```
C:\laragon\www\MyB68xxxxCRUD(master -> origin)
λ php artisan make:controller EnrollmentController
[INFO] Controller [C:\laragon\www\MyB68xxxxCRUD\app\Http\Controllers\EnrollmentController.php] created successfully.

C:\laragon\www\MyB68xxxxCRUD(master -> origin)
λ |
```

เราจะได้ไฟล์ของ controller ขึ้นมา ดังรูปภาพ



3.11. ให้เพิ่มโค๊ดต่อไปนี้ตามตัวอย่างรูปภาพ ที่ไฟล์ EnrollmentController.php

EXPLORER

MYB68XXXXCRUD

- app
 - Http\Controllers\Controller.php
 - EnrollmentController.php
 - Models
 - Course.php
 - Department.php
 - Enrollment.php
 - EnrollmentDetail.php
 - Instructor.php
 - Student.php
 - User.php
 - Providers
 - AppServiceProvider.php
 - bootstrap
 - config
 - database
 - factories
 - migrations
 - 0001_01_01_00000_create_users_table.php
 - 0001_01_01_00001_create_cache_table.php
 - 0001_01_01_00002_create_jobs_table.php
 - 2025_12_04_044309_create_university.php
 - 2025_12_04_044318_create_enrollment.php
 - seeders
 - DatabaseSeeder.php
 - UniversitySeeder.php
 - .gitignore
 - database.sqlite
 - public
 - resources
 - routes
 - storage
 - tests
 - vendor
- .editorconfig
- .env
- .env.example
- .gitattributes
- .gitignore
- OUTLINE
- TIMELINE

... EnrollmentDetail.php U Enrollment.php U Student.php U UniversitySeeder.php

```

app > Http > Controllers > EnrollmentController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  use App\Models\EnrollmentDetail;
8  use App\Models\Enrollment;
9  use App\Models\Student;
10 use App\Models\Course;
11 use App\Models\Department;
12
13 class EnrollmentController extends Controller
14 {
15     //
16     public function index()
17     {
18         $enrollments = EnrollmentDetail::all();
19         $students = Student::all();
20         $courses = Course::all();
21         $departments = Department::all(); // <--- 1. เพิ่มบรรทัดนี้
22
23         // ส่ง $departments ไปด้วย
24         return view('enrollments.index',
25             compact('enrollments',
26                 'students',
27                 'courses',
28                 'departments'));
29     }
30
31     public function storeStudent(Request $request)
32     {
33         // เช็คฟอร์มต้องห้าม
34         Student::create($request->validate([
35             'student_code' => 'required|unique:students',
36             'name' => 'required',
37             'department_id' => 'required'
38         ]));
39
40         return back()->with('success', 'เพิ่มรายชื่อนักศึกษาใหม่เรียบร้อย');
41     }
42
43     public function showStudent($id)
44     {
45         // 1. Master Data: ข้อมูลนักศึกษา + คณะ
46         $student = Student::with('department')->findOrFail($id);
47
48         // 2. Details Data: ดึงจาก View ที่เราทำไว้ (Where ตาม student_id)
    
```

มีต่อหน้าถัดไป

EXPLORER

MYB68XXXXCRUD

- app
 - Http\Controllers\Controller.php
 - EnrollmentController.php
- Models
 - Course.php
 - Department.php
 - Enrollment.php
 - EnrollmentDetail.php
 - Instructor.php
 - Student.php
 - User.php
- Providers
 - AppServiceProvider.php
- bootstrap
- config
- database
 - factories
 - migrations
 - 0001_01_01_00000_create_users_table.php
 - 0001_01_01_00001_create_cache_table.php
 - 0001_01_01_00002_create_jobs_table.php
 - 2025_12_04_044309_create_university.php
 - 2025_12_04_044318_create_enrollment.php
 - seeders
 - DatabaseSeeder.php
 - UniversitySeeder.php
- .gitignore
- database.sqlite
- public
- resources
- routes
- storage
- tests
- vendor

.editorconfig
.env
.env.example
.gitattributes
.aitianore
OUTLINE
TIMELINE

EnrollmentDetail.php U Enrollment.php U Student.php U UniversitySeeder.php

```

app > Http > Controllers > EnrollmentController.php
14 {
43     public function showStudent($id)
44     {
45         // 1. Master Data: ข้อมูลนักศึกษา + คณะ
46         $student = Student::with('department')->findOrFail($id);
47
48         // 2. Details Data: ดึงจาก View ที่เราทำไว้ (Where ตาม student_id)
49         $history = EnrollmentDetail::where('student_id', $id)->get();
50
51         // 3. คำนวณหน่วยกิตรวม (ลอกเล่นเพิ่มเติม)
52         $totalCredits = $history->sum(function($row) {
53             // (ต้องถึง credit จาก Course จะถึง หรือจะ Join มาใน View ได้)
54             // เพื่อความง่ายในการดูยังไงนี่ ผูกของข้างการตั้ง Credit ไปก่อน หรือ
55             // ถ้าใน View ยังไม่มี credit ให้กับส่วนไปแทรก View เพิ่ม column c.credits
56             return $row->credit;
57         });
58
59         return view('enrollments.student_detail', compact('student', 'history'));
60     }
61
62     public function store(Request $request)
63     {
64         // บันทึกข้อมูลลง Table จริง
65         Enrollment::create($request->validate([
66             'student_id' => 'required',
67             'course_id' => 'required',
68             'grade' => 'nullable'
69         ]));
70         return back()->with('success', 'ลงทะเบียนเรียนร้อยแล้ว');
71     }
72
73     public function update(Request $request, $id)
74     {
75         // อัปเดตเกรด
76         $enrollment = Enrollment::find($id);
77         $enrollment->update(['grade' => $request->grade]);
78         return back()->with('success', 'อัปเดตเกรดเรียบร้อย');
79     }
80
81     public function destroy($id)
82     {
83         // ลบข้อมูล
84         Enrollment::destroy($id);
85         return back()->with('success', 'ถอนรายวิชาเรียนร้อย');
86     }
87
88     // เพิ่มฟังก์ชันนี้ตอนท้ายสุดใน Controller
89     public function report(Request $request)

```

มีต่อในหน้าลัดไป

```

EXPLORER ... EnrollmentDetail.php U Enrollment.php U Student.php U UniversitySeeder.php
MYB68xxxxCRUD app Http\Controllers EnrollmentController.php
    app Controller.php EnrollmentController.php U
        Models Course.php Department.php Enrollment.php EnrollmentDetail.php Instructor.php Student.php User.php
        Providers AppServiceProvider.php
            bootstrap config
        database factories migrations
            0001_01_01_000000_create_users_table.php 0001_01_01_000001_create_cache_table.php 0001_01_01_000002_create_jobs_table.php 2025_12_04_044309_create_university_table.php 2025_12_04_044318_create_enrollments_table.php
        seeders
... EnrollmentDetail.php U Enrollment.php U Student.php U UniversitySeeder.php
app > Http > Controllers > EnrollmentController.php
14 {
    // เพิ่มฟังก์ชันนี้ตอบท้ายสุดใน Controller
    public function report(Request $request)
    {
        // 1. ดึงรายชื่อวิชาทั้งหมด เพื่อเอาไปใส่ใน Dropdown ตัวเลือก
        $courses = Course::all();

        // 2. ตัวแปรเก็บผลลัพธ์ (เริ่มนับเนื่องค่าว่าง)
        $results = collect();
        $selectedCourse = null;

        // 3. ถ้ามีการเลือกวิชา (กดปุ่มสัมภាន)
        if ($request->has('course_id')) {
            $courseId = $request->course_id;

            // ดึงข้อมูลจาก View โดย Filter ตาม Course ID
            $results = EnrollmentDetail::where('course_id', $courseId)
                ->orderBy('student_code') // เรียงตามรหัสนักศึกษา
                ->get();

            // ดึงข้อมูลวิชาที่เลือกมาใช้วิจารณ์
            $selectedCourse = Course::find($courseId);
        }

        return view('enrollments.report',
            compact('courses', 'results', 'selectedCourse'));
    }
}

```

***ถึงค้กคัดลอกโค๊ด

<https://raw.githubusercontent.com/pyrnmsutdgt/MyB68xxxxCRUD/refs/heads/CR/app/Http/Controllers/EnrollmentController.php>

โค๊ดชุดนี้เป็น Controller ใน Laravel ซึ่งทำหน้าที่ควบคุมการทำงานของระบบ "ลงทะเบียนเรียน" โดยจัดการข้อมูลระหว่าง Database (Model) กับหน้าเว็บ (View)

คำอธิบายการทำงานของแต่ละฟังก์ชัน

1. public function index()

หน้าที่: แสดงหน้าแรก (Dashboard หรือหน้ารวม)

- การทำงาน:

- ดึงข้อมูลทั้งหมดจากตาราง EnrollmentDetail (ประวัติการลงทะเบียน), Student (นักศึกษา), Course (วิชา) และ Department (แผนก/คณะ)

- **จุดสังเกต:** มีการเพิ่ม \$departments เข้ามาเพื่อเตรียมส่งไปให้หน้าเว็บใช้ (เช่น ใช้ใน Dropdown เลือกคณาจารย์ตอนสมัครนักศึกษาใหม่)

- **Return:** ส่งข้อมูลทั้งหมดผ่านฟังก์ชัน compact ไปแสดงผลที่หน้า View ชื่อ enrollments.index

2. public function storeStudent(Request \$request)

หน้าที่: บันทึกข้อมูลนักศึกษาใหม่

- **การทำงาน:**

- ทำการ **Validate** (ตรวจสอบ) ค่าที่ส่งมาจากฟอร์มว่า:

- student_code: ห้ามว่าง และต้องไม่ซ้ำกับที่มีอยู่เดิม (unique:students)
- name: ห้ามว่าง
- department_id: ห้ามว่าง

- นำข้อมูลที่ผ่านการตรวจสอบแล้วไปสร้าง Record ใหม่ในตาราง students

- **Return:** รีเฟรชหน้าเดิม (back()) พร้อมข้อความแจ้งเตือนว่าเพิ่มเรียบร้อย

3. public function showStudent(\$id)

หน้าที่: แสดงหน้ารายละเอียดของนักศึกษาคนนั้นๆ (Profile & History)

- **การทำงาน:** แบ่งเป็น 3 ส่วน

1. **Master Data:** ค้นหาด้วย \$id และดึงข้อมูลแผนก (department) มาด้วย หากไม่เจอ จะแจ้ง Error 404 (findOrFail)

2. **Details Data:** ดึงประวัติการลงทะเบียน (EnrollmentDetail) เฉพาะของนักศึกษาคนนี้

3. **Calculation:** โค้ดส่วนนี้เตรียมไว้คำนวณหน่วยกิตรวม (sum) แต่ในตัวอย่างนี้ใส่ return 0 ไว้ เป็นค่าเริ่มต้น (Placeholder)

- **Return:** ส่งข้อมูลนักศึกษาและประวัติการเรียนไปที่ View ชื่อ enrollments.student_detail

4. public function store(Request \$request)

หน้าที่: ลงทะเบียนเรียน (เพิ่มวิชาให้นักศึกษา)

- **การทำงาน:**

- ตรวจสอบข้อมูล (student_id, course_id) ว่าต้องมีค่า
- บันทึกข้อมูลลงตาราง enrollments (ตารางที่เชื่อมระหว่าง นักศึกษา กับ วิชา)

- **Return:** รีเฟรชหน้าเดิม พร้อมแจ้งเตือนว่าลงทะเบียนสำเร็จ

5. public function update(Request \$request, \$id)

หน้าที่: อัปเดตเกรด

- **การทำงาน:**

- ค้นหาข้อมูลการลงทะเบียนตาม \$id (PK ของตาราง enrollments)
- แก้ไขค่าในคอลัมน์ grade ตามที่ส่งมาจากฟอร์ม
- Return: รีเฟรชหน้าเดิม พร้อมแจ้งเตือนว่าอัปเดตเกรดแล้ว

6. public function destroy(\$id)

หน้าที่: ถอนรายวิชา (ลบข้อมูลการลงทะเบียน)

- การทำงาน:
 - ลบข้อมูลในตาราง enrollments ตาม \$id ที่ส่งมา
- Return: รีเฟรชหน้าเดิม พร้อมแจ้งเตือนว่าถอนวิชาเรียบร้อย

7. public function report(Request \$request)

หน้าที่: แสดงรายงานรายชื่อนักศึกษาในแต่ละวิชา (Class Roster)

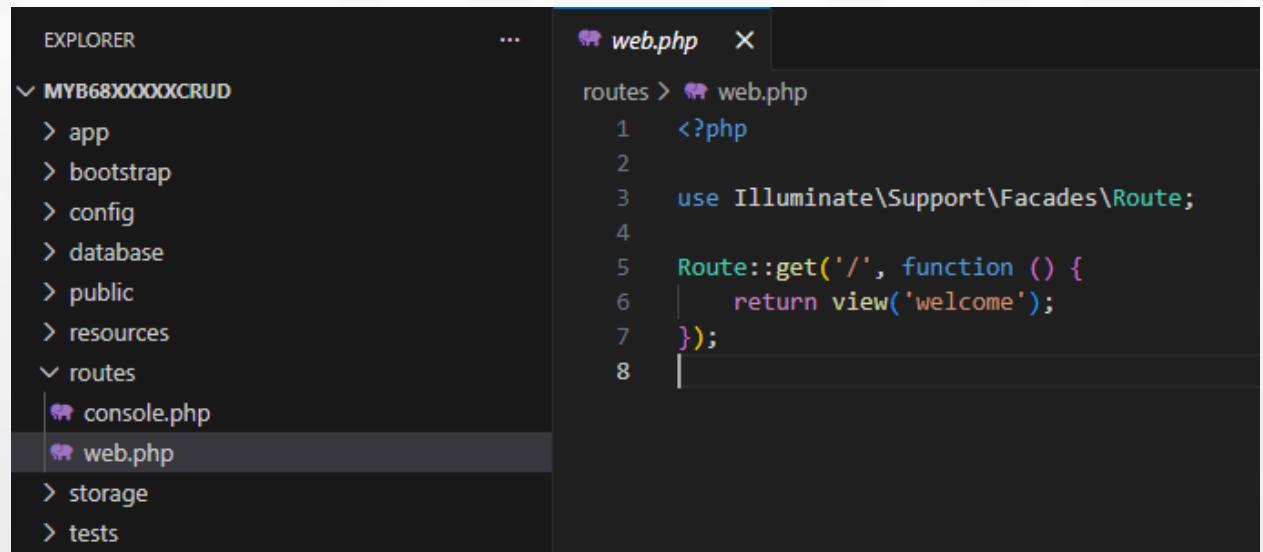
- การทำงาน:
 1. เตรียมข้อมูล: ดึงรายชื่อวิชาทั้งหมด (Course::all()) เพื่อไปทำตัวเลือก Dropdown ให้ผู้ใช้กดค้นหา
 2. ตรวจสอบการค้นหา: เช็คว่ามีการส่งค่า course_id มาหรือไม่
 - ถ้ามี: จะไป Query ข้อมูลจาก EnrollmentDetail โดยเลือกเฉพาะวิชานั้น และเรียงลำดับตามรหัสนักศึกษา (orderBy)
 - ถ้าไม่มี: ตัวแปร \$results จะเป็นค่าว่าง (ยังไม่แสดงตาราง)
- Return: ส่งข้อมูลไปที่หน้า View enrollments.report

3.12. กำหนด Routes (routes/web.php) สำหรับแสดงหน้าเว็บ

โดยให้เพิ่มโค้ดบรรทัดนี้ลงไป:

```
use Illuminate\Support\Facades\Route;  
use App\Http\Controllers\EnrollmentController;  
  
Route::resource('enrollments', EnrollmentController::class);  
Route::get('/', [EnrollmentController::class, 'index']);  
Route::post('/students', [EnrollmentController::class, 'storeStudent'])-  
>name('students.store');  
// จ่อเพิ่ม route details  
// จ่อเพิ่ม route report
```

ก่อนเพิ่ม



The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane displays a file structure under 'MYB68XXXX(CRUD)'. The 'routes' folder is expanded, showing 'console.php' and 'web.php'. The 'web.php' file is currently selected. On the right, the Editor pane shows the code for 'web.php':

```
<?php  
use Illuminate\Support\Facades\Route;  
Route::get('/', function () {  
    return view('welcome');  
});
```

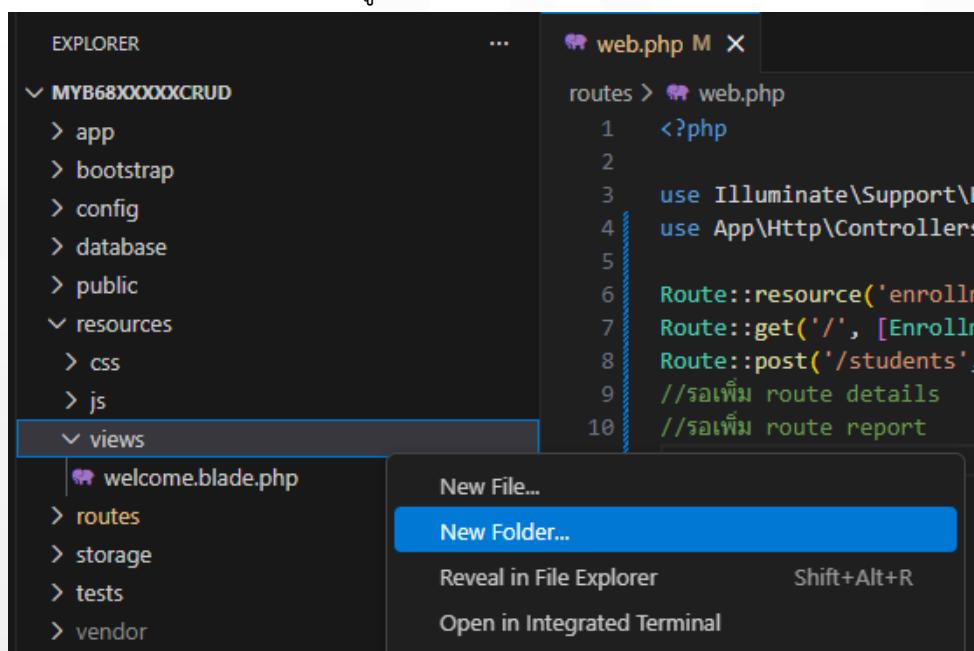
หลังเพิ่ม

```

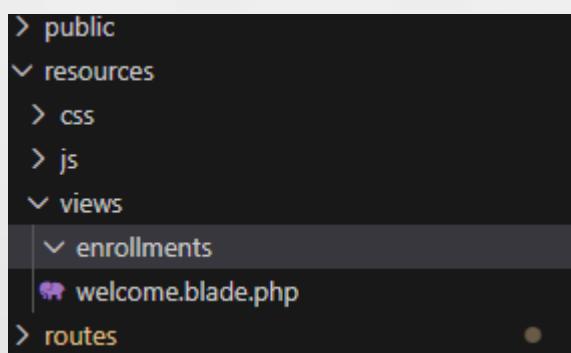
EXPLORER ... routes > web.php M X
MYB68XXXXCRUD <?php
> app
> bootstrap
> config
> database
> public
> resources
> routes
  console.php
  web.php M
  storage
  tests
routes > web.php
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\EnrollmentController;
5
6 Route::resource('enrollments', EnrollmentController::class);
7 Route::get('/', [EnrollmentController::class, 'index']);
8 Route::post('/students', [EnrollmentController::class, 'storeStudent'])->name('students.store');
9 //เพิ่ม route details
10 //เพิ่ม route report
11
12

```

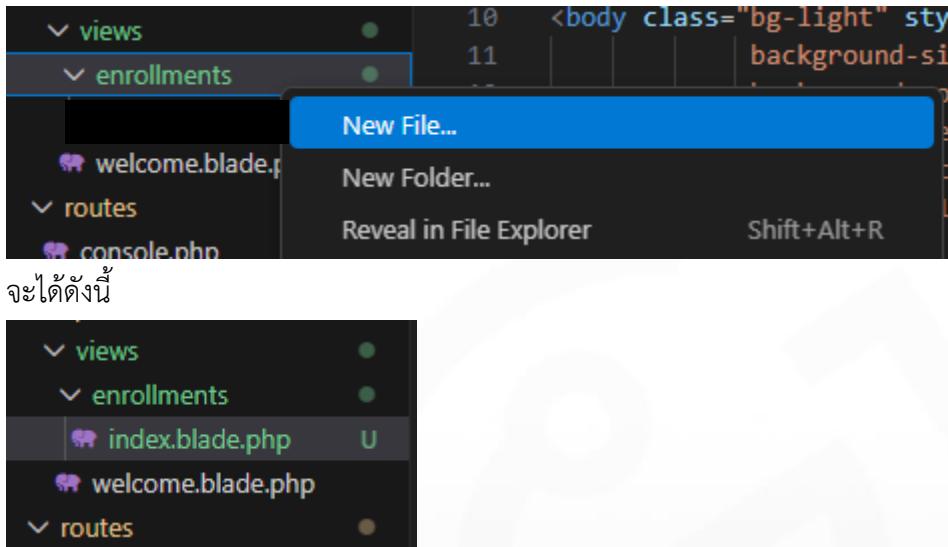
3.13. จักนั้นให้รีบสร้างหน้าเว็บ โดยใช้สร้างโฟลเดอร์ที่มีชื่อว่า “enrollments” โดยการคลิกขวาที่โฟลเดอร์ views และพิมพ์ชื่อตั้งรูป



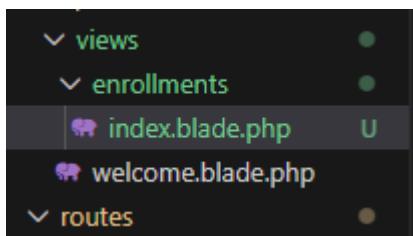
เมื่อสร้างโฟลเดอร์แล้วจะได้ดังนี้



3.14. จากนั้นให้กดคลิกขวาที่โฟลเดอร์ enrollments > จากนั้นเลือก New file > จากนั้นตั้งชื่อไฟล์ว่า “index.blade.php”



จะได้ดังนี้



จากนั้นให้เพิ่มโค้ดต่างในไฟล์ index.blade.php ให้ครบถ้วนตามรูปต่อไปนี้

***รูปโค้ดอยู่ในหน้าถัดไป

```

web.php M index.blade.php U
resources > views > enrollments > index.blade.php
1  <!DOCTYPE html>
2  <html lang="th">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>ระบบลงทะเบียนเรียน (Database Project)</title>
7      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
8      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
9  </head>
10 <body class="bg-light" style="background-image: url('{{ asset('images/bg.png') }}');>
11     <div style="background-size: cover; background-position: center; background-repeat: no-repeat; background-attachment: fixed; min-height: 100vh;">
12
13     <div class="container mt-5">
14         <div class="row mb-4">
15             <div class="col-md-8">
16                 <!-- ข้อความหัวข้อ -->
17                 <h2 class="text-primary text-muted"><i class="fas fa-university"></i> ระบบลงทะเบียนเรียน</h2>
18                 <p class="text-muted">ตัวอย่างการใช้ SQL View ร่วมกับ Laravel</p>
19
20                 <!-- ปุ่มรายงาน ให้เพิ่มค่าที่ต้องไป -->
21
22                 <!-- สั่นสอดบุญรายงาน -->
23
24             </div>
25             <div class="col-md-4 text-end">
26                 <!-- ปุ่มเพิ่มใหม่เพื่อข้อมูล -->
27                 <button type="button" class="btn btn-success" data-bs-toggle="modal" data-bs-target="#createModal">
28                     <i class="fas fa-plus"></i> ลงทะเบียนเพิ่ม
29                 </button>
30                 <!-- ปุ่มเพิ่มใหม่เพื่อเพิ่มผู้ใช้งาน -->
31                 <button type="button" class="btn btn-warning ms-2" data-bs-toggle="modal" data-bs-target="#createStudentModal">
32                     <i class="fas fa-user-plus"></i> เพิ่มผู้ใช้งาน
33                 </button>
34             </div>
35         </div>
36     </div>
37
38     <!-- แสดงข้อความสถานะ -->
39     @if(session('success'))
40         <div class="alert alert-success">{{ session('success') }}</div>
41     @endif
42
43     <!-- ตารางแสดงข้อมูลการลงทะเบียน -->
44     <div class="card shadow-sm">
45         <div class="card-body">
46             <table class="table table-hover align-middle">
47                 <thead class="table-dark">
48                     <tr>
49                         <th>รหัสผู้ใช้งาน</th>
50                         <th>ชื่อผู้ใช้งาน</th>
51                         <th>คณะ</th>
52                         <th>วิชา</th>
53                         <th>อาจารย์ผู้สอน</th>
54                         <th>เกรด</th>
55                         <th class="text-center">จัดการ</th>
56                     </tr>
57                 </thead>
58                 <tbody>
59                     <!-- รายการลงทะเบียน -->
60                     @foreach($enrollments as $row)
61                         <tr>
62                             <!-- ID -->
63                             <td><span class="badge bg-secondary">{{ $row->student_code }}</span></td>
64                             <!-- ลิงก์ไปยังหน้ารายละเอียดผู้ใช้งาน -->
65                             <td>
66                                 <!-- ชื่อผู้ใช้งานไปที่หน้า details ได้ ไม่ต้องคลิก -->
67                                 <a class="text-decoration-none fw-bold">
68                                     {{ $row->student_name }}
69                                 </a>
70                             </td>
71                             <!-- คณะ -->
72                             <td><small class="text-muted">{{ $row->department_name }}</small></td>
73                             <!-- ข้อมูลวิชา -->
74                             <td>
75                                 <strong>{{ $row->course_code }}</strong><br>
76                                 {{ $row->course_name }}
77                             </td>
78                         </tr>
79                     @endforeach
80                 </tbody>
81             </table>
82         </div>
83     </div>

```

```

80      </td>
81      <!-- ชื่ออาจารย์ผู้สอน -->
82      <td>{{ $row->instructor_name }}</td>
83      <!-- เกรด -->
84      <td>
85          @if($row->grade)
86              <span class="badge bg-primary">{{ $row->grade }}</span>
87          @else
88              <span class="badge bg-warning text-dark">รอเกรด</span>
89          @endif
90      </td>
91      <!-- ปุ่มจัดการ -->
92      <td class="text-center">
93          <!-- ปุ่มแก้ไขเกรด ให้เพิ่มอาทิตย์ต่อไป -->
94
95          <!-- สีน้ำเงินแก้ไขเกรด -->
96          <!-- ฟอร์มเพิ่มการลงทะเบียน ให้เพิ่มอาทิตย์ต่อไป -->
97
98          <!-- สีน้ำเงินฟอร์มลบการลงทะเบียน -->
99      </td>
100     <!-- ปุ่มลบ -->
101
102     <!-- ไม่ต้องแก้ไขเกรด ให้เพิ่มอาทิตย์ต่อไป -->
103
104     <!-- สีน้ำเงินตัวเลือกแก้ไขเกรด -->
105     @endforeach
106
107     </tbody>
108
109     </table>
110
111     </div>
112
113     <!-- โน้มตัวเพิ่มการลงทะเบียนใหม่ -->
114     <div class="modal fade" id="createModal" tabindex="-1">
115         <div class="modal-dialog">
116             <div class="modal-content">
117                 <form action="{{ route('enrollments.store') }}" method="POST">
118                     @csrf
119                     <div class="modal-header">
120                         <h5 class="modal-title">ลงทะเบียนรายวิชาใหม่</h5>
121                         <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
122                     </div>
123                     <div class="modal-body">
124                         <div class="mb-3">
125                             <label class="form-label">นักศึกษา</label>
126                             <select name="student_id" class="form-select" required>
127                                 <option value="">-- เลือกนักศึกษา --</option>
128                                 @foreach($students as $s)
129                                     <option value="{{ $s->id }}>{{ $s->student_code }} - {{ $s->name }}</option>
130                                 @endforeach
131                             </select>
132                         </div>
133                         <div class="mb-3">
134                             <label class="form-label">วิชาที่ลงทะเบียน</label>
135                             <select name="course_id" class="form-select" required>
136                                 <option value="">-- เลือกวิชา --</option>
137                                 @foreach($courses as $c)
138                                     <option value="{{ $c->id }}>{{ $c->code }} - {{ $c->name }} ({{ $c->credits }} หน่วยกิต)</option>
139                                 @endforeach
140                             </select>
141                         </div>
142                     <div class="modal-footer">
143                         <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">ปิด</button>
144                         <button type="submit" class="btn btn-success">ลงทะเบียน</button>
145                     </div>
146                 </form>
147             </div>
148         </div>
149     </div>
150
151     <!-- สีน้ำเงินตัวเพิ่มการลงทะเบียนใหม่ -->

```

```

152
153    <!-- เพิ่มฐานข้อมูลนักศึกษาใหม่ -->
154    <div class="modal fade" id="createStudentModal" tabindex="-1">
155        <div class="modal-dialog">
156            <div class="modal-content">
157                <form action="{{ route('students.store') }}" method="POST">
158                    @csrf
159                    <div class="modal-header">
160                        <h5 class="modal-title">เพิ่มฐานข้อมูลนักศึกษา</h5>
161                        <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
162                    </div>
163                    <div class="modal-body">
164                        <div class="mb-3">
165                            <label class="form-label">รหัสนักศึกษา</label>
166                            <input type="text" name="student_code" class="form-control" required placeholder="เช่น 661234">
167                        </div>
168                        <div class="mb-3">
169                            <label class="form-label">ชื่อ-นามสกุล</label>
170                            <input type="text" name="name" class="form-control" required placeholder="เช่น นายรักเรียน เพียรศึกษา">
171                        </div>
172                        <div class="mb-3">
173                            <label class="form-label">สังกัดคณะ</label>
174                            <select name="department_id" class="form-select" required>
175                                <option value="">-- เลือกคณะ --</option>
176                                @foreach($departments as $dept)
177                                    <option value="{{ $dept->id }}>{{ $dept->name }}</option>
178                                @endforeach
179                            </select>
180                        </div>
181                    </div>
182                    <div class="modal-footer">
183                        <button type="submit" class="btn btn-primary">บันทึกข้อมูล</button>
184                    </div>
185                </form>
186            </div>
187        </div>
188        <!-- สิ้นสุดเพิ่มฐานข้อมูลนักศึกษาใหม่ -->
189    </div>
190 </div>
191 <!-- Bootstrap JS -->
192 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
193 </body>
194 </html>

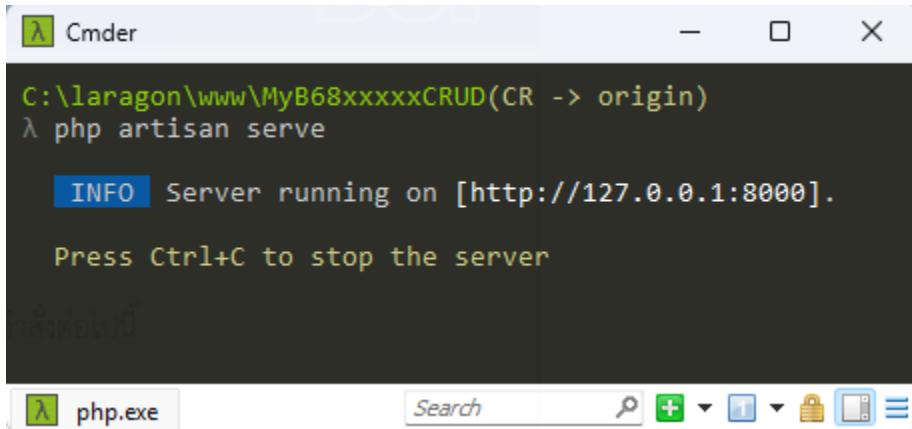
```

***โค้ดสามารถคัดลอกได้โดยไปที่ลิงค์นี้

<https://raw.githubusercontent.com/pyrnmsutdgt/MyB68xxxxCRUD/refs/heads/CR/resources/views/enrollments/index.blade.php>

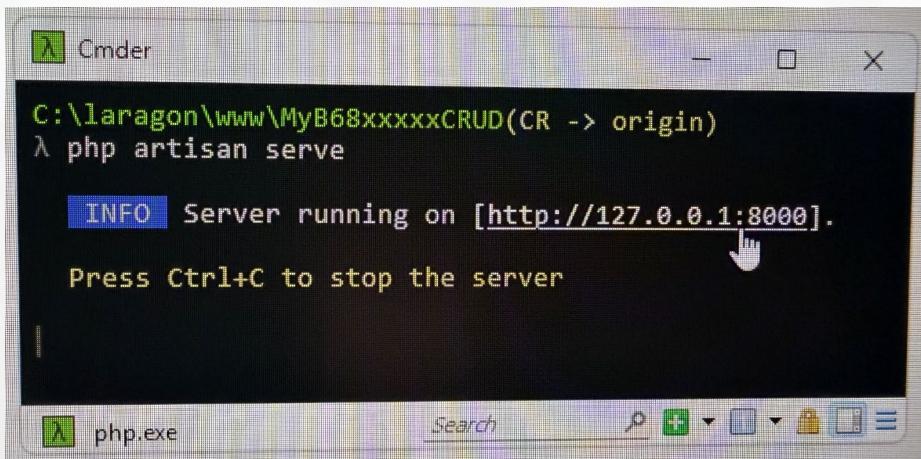
3.15. จากนั้น ให้เปิด Terminal ใน Laragon (Cmder) และใช้คำสั่งต่อไปนี้

```
php artisan serve
```

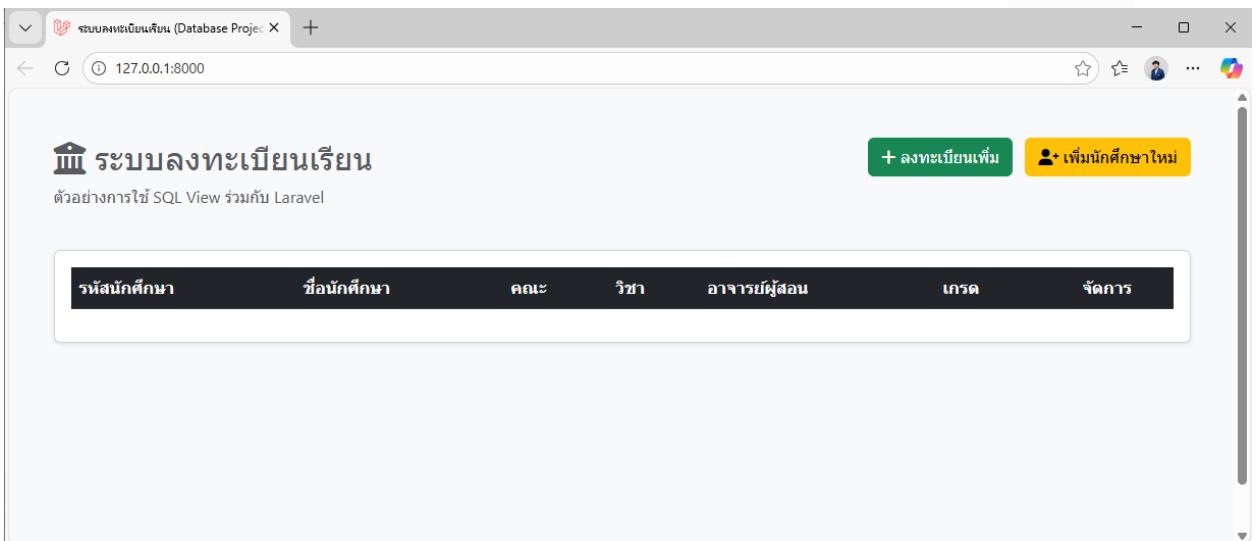


The screenshot shows a Cmder terminal window with the following text:
C:\laragon\www\MyB68xxxxxCRUD(CR -> origin)
λ php artisan serve
INFO Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server

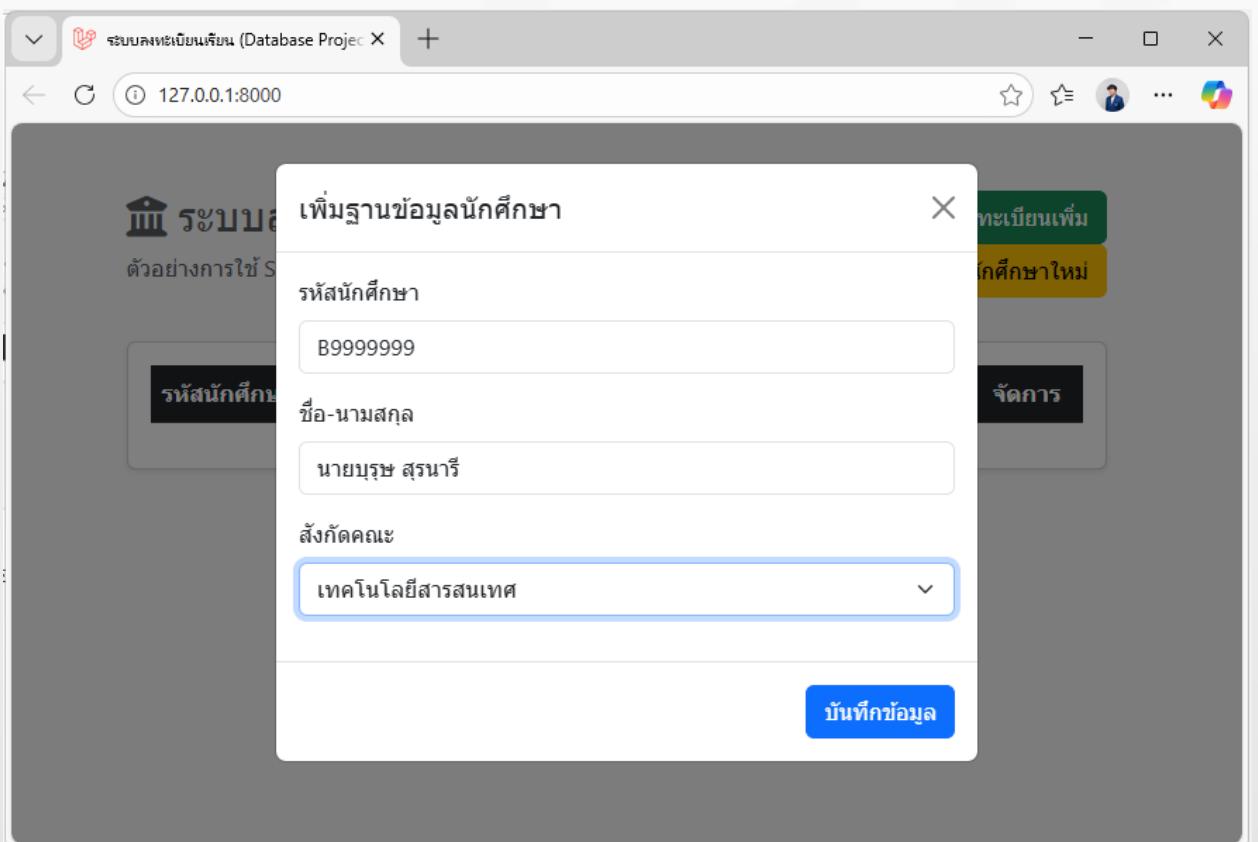
3.16. จากนั้นให้กด ctrl + เลื่อนกursorมาสีเป็นสีที่ [http://127.0.0.1:8000] และกดที่ลิงค์นั้นเพื่อเปิดหน้าเว็บ



จากนั้นหน้าเว็บจะเปิดขึ้นมา (ตัวอย่างหน้าถัดไป)



3.17. จากนั้นให้ลองเพิ่มชื่อโดยกดปุ่ม **+ เพิ่มนักศึกษาใหม่** โดยเพิ่มชื่อต่อไปนี้ลงไป แล้ว กดบันทึกข้อมูล



หากเพิ่มได้จะมีข้อความขึ้นมา

The screenshot shows the main interface of the system with a navigation bar at the top. A green button labeled '+ ลงทะเบียนเพิ่ม' (Add Registration) is highlighted. Below it, a modal window titled 'ลงทะเบียนรายวิชาใหม่' (New Subject Registration) is open. Inside the modal, there are fields for 'รหัสนักศึกษา' (Student ID) set to '89999999 - นายบุรุษ สุวนารี', 'วิชาที่ลงทะเบียน' (Subject Registered) set to 'IT202 - ฐานข้อมูล (3 หน่วยกิต)', and two buttons at the bottom: 'ปิด' (Close) and 'ลงทะเบียน' (Register).

3.18. ให้กดปุ่ม **+ ลงทะเบียนเพิ่ม** เพื่อทดลองเพิ่มและแสดงรายชื่อผู้ลงทะเบียน ดังนี้

This screenshot shows the same modal window from the previous step, but now it is filled with sample data. The student ID is '89999999 - นายบุรุษ สุวนารี', the subject registered is 'IT202 - ฐานข้อมูล (3 หน่วยกิต)', and the 'ลงทะเบียน' (Register) button is highlighted.

จากนั้นกดปุ่ม ลงทะเบียน หลังลงทะเบียนได้ จะขึ้นข้อความและรายชื่อขึ้นมา ดังนี้

The screenshot shows the main list of registered students. A single row is selected, showing the student's information: '89999999' (รหัสนักศึกษา), 'นายบุรุษ สุวนารี' (ชื่อนักศึกษา), 'เทคโนโลยีสารสนเทศ' (คณะ), 'IT202' (วิชา), 'ฐานข้อมูล' (รายวิชา), 'อ.วิภา รักเรียน' (อาจารย์ผู้สอน), and 'รอบรด' (เกรด). The 'ลงทะเบียน' (Register) button is also visible in the list.

ลายเซ็นต์ที่ 1 มีดังนี้

- สามารถกำหนดและตั้งค่าการติดต่อฐานข้อมูลได้
- สามารถสร้างตารางได้ 5 ตาราง ได้แก่ ตาราง Departments , Students , Instructors , Courses และ Enrollments
- เมื่อใช้คำสั่ง **php artisan db:show** ต้องเขียนตามรูปดังนี้
(ลูกศรแดงคือตาราง ลูกศรเขียวคือ view)

```
C:\laragon\www\MyB68xxxxCRUD(CR -> origin)
λ php artisan db:show

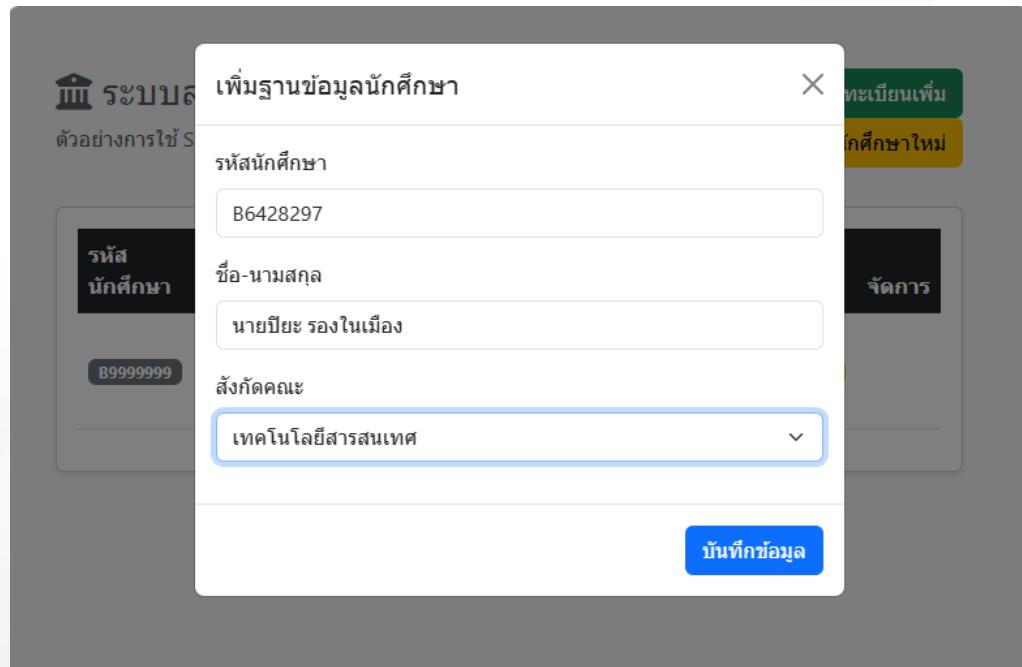
MySQL ..... 8.0.30
Connection ..... mysql
Database ..... MyB68xxxxxCRUD
Host ..... 127.0.0.1
Port ..... 3307
Username ..... root
URL ..... .
Open Connections ..... 1
Tables ..... 39
Total Size ..... 0.91 MB

Schema / Table ..... Size
myb68xxxxxcrud / cache ..... 16.00 KB
myb68xxxxxcrud / cache_locks ..... 16.00 KB
myb68xxxxxcrud / courses ..... 32.00 KB
myb68xxxxxcrud / departments ..... 16.00 KB
myb68xxxxxcrud / enrollment_details_view ..... 16.00 KB
myb68xxxxxcrud / enrollments ..... 48.00 KB
myb68xxxxxcrud / failed_jobs ..... 16.00 KB
myb68xxxxxcrud / instructors ..... 32.00 KB
myb68xxxxxcrud / job_batches ..... 16.00 KB
myb68xxxxxcrud / jobs ..... 16.00 KB
myb68xxxxxcrud / migrations ..... 16.00 KB
myb68xxxxxcrud / password_reset_tokens ..... 16.00 KB
myb68xxxxxcrud / sessions ..... 48.00 KB
myb68xxxxxcrud / students ..... 32.00 KB
myb68xxxxxcrud / university_tables ..... 16.00 KB
myb68xxxxxcrud / users ..... 16.00 KB

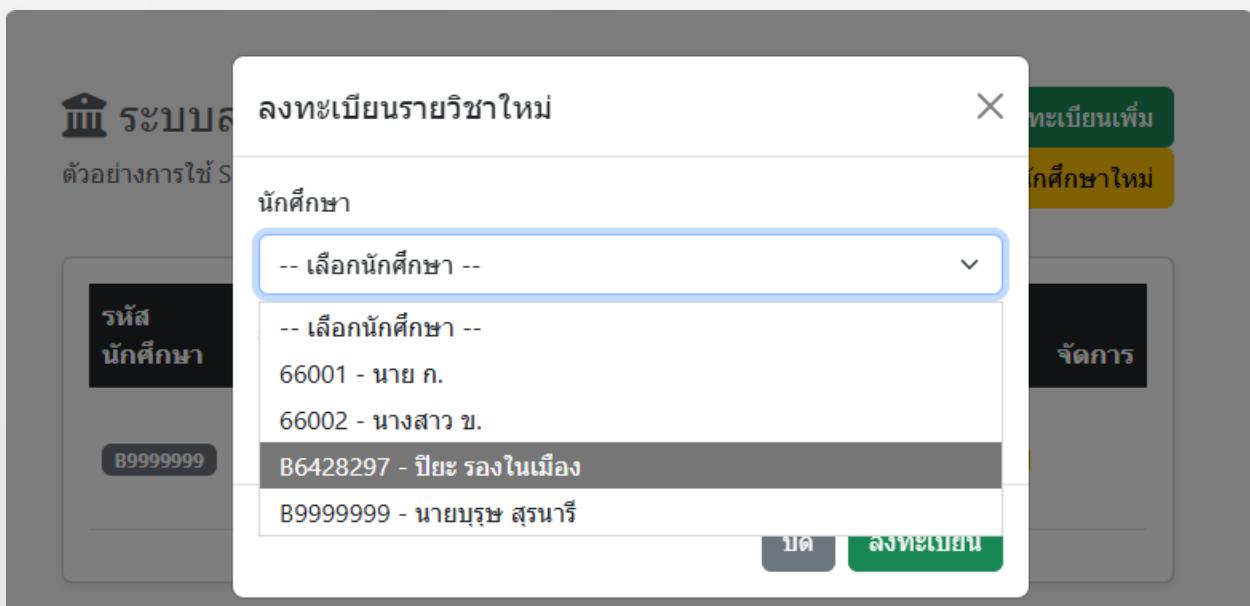
C:\laragon\www\MyB68xxxxCRUD(CR -> origin)
λ
```

ลายเข็มที่ 2 สามารถเพิ่มข้อมูลได้ โดยให้เพิ่มข้อมูลชื่อและรหัสนักศึกษา ตัวอย่างตามรูป

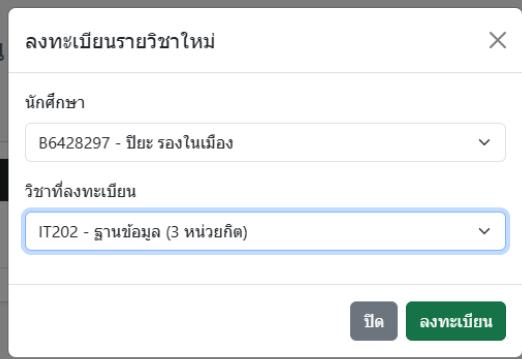
- รหัสนักศึกษา – ใช้รหัสของนักศึกษา
- ชื่อ-นามสกุล – ชื่อ-นามสกุล ของนักศึกษา
- สังกัดคณะ – เลือกคณะได้



ถ้าเพิ่มได้ เมื่อกดปุ่ม +ลงทะเบียนเพิ่ม จะมี ชื่อ-นามสกุลของนักศึกษา



ลายเขียน 3 สามารถเพิ่มการลงทะเบียนรายวิชาได้ และแสดงผลการลงทะเบียนรายวิชาได้ ตามรายชื่อนักศึกษา ตัวอย่างตามรูป



ระบบลงทะเบียนเรียน

ตัวอย่างการใช้ SQL View ร่วมกับ Laravel

ลงทะเบียนรายวิชาใหม่

นักศึกษา

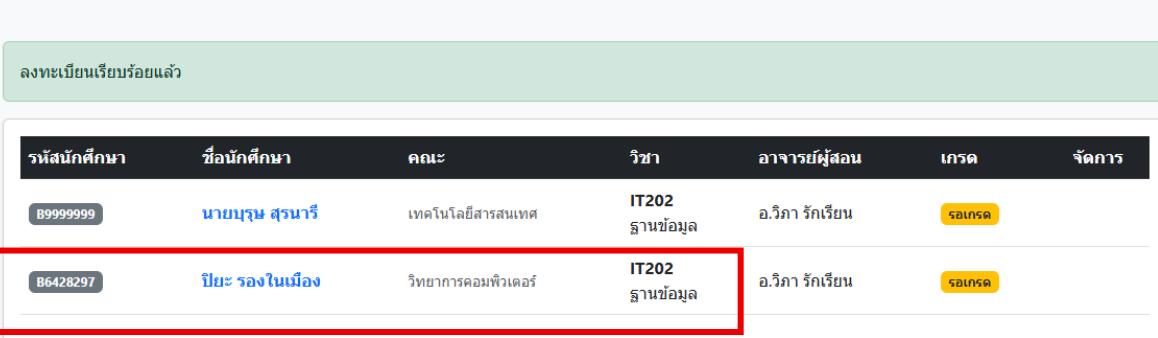
B6428297 - ปีช รองในเมือง

วิชาที่ลงทะเบียน

IT202 - ฐานข้อมูล (3 หน่วยกิต)

ปิด ลงทะเบียน

เมื่อเลือกชื่อ-นามสกุลของนักศึกษาและวิชาตามรูปแล้ว เมื่อกดปุ่มลงทะเบียน จะมีข้อความแจ้งเตือน ใจนั้นข้อมูลนักศึกษาขึ้นมา ดังรูปต่อไปนี้



ระบบลงทะเบียนเรียน

ตัวอย่างการใช้ SQL View ร่วมกับ Laravel

รหัสนักศึกษา	ชื่อนักศึกษา	คณะ	วิชา	อาจารย์ผู้สอน	เกรด	จัดการ
B9999999	นายบุญ สุวนารี	เทคโนโลยีสารสนเทศ	IT202 ฐานข้อมูล	อ.วิภา รักเรียน	รอบทดลอง	
B6428297	ปีช รองในเมือง	วิทยาการคอมพิวเตอร์	IT202 ฐานข้อมูล	อ.วิภา รักเรียน	รอบทดลอง	